

N. R. Shetty · L. M. Patnaik ·
H. C. Nagaraj · Prasad Naik Hamsavath ·
N. Nalini *Editors*

Emerging Research in Computing, Information, Communication and Applications

ERCICA 2018, Volume 1

Advances in Intelligent Systems and Computing

Volume 882

Series Editor

Janusz Kacprzyk, Systems Research Institute, Polish Academy of Sciences,
Warsaw, Poland

Advisory Editors

Nikhil R. Pal, Indian Statistical Institute, Kolkata, India

Rafael Bello Perez, Faculty of Mathematics, Physics and Computing, Universidad
Central de Las Villas, Santa Clara, Cuba

Emilio S. Corchado, University of Salamanca, Salamanca, Spain

Hani Hagrass, Electronic Engineering, University of Essex, Colchester, UK

László T. Kóczy, Department of Automation, Széchenyi István University,
Gyor, Hungary

Vladik Kreinovich, Department of Computer Science, University of Texas
at El Paso, El Paso, TX, USA

Chin-Teng Lin, Department of Electrical Engineering, National Chiao Tung
University, Hsinchu, Taiwan

Jie Lu, Faculty of Engineering and Information Technology, University of
Technology Sydney, Sydney, NSW, Australia

Patricia Melin, Graduate Program of Computer Science, Tijuana Institute of
Technology, Tijuana, Mexico

Nadia Nedjah, Department of Electronics Engineering, University of Rio de
Janeiro, Rio de Janeiro, Brazil

Ngoc Thanh Nguyen, Faculty of Computer Science and Management, Wrocław
University of Technology, Wrocław, Poland

Jun Wang, Department of Mechanical and Automation Engineering, The Chinese
University of Hong Kong, Shatin, Hong Kong

The series “Advances in Intelligent Systems and Computing” contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing such as: computational intelligence, soft computing including neural networks, fuzzy systems, evolutionary computing and the fusion of these paradigms, social intelligence, ambient intelligence, computational neuroscience, artificial life, virtual worlds and society, cognitive science and systems, Perception and Vision, DNA and immune based systems, self-organizing and adaptive systems, e-Learning and teaching, human-centered and human-centric computing, recommender systems, intelligent control, robotics and mechatronics including human-machine teaming, knowledge-based paradigms, learning paradigms, machine ethics, intelligent data analysis, knowledge management, intelligent agents, intelligent decision making and support, intelligent network security, trust management, interactive entertainment, Web intelligence and multimedia.

The publications within “Advances in Intelligent Systems and Computing” are primarily proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

**** Indexing: The books of this series are submitted to ISI Proceedings, EI-Compendex, DBLP, SCOPUS, Google Scholar and Springerlink ****

More information about this series at <http://www.springer.com/series/11156>

N. R. Shetty · L. M. Patnaik ·
H. C. Nagaraj · Prasad Naik Hamsavath ·
N. Nalini
Editors

Emerging Research in Computing, Information, Communication and Applications

ERCICA 2018, Volume 1

 Springer

Editors

N. R. Shetty
Central University of Karnataka
Kalaburagi, Karnataka, India

L. M. Patnaik
National Institute of Advanced Studies
Bangalore, Karnataka, India

H. C. Nagaraj
Nitte Meenakshi Institute of Technology
Bangalore, Karnataka, India

Prasad Naik Hamsavath
Nitte Meenakshi Institute of Technology
Bangalore, Karnataka, India

N. Nalini
Nitte Meenakshi Institute of Technology
Bangalore, Karnataka, India

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-981-13-5952-1 ISBN 978-981-13-5953-8 (eBook)
<https://doi.org/10.1007/978-981-13-5953-8>

Library of Congress Control Number: 2018966829

© Springer Nature Singapore Pte Ltd. 2020

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd. The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Organizing Committee

ERCICA 2018

The Fifth International Conference on “Emerging Research in Computing, Information, Communication and Applications”, ERCICA 2018, was held during July 27–28, 2018, at the Nitte Meenakshi Institute of Technology (NMIT), Bangalore, and organized by the Departments of CSE and MCA, NMIT

Chief Patrons

Dr. N. V. Hegde, President, Nitte Education Trust, Mangalore, India
Dr. N. R. Shetty, Chancellor, Central University of Karnataka, Kalburgi, and
Advisor, Nitte Education Trust, Mangalore, India

Conference Chair

Dr. H. C. Nagaraj, Principal, NMIT, Bangalore, India

Program Chairs

Dr. Prasad Naik Hamsavath, HOD, MCA, NMIT, Bangalore, India
Dr. N. Nalini, Professor, CSE, NMIT, Bangalore, India

Publisher

Springer

Advisory Chairs

Dr. K. Sudha Rao, Advisor, Admin and Management, NMIT, Bangalore, India

Mr. Rohit Punja, Administrator, NET, Mangalore, India

Dr. Jharna Majumdar, Dean (R&D), NMIT, Bangalore, India

Mr. K. A. Ranganatha Setty, Dean (Academic), NMIT, Bangalore, India

Advisory Committee

Dr. L. M. Patnaik, INSA Senior Scientist, NIAS, Bangalore, India

Dr. B. S. Sonde, Former Vice Chancellor, Goa University, Goa, India

Dr. D. K. Subramanian, Former Dean and Professor, IISc, Bangalore, India

Dr. K. D. Nayak, Former OS & CC, R&D (MED & MIST), DRDO, India

Dr. Kalidas Shetty, Founding Director of Global Institute of Food Security and International Agriculture (GIFSIA), North Dakota State University, Fargo, USA

Dr. Kendall E. Nygard, Professor of Computer Science and Operations Research, North Dakota State University, Fargo, USA

Dr. Sathish Udpa, Dean and Professor, Michigan State University, Michigan, USA

Dr. K. N. Bhat, Visiting Professor, Center for Nano Science and Engineering-CeNSE, IISc, Bangalore, India

Dr. K. R. Venugopal, Principal, UVCE, Bangalore, India

Dr. C. P. Ravikumar, Director, Technical Talent Development at Texas Instruments, Bangalore, India

Dr. Navakanta Bhat, Chairperson, Center for Nano Science and Engineering-CeNSE, IISc, Bangalore, India

Dr. Anand Nayyar, Professor, Researcher and Scientist in Graduate School, Duy Tan University, Da Nang, Vietnam

Program Committee

Dr. Savitri Bevinakoppa, Professional Development and Scholarship Coordinator, School of IT and Engineering, Melbourne Institute of Technology (MIT), Australia

Dr. P. Ramprasad, Professor, Department of CSE and IT, Manipal University, Dubai

Dr. Ohta Tsuyoshi, Department of Computer Sciences, Shizuoka University, Japan

Dr. Sonajharia Minz, Professor, School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India

Dr. Sanjay Kumar Dhurandher, Professor and Head, Department of Information Technology, Netaji Subhas Institute of Technology, New Delhi, India

Dr. Ramesh R. Galigekere, Professor and HOD, Department of Biomedical Networks, University Putra Malaysia, Malaysia

Dr. K. G. Srinivasa, CBP Government Engineering College, New Delhi, India

Dr. S. Ramanarayana Reddy, HOD, Department of CSE, IGDTU for Women, Kashmere Gate, Delhi, India

Dr. K. Raghavendra, Professor, School of Computing, National University of Singapore, Singapore

Dr. Abhijit Lele, Principal Consultant, Robert Bosch, Bangalore, India

Dr. Bappaditya Mandal, Faculty of Science, Engineering and Computing, Kingston University, London

Organizing Co-chairs

Dr. M. N. Thippeswamy, Professor and Head, Department of CSE, NMIT, Bangalore, India

Dr. H. A. Sanjay, Professor and Head, Department of ISE, NMIT, Bangalore, India

Dr. S. Sandya, Professor and Head, Department of ECE, NMIT, Bangalore, India

Dr. H. M. Ravikumar, Professor and Head, Department of EEE, NMIT, Bangalore, India

Theme Editors

Computing

Dr. N. Nalini
Dr. Thippeswamy M. N.

Information

Dr. Sanjay H. A.
Dr. Shakti Mishra

Communication

Dr. H. C. Nagaraj
Dr. Raghunandan S.
Prof. Sankar Dasiga

Application

Dr. Prasad Naik Hamsavath
Prof. Sitaram Yaji

Preface

The Fifth International Conference on “Emerging Research in Computing, Information, Communication and Applications,” ERCICA 2018, is an annual event organized at the Nitte Meenakshi Institute of Technology (NMIT), Yelahanka, Bangalore, India.

ERCICA aims to provide an interdisciplinary forum for discussion among researchers, engineers and scientists to promote research and exchange of knowledge in computing, information, communication and related applications. This conference will provide a platform for networking of academicians, engineers and scientists and also will entuse the participants to undertake high-end research in the above thrust areas.

ERCICA 18 received more than 400 papers from all over the world, viz. from China, UK, Africa, Saudi Arabia and India. The ERCICA Technical Review Committee has followed all necessary steps to screen more than 400 papers by going through six rounds of quality checks on each paper before selection for presentation/publication.

The acceptance ratio is only 1:3.

Kalaburagi, India
Bangalore, India
July 2018

N. R. Shetty
L. M. Patnaik
H. C. Nagaraj
Prasad Naik Hamsavath
N. Nalini

Acknowledgements

First of all, we would like to thank Prof. N. R. Shetty who has always been the guiding force behind this event's success. It was his dream that we have striven to make a reality. Our thanks to Prof. L. M. Patnaik, who has monitored the whole activity of the conference from the beginning till its successful end.

Our special thanks to Springer and especially the editorial staff who were patient, meticulous and friendly with their constructive criticism on the quality of papers and outright rejection at times without compromising the quality of the papers as they are always known for publishing the best international papers.

We would like to express our gratitude to all the review committee members of all the themes of computing, information, communication and applications and the best-paper-award review committee members.

Finally, we would like to express our heartfelt gratitude and warmest thanks to the ERCICA 2018 organizing committee members for their hard work and outstanding efforts. We know how much time and energy this assignment demanded, and we deeply appreciate all the efforts to make it a grand success.

Our special thanks to all the authors who have contributed to publishing their research work in this conference and participated to make this conference a grand success. Thanks to everyone who have directly or indirectly contributed to the success of this conference ERCICA 2018.

Regards
Program Chairs
ERCICA 2018

About the Conference

ERCICA 2018

The Fifth International Conference on “Emerging Research in Computing, Information, Communication and Applications,” ERCICA 2018, is an annual event jointly organized by the Departments of CSE and MCA during July 27–28, 2018, at the Nitte Meenakshi Institute of Technology (NMIT), Yelahanka, Bangalore, India.

ERCICA 2018 is organized under the patronage of Prof. N. R. Shetty, Advisor, Nitte Education Trust. Dr. L. M. Patnaik, Technical Advisor, NMIT, and Dr. H. C. Nagaraj, Principal, served as the Conference Chairs, and the Program Chairs of the conference were Dr. Prasad Naik Hamsavath, Professor and Head, MCA, and Dr. N. Nalini, Professor, CSE, NMIT, Bangalore, Karnataka.

ERCICA aims to provide an interdisciplinary forum for discussion among researchers, engineers and scientists to promote research and exchange of knowledge in computing, information, communication and related applications. This conference will provide a platform for networking of academicians, engineers and scientists and also will enthruse the participants to undertake high-end research in the above thrust areas.

Contents

A Computational Segmentation Tool for Processing Patient Brain MRI Image Data to Automatically Extract Gray and White Matter Regions	1
Ayush Goyal, Sunayana Tirumalasetty, Disha Bathla, Manish K. Arya, Rajeev Agrawal, Priya Ranjan, Gahangir Hossain and Rajab Chaloo	
Developing Ontology for Smart Irrigation of Vineyards	17
Archana Chougule and Debajyoti Mukhopadhyay	
An Intensive Review of Data Replication Algorithms for Cloud Systems	25
Chetna Dabas and Juhi Aggarwal	
Integrated Cryptography for Internet of Things Using TBF Approach	41
Santosh Kumar Sharma, Muppidi Somasundara Rao, Lavudi Poorna Chandar Rao and Pendyala Chaitanya	
Data Governance on Local Storage in Offsite	53
G. Priyadharshini and K. Shyamala	
Extraction of Character Personas from Novels Using Dependency Trees and POS Tags	65
Nikhil Prabhu and S. Natarajan	
A Multifactor Authentication Model to Mitigate the Phishing Attack of E-Service Systems from Bangladesh Perspective	75
Md. Zahid Hasan, Abdus Sattar, Arif Mahmud and Khalid Hasan Talukder	
Study on Energy-Efficient and Lifetime-Enhanced Clustering Algorithm-Based Routing Protocols in Wireless Sensor Network	87
Ishita Banerjee and P. Madhumathy	

Role of Fog Computing in IoT-Based Applications	99
Charul Thareja and N. P. Singh	
Single Horizontal Camera-Based Object Tracking Quadcopter Using StaGaus Algorithm	113
Lakshmi Shrinivasan and N. R. Prasad	
IoT-Enabled Medicine Bottle	127
A. R. Shreyas, Soumya Sharma, H. Shivani and C. N. Sowmyarani	
Revamp Perception of Bitcoin Using Cognizant Merkle	141
J. Vijayalakshmi and A. Murugan	
A Novel Algorithm for DNA Sequence Compression	151
K. Punitha and A. Murugan	
DigiPen: An Intelligent Pen Using Accelerometer for Character Recognition	161
Ankur Agarwal and Swaroop Sudhanva Belur	
Design of FPGA-Based Radar and Beam Controller	171
Adesh Panwar and Neha Goyal	
Effect of Lattice Topologies and Distance Measurements in Self-Organizing Map for Better Classification	183
Sathiapriya Ramiah	
Evaluation and Classification of Road Accidents Using Machine Learning Techniques	193
Jaspreet Singh, Gurvinder Singh, Prithvipal Singh and Mandeep Kaur	
Multi-language Handwritten Recognition in DWT Accuracy Analysis	205
T. P. Umadevi and A. Murugan	
A Novel $H-\infty$ Filter Based Indicator for Health Monitoring of Components in a Smart Grid	221
E. Ranjini Warriar, P. V. Sunil Nag and C. Santhosh Kumar	
A Survey on Intelligent Transportation System Using Internet of Things	231
Palak Patel, Zunnun Narmawala and Ankit Thakkar	
CRUST: A C/C++ to Rust Transpiler Using a “Nano-parser Methodology” to Avoid C/C++ Safety Issues in Legacy Code	241
Nishanth Shetty, Nikhil Saldanha and M. N. Thippeswamy	
Species Environmental Niche Distribution Modeling for Panthera Tigris Tigris ‘Royal Bengal Tiger’ Using Machine Learning	251
Shaurya Bajaj and D. Geraldine Bessie Amali	

Organizational Digital Footprint for Traceability, Provenance Approach 265
 Sheetal Arya, Kumar Abhishek and Akshay Deepak

Bidirectional Long Short-Term Memory for Automatic English to Kannada Back-Transliteration 277
 B. S. Sowmya Lakshmi and B. R. Shambhavi

A Dominant Point-Based Algorithm for Finding Multiple Longest Common Subsequences in Comparative Genomics 289
 Manish M. Motghare and Preeti S. Voditel

Fast and Accurate Fingerprint Recognition in Principal Component Subspace 301
 S. P. Ragendhu and Tony Thomas

Smart Meter Analysis Using Big Data Techniques 317
 Neha Pandey, Nivedita Das, Sandeep Agarwal, Kashyap Barua, Manjusha Pandey and Siddharth Swarup Rautray

Movie Recommendation System 329
 S. Rajarajeswari, Sharat Naik, Shagun Srikant, M. K. Sai Prakash and Prarthana Uday

A Semiautomated Question Paper Builder Using Long Short-Term Memory Neural Networks 341
 Rajarajeswari Subramanian, Akhilesh P. Patil, Karthik Ganesan and T. S. Akarsh

Time-Critical Transmission Protocols in Wireless Sensor Networks: A Survey 351
 Archana R. Raut, S. P. Khandait and Urmila Shrawankar

Impact of Shuffler Design Pattern on Software Quality 365
 G. Priyalakshmi, R. Nadarajan, Joseph W. Yoder, S. Arthi and G. Jayashree

Efficient Algorithms for Text Lines and Words Segmentation for Recognition of Arabic Handwritten Script 387
 Amani Ali Ahmed Ali and M. Suresha

Privacy-Preserving Lightweight Image Encryption in Mobile Cloud 403
 M. Sankari and P. Ranjana

Performance Evaluation of Ensemble-Based Machine Learning Techniques for Prediction of Chronic Kidney Disease 415
 K. M. Zubair Hasan and Md. Zahid Hasan

Proficient Cooperative Caching in SWNET Using Twin Segments Approach	427
B. N. Lakshmi Narayan, Prasad N. Hamsavath, Meher Taj, E. G. Satish, Vivek Bharadwaj and S. Rabendranath	
A Particle Swarm Optimization-Backpropagation (PSO-BP) Model for the Prediction of Earthquake in Japan	435
Abey Abraham and V. Rohini	
Analysis and Detection of Diabetes Using Data Mining Techniques—A Big Data Application in Health Care	443
B. G. Mamatha Bai, B. M. Nalini and Jharna Majumdar	
Cyclic Scheduling Algorithm	457
Ravin Kumar	
A Framework for Monitoring Clustering Stability Over Time	467
K. Namitha and G. Santhosh Kumar	
Efficient Dynamic Double Threshold Energy Detection of Cooperative Spectrum Sensing in Cognitive Radio	479
Shahbaz Soofi, Anjali Potnis and Prashant Diwivedy	
ConvFood: A CNN-Based Food Recognition Mobile Application for Obese and Diabetic Patients	493
Kaiz Merchant and Yash Pande	
Segmentation and Recognition of <i>E. coli</i> Bacteria Cell in Digital Microscopic Images Based on Enhanced Particle Filtering Framework	503
Manjunatha Hiremath	
Automatic Bengali Document Categorization Based on Deep Convolution Nets	513
Md. Rajib Hossain and Mohammed Moshiul Hoque	
Artist Recommendation System Using Hybrid Method: A Novel Approach	527
Ajay Dhruv, Aastha Kamath, Anuja Powar and Karan Gaikwad	
Anomaly Detection of DDOS Attacks Using Hadoop	543
Y. S. Kalai vani and P. Ranjana	
Pedestrian Detection and Tracking: A Driver Assistance System	553
Shruti Maralappanavar, Nalini C. Iyer and Meena Maralappanavar	
Exploiting Parallelism Available in Loops Using Abstract Syntax Tree	563
Anil Kumar and Hardeep Singh	

A Study on Cooperation and Navigation Planning for Multi-robot Using Intelligent Water Drops Algorithm	577
D. Chandrasekhar Rao and Manas Ranjan Kabat	
Customer's Activity Recognition in Smart Retail Environment Using AltBeacon	591
M. Lakshmi, Alolika Panja, Naini and Shakti Mishra	
An Active Mixer Design For Down Conversion in 180 nm CMOS Technology for RFIC Applications	605
B. H. Shraddha and Nalini C. Iyer	
Analysis of PAPR for Performance QPSK and BPSK Modulation Techniques	621
K. Bhagyashree, S. Ramakrishna and Priyatam Kumar	
Implementation of Modified Array Multiplier for WiMAX Deinterleaver Address Generation	629
Patil Nikita, Arun Kakhandki, S. Ramakrishna and Priyatam Kumar	
Link Quality-Based Mobile-Controlled Handoff Analysis Using Stochastic Models	641
S. Akshitha and N. G. Goudru	
A Comparative Analysis of Lightweight Cryptographic Protocols for Smart Home	663
Rupali Syal	
Algorithm Study and Simulation Analysis by Different Techniques on MANET	671
Nithya Rekha Sivakumar and Abeer Al Garni	
Cloud-Based Agricultural Framework for Soil Classification and Crop Yield Prediction as a Service	685
K. Aditya Shastry and H. A. Sanjay	

About the Editors

Prof. N. R. Shetty is the Chancellor of Central University of Karnataka, Kalaburagi, and Chairman of the Review Commission for the State Private University Karnataka. He is currently serving as an advisor to the Nitte Meenakshi Institute of Technology (NMIT), Bangalore. He is also founder Vice-President of the International Federation of Engineering Education Societies (IFEES), Washington DC, USA. He served as Vice Chancellor of Bangalore University for two terms and President of the ISTE, New Delhi for three terms. He was also a member of the AICTE's Executive Committee and Chairman of its South West Region Committee.

Prof. L. M. Patnaik obtained his PhD in Real-Time Systems in 1978, and his DSc in Computer Systems and Architectures in 1989, both from the Indian Institute of Science, Bangalore. From 2008 to 2011, he was Vice Chancellor of the Defense Institute of Advanced Technology, Deemed University, Pune. Currently he is an Honorary Professor with the Department of Electronic Systems Engineering, Indian Institute of Science, Bangalore, and INSA Senior Scientist and Adjunct Professor with the National Institute of Advanced Studies, Bangalore.

Dr. H. C. Nagaraj completed his B.E in Electronics & Communication from the University of Mysore in 1981, his M.E in Communication Systems from P.S.G College of Technology, Coimbatore in 1984. He was awarded Ph.D (Biomedical Signal Processing and Instrumentation) from Indian Institute of Technology Madras, Chennai in 2000. Dr. Nagaraj has teaching experience spanning more than 35 years. He was the Chairman, BOS of IT/BM/ML of Visvesvaraya Technological University, Belagavi for 2010-13 and Member, Academic Senate of VTU for 06 years w.e.f. April 2010. Further extended for a period of three years w.e.f 02-06-2016. He has the credit of publishing more than 40 technical papers and has also published a book- "VLSI Circuits", Star- Tech Education, Bangalore in 2006. He has won the Best Student Paper Award at the 5th National Conference of Biomechanics held at I.I.T. Madras, Chennai in 1996 and Best Paper Award at the Karnataka State Level Seminar on "Introduction of Flexible System in Technical

Education” under Visveswaraiiah Technological University in 1999 at P.E.S. Institute of Technology, Bangalore. Presently, he is the Dean, Faculty of Engineering, Visvesvaraya Technological University Belgaum, for three years from 2016 to 2019 and Member of the Court, Pondicherry University. He is the Member of Karnataka State Innovation Council, Government of Karnataka and Member of NAAC (UGC) Peer Team to assess the institutions for Accreditation. He has also visited as an Expert Member of the UGC, New Delhi for inspecting the colleges seeking Autonomous Status.

Dr. Prasad Naik Hamsavath is a Professor and Head of the Department of Master of Computer Applications at Nitte Meenakshi Institute of Technology, Bangalore. He completed his PhD at Jawaharlal Nehru University, New Delhi, India. Dr. Prasad N H has more than 12 years of experience in different roles in both public and private sector enterprises, including the Ministry of Human Resource and Development, New Delhi, Government of India. He received the prestigious “Dr. Abdul Kalam Life Time Achievement Award” and also received a “Young Faculty” award at the 2nd Academic Brilliance Awards.

Dr. N. Nalini is a Professor at the Department of Computer Science and Engineering at Nitte Meenakshi Institute of Technology, Bangalore. She received her MS from BITS, Pilani in 1999 and her PhD from Visvesvaraya Technological University in 2007. She has more than 21 years of teaching and 14 years of research experience. She has written numerous international publications, and **Received “Bharath Jyoti Award” by India International Friendship Society, New Delhi on 2012**, from Dr. Bhishma Narain Singh, former Governor of Tamilnadu and Assam. She received the **“Dr. Abdul Kalam Life time achievement National Award” for excellence in Teaching, Research Publications, and Administration by International Institute for Social and Economic Reforms, IISER, Bangalore on 29th Dec 2014**. She is also the recipient of “Distinguished Professor” award by TechNext India 2017 in association with Computer Society of India-CSI, Mumbai Chapter and “Best Professor in Computer Science & Engineering “ award by **26th Business School Affaire & Dewang Mehta National Education Awards (Regional Round) on 5th September 2018, at Bangalore**. She is a lifetime member of the ISTE, CSI, ACEEE and IIFS.

A Computational Segmentation Tool for Processing Patient Brain MRI Image Data to Automatically Extract Gray and White Matter Regions



Ayush Goyal, Sunayana Tirumalasetty, Disha Bathla, Manish K. Arya, Rajeev Agrawal, Priya Ranjan, Gahangir Hossain and Rajab Chaloo

Abstract Brain MRI imaging is necessary to screen and detect diseases in the brain, and this requires processing, extracting, and analyzing a patient's MRI medical image data. Neurologists and neurological clinicians, technicians, and researchers would be greatly facilitated and benefited by a graphical user interface-based computational tool that could perform all the required medical MRI image processing functions automatically, thus minimizing the cost, effort, and time required in screening disease from the patient's MRI medical image data. Thus, there is a need for automatic medical image processing software platforms and for developing tools with applications in the medical field to assist neurologists, scientists, doctors, and academicians to analyze medical image data automatically to obtain patient-specific clinical parameters and information. This research develops an automatic brain MRI segmentation computational tool with a wide range of neurological applications to detect brain patients' disease by analyzing the special clinical parameters extracted from the images and to provide patient-specific medical care, which can be especially helpful at early stages of the disease. The automatic brain MRI segmentation is performed based on modified pixel classification technique called fuzzy c-means followed by connected component labeling.

Keywords Segmentation · Medical imaging · Fuzzy c-means · Neurological application

A. Goyal (✉) · S. Tirumalasetty · G. Hossain · R. Chaloo
Texas A&M University, Kingsville, TX, USA
e-mail: ayush.goyal@tamuk.edu

D. Bathla · P. Ranjan
Amity University Uttar Pradesh, Noida, UP, India

M. K. Arya · R. Agrawal
G.L. Bajaj Institute of Management and Research, Greater Noida, UP, India

1 Introduction

In the field of medical image processing, the most challenging task to any neurologist or a doctor or a scientist is to detect the patient's disease by analyzing the patient's clinical information. Patient's data is extracted and analyzed to detect the abnormalities and to measure the illness of the disease which helps a medical practitioner to cure the disease at its early stages. Extraction of brain abnormalities in brain MRI images is performed by segmentation of gray and white matter regions in patient's brain MRI images. After segmentation is performed, patient's clinical data such as the area of the cortex, size of tumor, type of tumor (malignant or benign), and position of tumor are determined which help a [1] doctor to take early decisions for surgery or treatment to cure any brain disease.

During initial days, these segmentation techniques were performed manually by subject matter experts or neurological experts, which consumes time and effort of neurological specialists in the field. The segmentation results obtained [2, 3] from the manual segmentation techniques may not be accurate due to vulnerable and unsatisfactory human errors which may lead to inappropriate surgical planning. Therefore, it has become very much necessary for a neurologist or an academician or a researcher to introduce automatic segmentation techniques which give accurate segmentation results. These segmentation techniques that are performed automatically are of two types typically known as semiautomatic and fully automatic segmentation techniques. In a semiautomatic segmentation process, partial segmentation is performed automatically, and then, the results thus obtained are checked by neurological experts to modify for obtaining final segmentation results. In a fully automatic segmentation technique, there is no need for manual checking by neurological experts which minimizes his time and effort. These fully automatic segmentation techniques are classified as threshold-based, region-based, pixel classification based, and model-based techniques which are determined by the computer without any human participation.

In this paper, regions in the brain are segmented automatically using a technique called Fuzzy C-Means (FCM) algorithm, which is a pixel classification technique followed by component labeling technique which is used widely in biomedical image processing to perform fully automatic segmentation in brain MRI images. This clustering mechanism is the most widely used technique for segmentation and detection of tumor, lesions, and other abnormalities in brain MRI scans. The above pixel classification technique gives accurate results especially while analyzing non-homogenous and dissimilar patterns of [4–7] brain MRI images. FCM is a unique method that can be implemented in most of the MRI images to perform segmentation and obtain efficient results even for the noisy MRI images. The main concept of clustering algorithm is grouping of similar components (in this research, it is pixels of an image) within the same cluster. This simple idea is implemented in this work to develop a disease prediction framework that can automatically segment various regions of multidimensional brain MRI scans.

2 Automatic Segmentation

Recent studies have shown that the atrophy rate in the brain is the valid parameter to measure the severity of diseases such as dementia, Alzheimer's and other brain disorders from brain MRI images. Therefore, the necessity to calculate the atrophy rate has been increased which is the measurement of abnormalities in gray and white matter regions in brain MRI images of the patient [8]. The method herein presents a disease prediction framework that can automatically segment gray and white matter regions of patient's brain using modified adapted pixel clustering method. In the proposed method, the gray and white matter regions of cerebral structures are automatically segmented using a form of adaptive modified pixel clustering technique called Fuzzy C-Means (FCM) in which the pixels having similar intensity values are grouped [9] into similar clusters and followed by connected component labeling in which each pixel of gray and white matter regions are labeled.

A. Image Acquisition

The patient's brain MRI image and neurological data used in this research work was obtained from the Image and Data Archive (IDA) powered by Laboratory of Neuro Imaging (LONI) provided by the University of Southern California (USC) and from the Department of Neurosurgery at the All India Institute of Medical Sciences (AIIMS), New Delhi, India. The data was anonymized as well as followed all the ethical guidelines of the participating research institutions.

B. Segmentation Methodology

The segmentation methodology used in this research for automatically performing segmentation of gray and white matter regions in brain MRI images using fuzzy c-means clustering algorithm is shown as a block diagram in Fig. 1. The preliminary step in the process of segmentation is to remove the external sections of the image which is not required for brain MRI image analysis. Therefore, it is necessary to detect and remove the skull outline from the patient's brain MRI image. This mechanism is performed using elliptical Hough transform which is used in digital image processing applications that identify the arbitrary shapes such as circles, ellipses, and lines in an image data. After the skull outline removal, the inner brain slice is subjected to adapted fuzzy c-means clustering algorithm which is one of the pixel classification techniques mentioned above. In this process, the brain internal slice is separated into different regions using clustering mechanism which is based on the intensity values of the pixels in this research.

Among the above-described pixel classification segmentation techniques, clustering-based fuzzy c-means algorithm is used for segmentation of gray and white matter regions in this research. Also, this technique generated accurate, reliable, and robust results even with the noisy MR images of patients' brain. After clustering, the next step in the segmentation process is to perform connected component labeling based on the connectivity of the neighboring pixels. Even after performing clustering, some of the pixels positioning adjacent to each other different similar intensity values of pixels may be in the same clus-

Flowchart of Automatic Segmentation Methodology

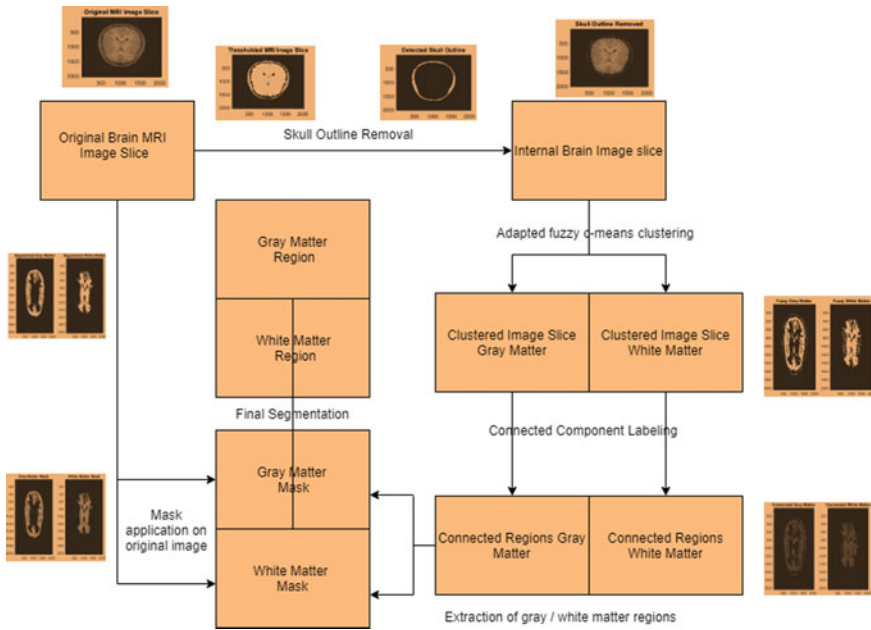


Fig. 1 Block diagram of automatic gray and white matter segmentation

ter. Therefore, it is necessary to perform connected component labeling in which each component of the image is labeled and given a membership to each of the pixels in the clusters to determine and differentiate the pixels accurately. Also, among many other techniques, this modified adaptive pixel classification technique called fuzzy c-means is used for both multi-featured and single featured extraction and analysis using spatial data. The segmentation technique [8, 9] used in this research is fully automatized unsupervised segmentation that can perform feature analysis, clustering in many medical imaging applications. A medical image data is formed with the combination of set of components or data points that have similar or dissimilar parametric values. These similar and dissimilar data points of the image are classified into various similar clusters which can be performed based on similarity criteria. Image pixels of medical image data can be correlated to each other which have similar characteristics or feature information to the data points that are sitting next to the data point in an image. In this segmentation mechanism, spatial data of the adjacent pixels is taken to perform clustering. This research work presents an algorithm for clustering of various regions of brain MRI images into various classes followed by connected component labeling using a knowledge-based algorithm. Below are the steps for fully automatic segmentation algorithm:

- (a) **Skull outline detection:**

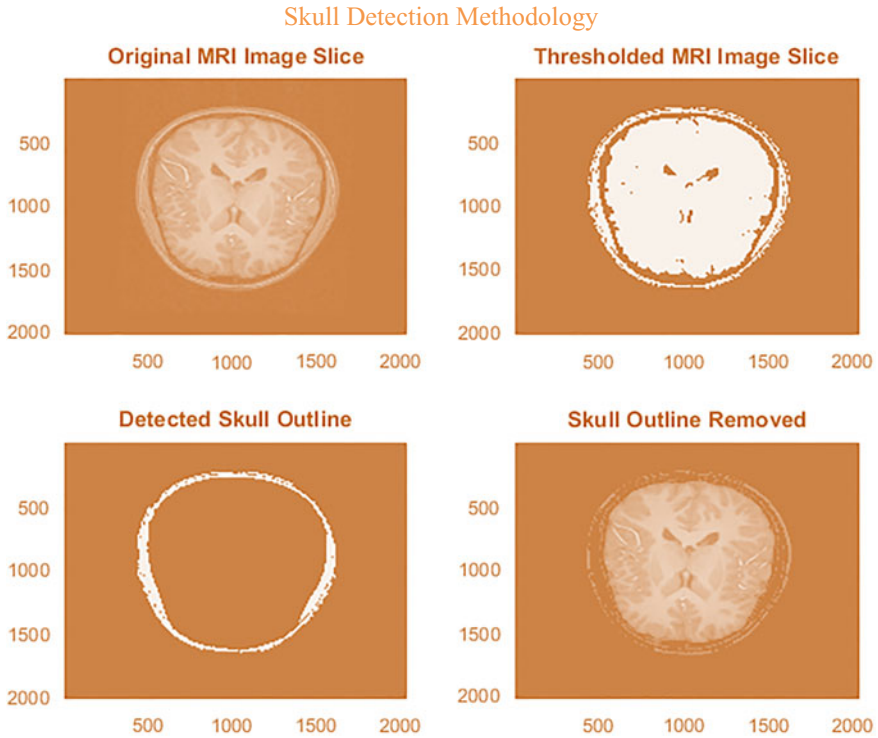


Fig. 2 Skull outline detection in brain MRI images

Skull outline present in the brain MRI scan is not required for analyzing brain abnormalities in this research. Therefore, detection of skull outline and removing it play a vital role during the segmentation process so that feature extraction and analysis becomes easier and results thus obtained without the skull outline part will be accurate. This skull outline sections in brain MRI regions are not our region of interest as this section are filled with fat, skin, and other unwanted materials. This step allows a researcher to focus more on the actual brain sections which are responsible for brain disorders and to obtain reliable outputs [10]. In this skull removal process, a widely used image feature extraction tool in digital image processing is used to detect the superfluous components of the brain MRI image data with in different shapes such as circle, ellipse, and lines. In this research, we have used the elliptical Hough transforms to extract the unwanted material (data objects or pixel components of an MRI image) from the actual brain MRI image data. This elliptical Hough transform is applied to the original brain MRI slice using a voting process in a parametric space [11]. Figure 2 shows the results obtained in the first step of the segmentation process:

- (b) **Adaptive fuzzy c-means clustering:**

Once the skull outline sections are detected and removed from the original brain MRI image, the next and very important step in the segmentation process is to perform clustering to the image that is obtained from the first step of the segmentation process. In this clustering, the medical image is classified into various regions of brain such as gray matter, white matter, and cerebrospinal fluid. The concept of clustering helps a researcher especially in digital image processing technique to classify different patterns of the image and for the segmentation of any medical image data. It is a widely used technique for various purposes for medical data analysis in the field of medical sciences. The process of classifying different clusters by grouping the similar components into same cluster based on some criteria is defined as clustering. In this research, clustering of the medical MRI image data having different regions such as gray matter and white matter is performed based on similar intensity values of the pixels. Due to several internal and external parameters, patient's MRI scans in the field of biomedical sciences may have more noise which when analyzed further may produce inappropriate results [12]. This is highly unacceptable as these inappropriate results may lead to improper diagnosis and surgical planning of the patient. And hence, an effective algorithm is required to avoid inaccurate results during the segmentation process. There are several types of clustering techniques available in the field to perform segmentation of brain MRI images in the medical field. In this research, we have used a modified pixel classification technique called fuzzy c-means, which is based on the clustering mechanism. This technique that is used for segmentation generates accurate results equally for noisy MRI patient data [13–18]. Among many other clustering algorithms, fuzzy c-means algorithm is the most popular technique which has a wide number of benefits comparatively as it performs well even with the uncertain medical image data. This technique used in our research enhances the features of fuzzy c-means algorithm minimizing computational errors during the segmentation process and this modified algorithm is called adaptive fuzzy c-means clustering algorithm [19].

(c) **Connected component labeling:**

The next step in the proposed automatic segmentation is to perform connected component labeling to the clustered image based on pixel connectivity mechanism. In this stage, positions of several pixels which are located on the clustered image are extracted and classified. In this process, several disjoint and also connected components are labeled based on the connectivity procedures, which is a very essential step in the segmentation process in order to reduce inaccuracy [20]. Every medical MRI image consists of pixels that are located side by side sitting together forming connected components will have similar intensity values. Therefore, in this method, the image is scanned such that every pixel is detected and examined to extract the connected pixel regions of the MRI image that are positioned adjacent to each other having similar intensity values [21–25]. Each and every pixel component of the image irrespective of which group it belongs to are labeled based on the connectivity of pixels. In this research, connected component labeling is performed using two-dimensional

eight-connectivity measures to determine the way in which each pixel is related to its neighboring pixel in the medical MRI image.

(d) **Final segmentation mask after removing noise:**

Last but not the least, the step after skull outline detection, clustering, and connected component labeling is to generate the required segmented gray and white matter regions by superimposing gray and white matter masks on the actual brain MRI image that has to be analyzed. Our major goal in this step is to remove all background pixels and to only keep foreground pixels of region of interest of the original MRI image [25–28]. This process of overlaying masks on original brain MRI image and removing the background pixels from the image improves the segmentation process by further increasing the quality of separability of gray and white matter regions, and thus, accurate segmentation results are obtained. The results obtained by the process of clustering using fuzzy c-means followed by connected component labeling to extract gray and white matter regions as masks and when these masks are further processed for final segmentation of gray and white matter regions are shown in Fig. 3.

3 Graphical Computational Tool

A software tool is developed that can automatically perform the entire process of feature extraction, classification, preprocessing, and segmentation as an effective graphical computational tool with a user interface (GUI). This application is independent and a standalone GUI application that can be installed on the users' machine acting as a desktop application. Neurologists or any user can load the brain MRI image from his local machine and perform automatic segmentation to obtain various results instantaneously. This automated segmentation tool can perform segmentation and display the results as mask, color images, or boundaries of gray and white matter regions of brain MRI image with just clicks of buttons that takes very less amount of time and efforts of neurologists. The developed GUI system assists neurologists or any user making it easy to upload patient's brain image from his local computer, viewing and obtaining the results in very less time reducing efforts due to manual tracings [29–33] by the experts. The GUI has the following features:

- (1) Segmentation of brain MRI images is provided as a software.
- (2) It is freely accessible to all researchers in the medical field and neurologists, radiologists, and doctors in any part of the world.
- (3) It is user-friendly and easy to use.
- (4) It automatically segments the brain images and so no manual tracing is required by the user.
- (5) It supports all medical image data types (nifty, dicom, png, etc.).
- (6) Neurologists disease prediction framework is provided in this software tool.
- (7) Automatically calculates the areas of gray and white matter regions or lesions or tumors based on which it predicts disease in brain.

Automatic Gray and White Matter Segmentation Methodology

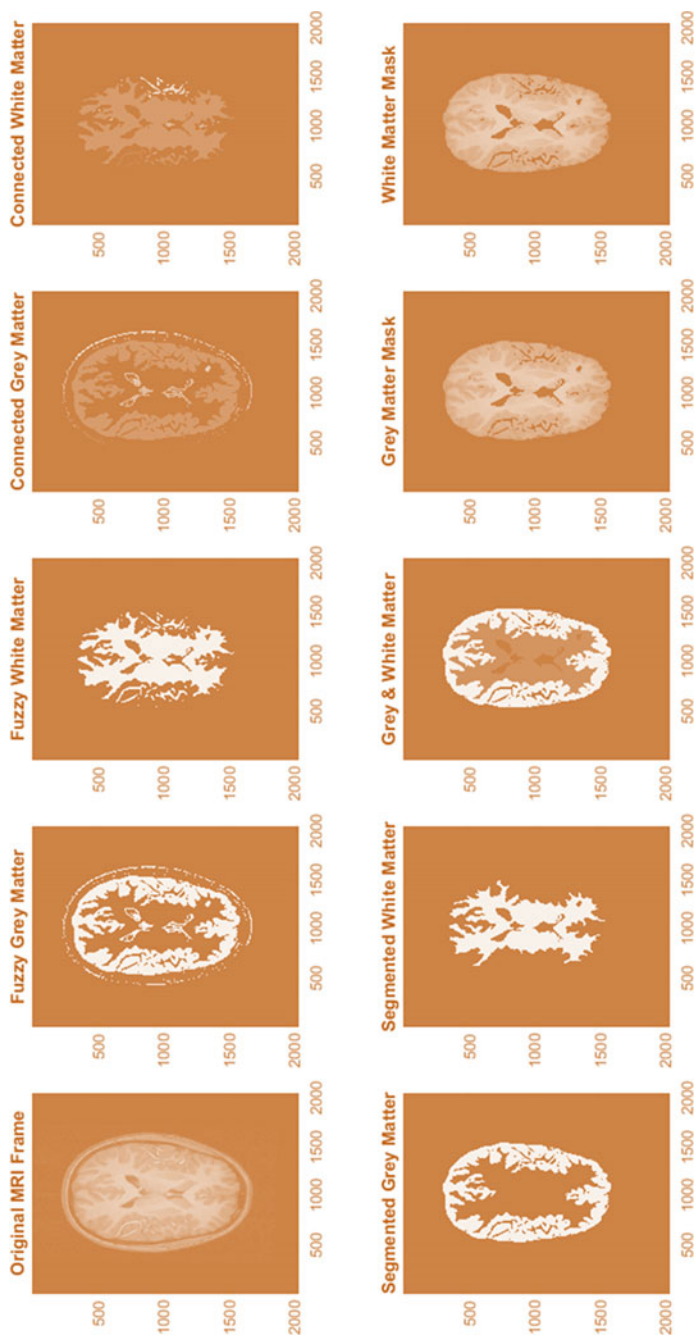


Fig. 3 Gray and white matter segmentation in brain MRI images

GUI of Computational Software for Automatic Brain MRI Segmentation

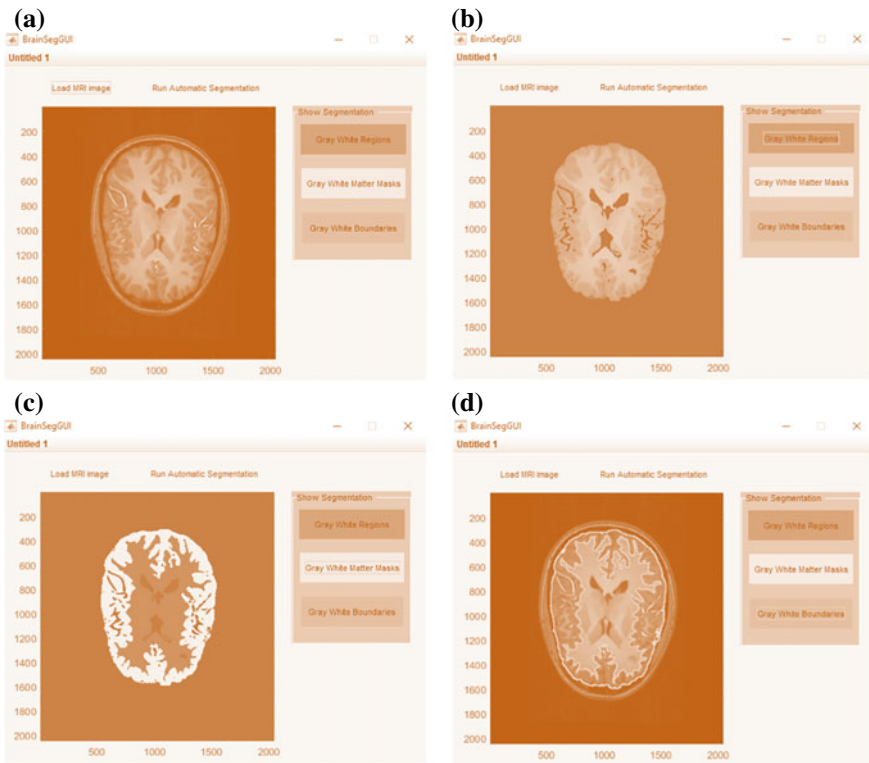


Fig. 4 Graphical User Interface (GUI) of computational brain segmentation software developed to **a** load MRI image, **b** gray and white matter regions, **c** gray and white matter masks, and **d** gray and white matter boundaries

Below are the three screenshots which show running the GUI for loading the brain MRI image (Fig. 4a), viewing the gray and white matter segmented regions (Fig. 4b), viewing the gray and white matter extracted masks (Fig. 4c) and viewing the gray and white matter region boundaries (Fig. 4d).

4 Manual Segmentation

In this research work, the manual segmentation is performed for several patients' MRI image data to validate and verify the automatic segmentation techniques using fuzzy c-means followed by connected component labeling performed in this research with the manual segmentations done by neurological tracings of gray matter and white matter regions by experts [34–38]. Figure 5 presents the manual segmentations

Manual Segmentation Illustration

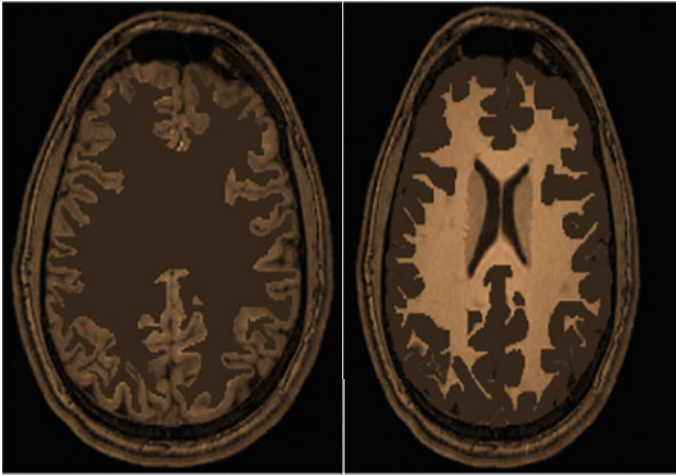


Fig. 5 Manual segmentation (labeling) by neurologist expert of the gray and white matter regions in brain MRI images. Gray matter region (left) and white matter region (right)

performed by neurological experts to segment gray and white matter regions of brain MRI images.

5 Validation

This research work presents the comparison of manual and automatic segmentation results of five different patients' sample MRI images. To compare these results, a statistical measure called Dice coefficients is used to calculate the similarity measures of these two techniques. Figure 6a–c shows the sample manual and automatic segmentation of three of the patients. The automatic segmentations are obtained from the proposed algorithm, and the manual segmentation is obtained here from the neurological tracings by experts in this research. For the validation purpose, five different sample image data are considered, and the manual and automatic segmentations are performed for the [39–42] same to compare both the segmentation results that are obtained. For each of the patient MRI images, a dice coefficient value is calculated between manual and automatic segmentation of patient brain MRI images. Manual segmentation is performed three times by neurological experts for each of the sample patient images among the five different MRI images. Finally, the Dice coefficient values are plotted using box plots for each of the patient brain images that compare manual and automatic segmentations for all sample patient images considered. Figure 7 shows the box plots of the Dice coefficients calculated as the similarity

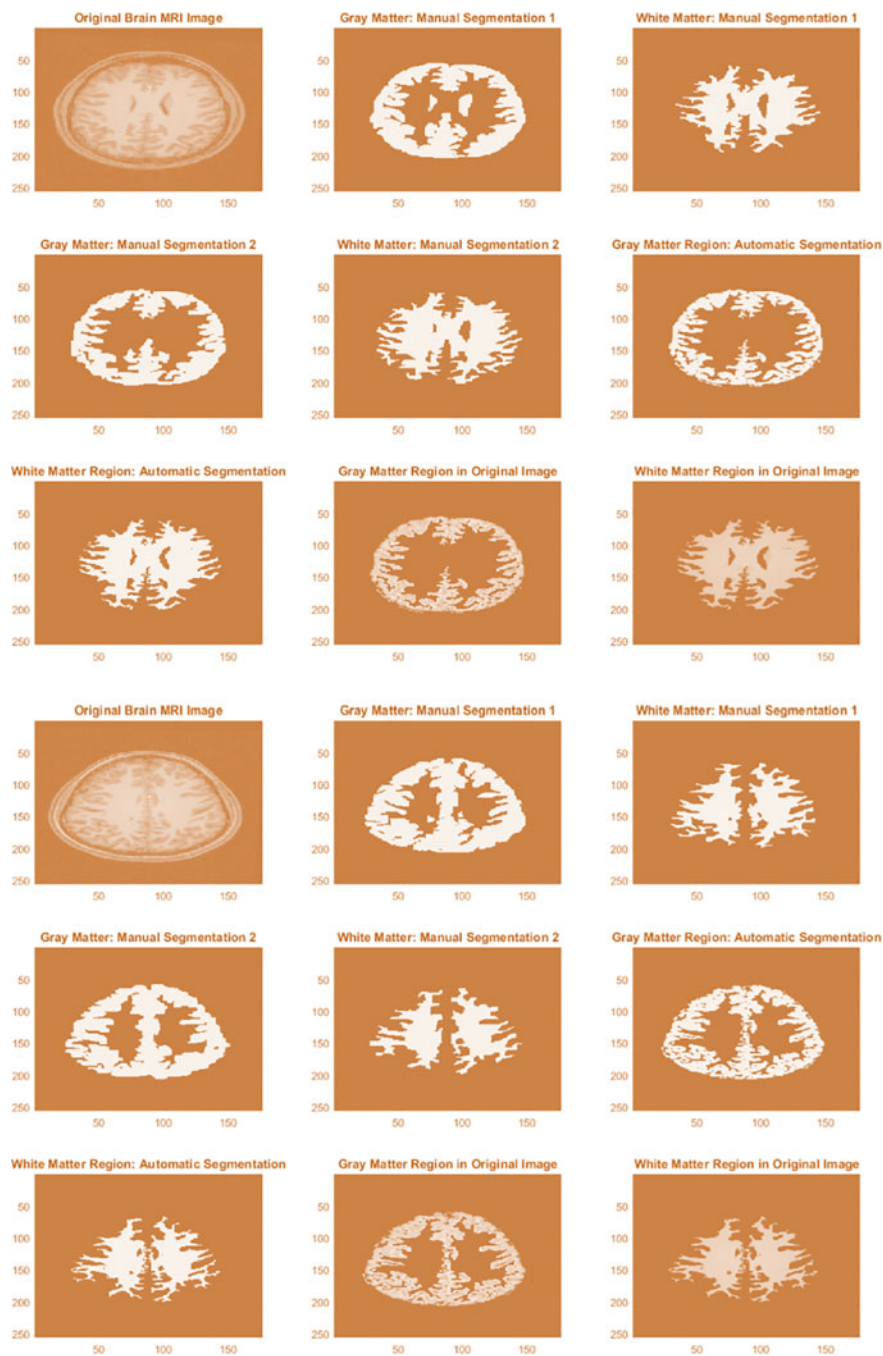


Fig. 6 Sample manual and automatic segmentation of three specimens

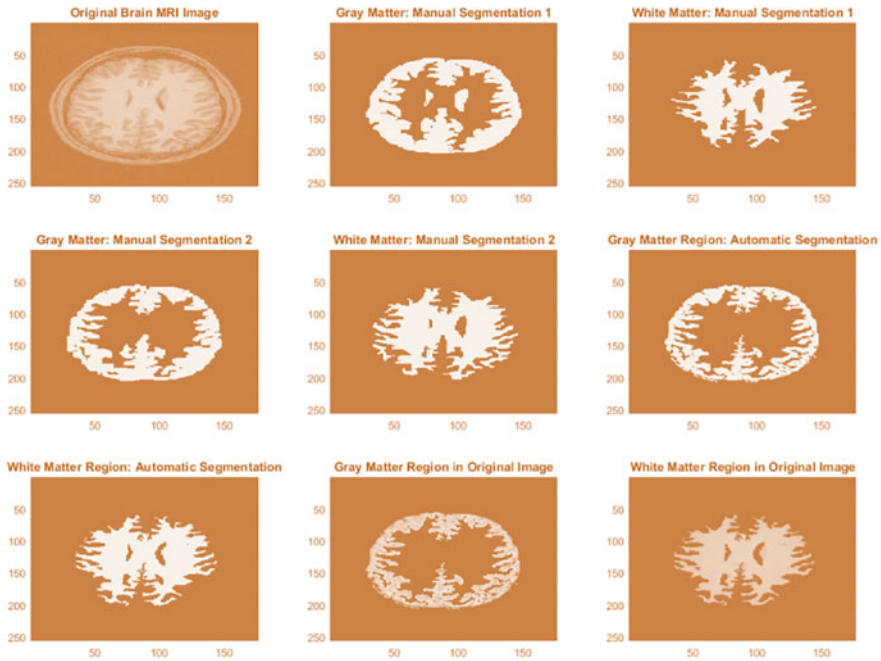


Fig. 6 (continued)

measure to compare manual and automatic segmentation of the brain MRI images for the five sample patients.

6 Discussion and Conclusion

The research presented in this work facilitates efficient and effective automatic segmentation of gray and white matter regions from brain MRI images, which has several clinical neurological applications. A fully automatic segmentation methodology using elliptical Hough transform along with pixel intensity and membership-based adapted fuzzy c -means clustering followed by connected component labeling and region analysis has been implemented in this research to perform segmentation of gray and white matter regions in brain MRI images. The algorithm was tested and verified for several sample brain MRI images. Manual segmentations were performed by neurological experts for several patient brain MRI images. These manual segmentations were used to compare and validate with the results obtained from the automatic segmentations in this research work. Validations were performed by calculating several Dice coefficient values between the automatic segmentation results and the manual segmentation results. The Dice coefficient values are similarity measures

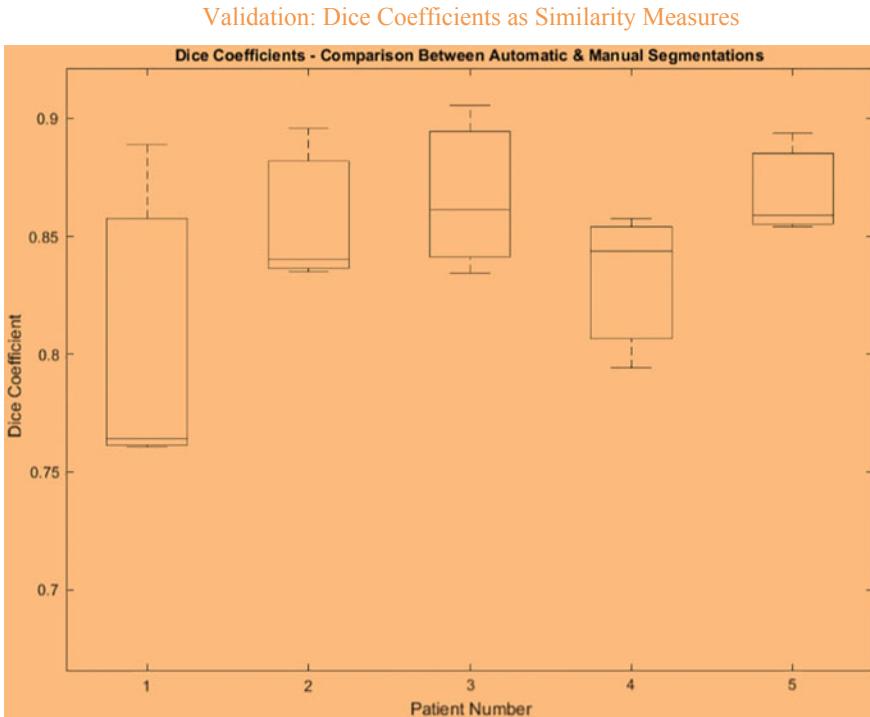


Fig. 7 Box plots for Dice coefficients to compare manual and automatic segmentation of brain MRI images of five patients

that are represented statistically using box plots in this research work. The automated computational segmentation tool developed in this research can be employed in hospitals and neurology divisions as a computational software platform for assisting neurologist in detection of disease from brain MRI images post MRI segmentation. This tool obviates manual tracing and saves the precious time of neurologists or radiologists. This research presented herein is foundational to a neurological disease prediction and disease detection framework, which in the future, with further research work, can be developed and implemented with a machine learning model-based prediction algorithm to detect and calculate the severity level of the disease, based on the gray and white matter region segmentations and estimated gray and white matter ratios to the total cortical matter, as outlined in this research.

References

1. Shasidhar, M., Raja, V. S. and Kumar, B. V., (June 2011). MRI brain image segmentation using modified fuzzy c-means clustering algorithm. In *2011 International Conference on Communi-*

- cation Systems and Network Technologies (CSNT)* (pp. 473–478).
2. Despotović, I., Goossens, B., Philips, W. (2015) MRI segmentation of the human brain: challenges, methods, and applications. *Computational and mathematical methods in medicine*, 2015.
 3. Gordillo, N., Montseny, E., & Sobrevilla, P. (2013). State of the art survey on MRI brain tumor segmentation. *Magnetic Resonance Imaging*, 31(8), 1426–1438.
 4. Suhag, S., & Saini, L. M. (May 2015). Automatic detection of brain tumor by image processing in matlab. In *SARC-IRF International Conference*.
 5. Hassan, E., & Aboshgifa, A. (2015). Detecting brain tumour from MRI image using matlab gui programme. *International Journal of Computer Science & Engineering Survey (IJCSSES)* 6(6).
 6. Sharma, N., & Aggarwal, L. M. (2010). Automated medical image segmentation techniques. *Journal of medical physics/Association of Medical Physicists of India*, 35(1), 3.
 7. Tsai, C., Manjunath, B. S., & Jagadeesan, B. (1995). Automated segmentation of brain MR images. *Pattern Recognition*, 28(12), 1825–1837.
 8. Despotović, I., Goossens, B., & Philips, W. (2015). MRI segmentation of the human brain: challenges, methods, and applications. *Computational and mathematical methods in medicine*, 2015.
 9. Chuang, K. S., Tzeng, H. L., Chen, S., Wu, J., & Chen, T. J. (2006). Fuzzy c-means clustering with spatial information for image segmentation. *Computerized medical imaging and graphics*, 30(1), pp. 9–15.
 10. Mokbel, H. A., Morsy, M. E. S., & Abou-Chadi, F. E. Z. (2000). Automatic segmentation and labeling of human brain tissue from MR images. In *17th NRSC'2000. Seventeenth National Radio Science Conference, 2000* (pp. K2–1). IEEE.
 11. Antolovic, D., (2008). Review of the Hough transform method, with an implementation of the fast Hough variant for line detection. *Department of Computer Science, Indiana University*.
 12. Kumar, N., & Nachamai, M. Noise Removal and filtering techniques used in medical images. *Oriental Journal of Computer Science and Tachnology* 10(1).
 13. Wang, H. R., Yang, J. L., Sun, H. J., Chen, D., & Liu, X. L. (August 2011). An improved region growing method for medical image selection and evaluation based on Canny edge detection. In *2011 International Conference on Management and Service Science (MASS)* (pp. 1–4). IEEE.
 14. Mubarak, D. M. N., Sathik, M. M., Beevi, S. Z., & Revathy, K. (2012). A hybrid region growing algorithm for medical image segmentation. *International Journal of Computer Science & Information Technology*, 4(3), 61.
 15. Wong, K. K., Tu, J., Kelso, R. M., Worthley, S. G., Sanders, P., Mazumdar, J., et al. (2010). Cardiac flow component analysis. *Medical Engineering & Physics*, 32(2), 174–188.
 16. Zanaly, E. A. (2013). An Approach based on fusion concepts for improving brain magnetic resonance images (MRIs) segmentation. *Journal of Medical Imaging and Health Informatics*, 3(1), 30–37.
 17. Zanaly, E. A., & Ghiduk, A. S. (2013). A novel approach for medical image segmentation based on genetic and seed region growing algorithms. *Journal of Computer Science and Information Systems ComSIS*, 10(3), 1319–1342.
 18. Zanaly, E. A., & Afifi, A. (2013). A watershed approach for improving medical image segmentation. *Computer methods in biomechanics and biomedical engineering*, 16(12), 1262–1272.
 19. Zanaly, E. A. (2013). An adaptive fuzzy C-means algorithm for improving MRI segmentation. *Open Journal of Medical Imaging*, 3(04), 125.
 20. [Online] Available: https://en.wikipedia.org/wiki/Connected-component_labeling [Accessed November 9, 2017].
 21. Wu, K., Otoo, E., & Shoshani, A. (2005). Optimizing connected component labeling algorithms. *Lawrence Berkeley National Laboratory*.
 22. Suzuki, K., Horiba, I., & Sugie, N. (2003). Linear-time connected-component labeling based on sequential local operations. *Computer Vision and Image Understanding*, 89(1), 1–23.
 23. Goyal, A., Lee, J., Lamata, P., van den Wijngaard, J., van Horsen, P., Spaan, J., et al. (2013). Model-based vasculature extraction from optical fluorescence cryomicrotome images. *IEEE Transactions on Medical Imaging*, 32(1), 56–72.

24. Sikarwar, B. S., Roy, M. K., Ranjan, P., & Goyal, A. (2016). Automatic Disease Screening Method Using Image Processing for Dried Blood Microfluidic Drop Stain Pattern Recognition. *Journal of Medical Engineering & Technology*, 40(5), 245–254.
25. Sikarwar, B. S., Roy, M. K., Ranjan, P., & Goyal, A. (2016). Imaging-based method for precursors of impending disease from blood traces. In *Advances in Intelligent Systems and Computing* (Vol. 468, pp. 411–424). Springer.
26. Sikarwar, B. S., Roy, M. K., Ranjan, P., & Goyal, A. (2015). Automatic pattern recognition for detection of disease from blood drop stain obtained with microfluidic device. In *Advances in Intelligent Systems and Computing* (Vol. 425, pp. 655–667). Springer.
27. Bhan, A., Bathla, D., & Goyal, A. (2016). Patient-specific cardiac computational modeling based on left ventricle segmentation from magnetic resonance images. *Advances in Intelligent Systems and Computing* (Vol. 469, pp. 179–187). Springer.
28. Ray, V., & Goyal, A. (2015) Automatic left ventricle segmentation in cardiac MRI images using a membership clustering and heuristic region-based pixel classification approach. In *Advances in Intelligent Systems and Computing* (Vol. 425, pp. 615–623). Springer.
29. Chhabra, M., & Goyal, A. (2017) Accurate and robust iris recognition using modified classical hough transform. In *Lecture Notes in Networks and Systems* (Vol. 10, pp. 493–507). Springer.
30. Goyal, A., & Ray, V. (2015). Belongingness clustering and region labeling based pixel classification for automatic left ventricle segmentation in cardiac MRI images. *Translational Biomedicine*, 6(3).
31. Goyal, A., Roy, M., Gupta, P., Dutta, M. K., Singh, S., & Garg, V. (2015) Automatic detection of mycobacterium tuberculosis in stained sputum and urine smear images. *Archives of Clinical Microbiology*, 6(3).
32. Bhan, A., Goyal, A., Chauhan, N., & Wang, C.W. (2016) Feature line profile based automatic detection of dental caries in bitewing radiography. In: *International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE)*, pp. 635–640, IEEE.
33. Bhan, A., Goyal, A., Dutta, M. K., Riha, K., Omran, Y. Image-Based Pixel Clustering and Connected Component Labeling in Left Ventricle Segmentation of Cardiac MR Images. In *7th International Congress on Ultra-Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pp. 339–342, IEEE, 2015.
34. Ray, V., & Goyal, A. (2015). Image-Based fuzzy c-means clustering and connected component labeling subsecond fast fully automatic complete cardiac cycle left ventricle segmentation in multi frame cardiac MRI images. In *International Conference on Systems in Medicine and Biology (ICSMB)*, IEEE.
35. Goyal, A., van den Wijngaard, J., van Horsen, P., Grau, V., Spaan, J., & Smith, N. (2009). Intramural spatial variation of optical tissue properties measured with fluorescence microsphere images of porcine cardiac tissue. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1408–1411.
36. Sharma, P., Sharma, S., Goyal, A. (2016) An MSE (mean square error) based analysis of deconvolution techniques used for deblurring/restoration of MRI and CT Images. In *2nd International Conference on Information and Communication Technology for Competitive Strategies (ICTCS-2016)*, March 04–05, 2016, Udaipur, India, Conference Proceedings by ACM—ICPS Proceedings Vol. ISBN 978-1-4503-3962-9/16/03, <http://dx.doi.org/10.1145/2905055.2905257>.
37. Goyal, A., Bathla, D., Sharma, P., Sahay, M., & Sood, S. (2016). MRI image based patient specific computational model reconstruction of the left ventricle cavity and myocardium. In *2016 International Conference on Computing, Communication and Automation (ICCCA)*, pp. 1065–1068, IEEE.
38. Duta, M., Thiyagalingam, J., Trefethen, A., Goyal, A., Grau, V., & Smith, N. (2010) Parallel simulation for parameter estimation of optical tissue properties. In *Euro-Par 2010-Parallel Processing* (pp. 51–62).
39. Atkins, M. S., & Mackiewicz, B. T. (1998). Fully automatic segmentation of the brain in MRI. *IEEE Transactions on Medical Imaging*, 17(1), 98–107.

40. Wagner, M., Yang, P., Schafer, S., Strother, C., & Mistretta, C. (2015). Noise reduction for curve-linear structures in real time fluoroscopy applications using directional binary masks. *Medical Physics*, 42(8), 4645–4653.
41. Meijs, M., Patel, A., Leemput, S. C., Prokop, M., Dijk, E. J., Leeuw, F. E., et al. (2017). Robust segmentation of the full cerebral vasculature in 4D CT of suspected stroke patients. *Scientific reports*, 7(1), 15622.
42. Bhan, A., Goyal, A., & Ray, V. (2015) Fast fully automatic multiframe segmentation of left ventricle in cardiac mri images using local adaptive k-means clustering and connected component labeling. In *2nd International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 114–119, IEEE.

Developing Ontology for Smart Irrigation of Vineyards



Archana Chougule  and Debajyoti Mukhopadhyay 

Abstract As the availability of groundwater is getting reduced these years, proper scheduling of irrigation is very important for survival and improvement of vineyards in the southern part of India. Well calculated irrigation scheduling also improves quality of grapes. The knowledge about irrigation to vineyards is available in various documents, and it is scattered. This knowledge can be made available to grape growers through computer-based applications. As Semantic Web is playing an important role, information sharing among varying automated systems, extraction of information from such documents and representing it as ontology is a good idea. This paper presents techniques for automated extraction of knowledge from text resources using natural language processing technique for building vineyard ontology. Smart irrigation system can be developed using IoT sensors and other ICT devices. The paper explains ontology built for resources used under smart irrigation system. It suggests how smart irrigation systems can be built by grape growers by utilizing vineyard ontology and smart irrigation ontology.

Keywords Ontology building · Irrigation scheduling · Grapes · Knowledge base · Natural language processing

1 Introduction

Water is the most important resource in agriculture. Irrigation scheduling is basically deciding on the frequency and duration of water supply to any crop. Proper scheduling of irrigation can minimize wasting water and can also help to improve the quality of grapes. There are different methods of irrigation. Drip irrigation is used by almost all vineyards in India. Manerajuri is a village in India, which is famous for grapes, but

A. Chougule (✉)

Walchand College of Engineering, Sangli, India
e-mail: chouguleab@gmail.com

D. Mukhopadhyay

NHITM, Mumbai University, Mumbai, India
e-mail: debajyoti.mukhopadhyay@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_2

more than 10,000 bore wells have run dry in year 2016. This fact motivated authors to develop the knowledge base and decision support system which can be used for micromanagement of available water.

Knowledge base generation for scheduling water supply to grapes in hot tropical region in India will help automation of water scheduling. Good irrigation practices in ontology will be good education material for farmers. Structured representation of knowledge is more useful than unstructured one. Ontologies can be used for structured representation of concepts in any domain. Along with concepts, relationships between concepts can also be mentioned using ontology. Maintenance and sharing of information can be facilitated using ontologies. Manual construction of domain ontology is time-consuming and error-prone. Semi-automated approach will help in enriching ontology building, reducing required time and enhancing the quality of built ontology. For IoT-based automation of irrigation systems, formal representation and management of IoT techniques and devices used important.

Automated irrigation system needs information like ambient temperature and humidity, atmospheric pressure, soil temperature, soil moisture and leaf wetness. System can be built with such information using sensors, actuators, monitors, collectors and transmitters. Data generated and required from such varying sources can be easily integrated using ontology [1].

This paper describes building ontology for developing IoT-based automated irrigation systems. It details mechanism used for constructing ontology. Two types of ontologies, namely, vineyard irrigation ontology and smart irrigation ontology, are described. These ontologies can be used for generating knowledge base and establishment of automated irrigation system based on available resources. It can also be used to educate grape growers about principles of vineyard irrigation.

2 Research Method

Building ontology consists of five basic steps as defining domain, scope and objective of ontology, listing important keywords from the domain, finding hierarchies between keywords, defining relations between them and the last step is to add those keywords as concepts in ontology and define relationships between concepts. Various formats for building ontology are available as RDF, RDF-S, OWL and OWL-DL. Here ontology is developed in Web Ontology Language (OWL) format. Classes, individuals, data properties, object properties and axioms are the main parts of OWL ontology. Classes represent core concepts in the domain. Individuals are instances of any specific class. Data properties are attributes of a class with specific values and object properties hold values as objects of some other class. Two ontologies are maintained separately as vineyard ontology and smart irrigation ontology.

2.1 Building Vineyard Irrigation Ontology

Water management is also important to maintain yield and quality of grapes. This section details how water management knowledge is represented in ontology form. Knowledge about irrigation scheduling of grapes is available in various documents published by researchers from organizations like National Research Center for Grapes. To make use of this knowledge in decision support system, it must be transformed to accessible form. As mentioned earlier, knowledge stored in ontology can be accessed and shared by automated systems. This section discusses converting the irrigation scheduling details available in text documents to ontology. As a reference ontology, AGROVOC [2] is used. AGROVOC is an agricultural vocabulary available as ontology. It is available in resource description framework (RDF) format.

The documents for irrigation scheduling are taken as input and the most relevant documents are only considered. Relevancy of documents is ranked using TF-IDF [3] algorithm. TF-IDF stands for term frequency, inverse document frequency. List of keyword related to agriculture taken from agriculture experts is considered as input for this purpose. Term frequency is considered as count of word occurring in document and inverse document frequency is the count of documents containing the word versus the total number of documents. Natural language processing techniques are used then to extract important keywords from the text. The text is converted to a number of tokens using StringTokenizer. The text is broken into words, phrases and symbols under tokenization. Tokenization helps in finding meaningful keywords from the text.

After tokenization, stop words are removed from the list of tokens. Stop words are words which are used for joining words together in a sentence. They occur very frequently in documents, but are of no importance for ontology building. Stop words like 'and', 'or', 'this' and 'the' are not used for the classification of documents, so they are removed. The process of conflating the variant forms of a word into a common representation is called stemming. There can be the same words with different forms. Such words are converted to common form using Porter's stemmer [4]. For example, the words: 'irrigation', 'irrigated' and 'irrigating' are reduced to a common representation 'irrigate'. OpenNLP [5] POS tagger is used for extracting important keywords from tokens. POS tagger assigns labels to keywords as NN for noun and NNP as proper nouns. Assigned labels are then considered for probable classes, data properties, individuals or object properties.

Along with concepts relationships between concepts are also found by processing sentences. Formal concept analysis [6] is used for deciding hierarchy among concepts. Formal concept analysis is a technique used for defining conceptual structures among data sets [7]. Extracted individuals, data properties and object properties are used for this purpose. The top-down approach is used here. Extracted concepts are stored in the concept table. Extracted relations are used for defining hierarchy among classes and among properties. For example, Table 1 has five concepts and four attributes, showing which concepts carry which corresponding attributes, and Fig. 1 shows corresponding concept lattice.

Table 1 Example of formal concept analysis

	Amount	Unit	Timestamp	Level
Sensor		X	X	
Irrigation	X	X	X	
Groundwater	X	X	X	X
Surface water			X	X
Soil moisture		X	X	X

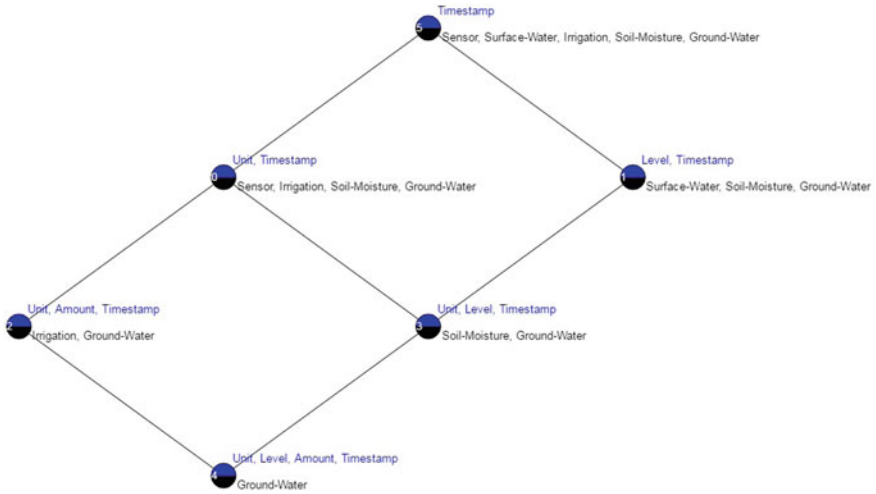


Fig. 1 Concept lattice corresponding to context in Table 1

The ontology is built using ProtegeOWL APIs [8] available in java language. The fundamental concepts related to irrigation are added as classes in ontology. The ontology is built using the top-down approach. More general term is added at higher level in the hierarchy, followed by more specific concepts at lower levels.

Vineyard ontology contains 16 classes, 56 individuals, 12 object properties and 19 data properties. The built ontology is further edited using Protégé 5.1 ontology editor. Protégé editor has facilities to add, edit and delete concepts, change hierarchy, view ontology and use reasoners on ontology. The irrigation knowledge is extracted from research documents published by national research for grapes, Pune, India [9]. Irrigation schedule can be decided based on knowledge stored in vineyard irrigation ontology in terms of individuals, object properties and data property values. Requirements for irrigation for grapes are mentioned as individuals of Irrigation-Schedule class. Under each object, water requirement is specified as data property value. Growth stage-wise water requirements of grapes are considered and mentioned for each irrigation schedule.

Figure 2 shows classes under vineyard ontology. Water-Shortage-Symptoms class stores all symptoms shown on grape leaves, stems, berries and in soil due to the

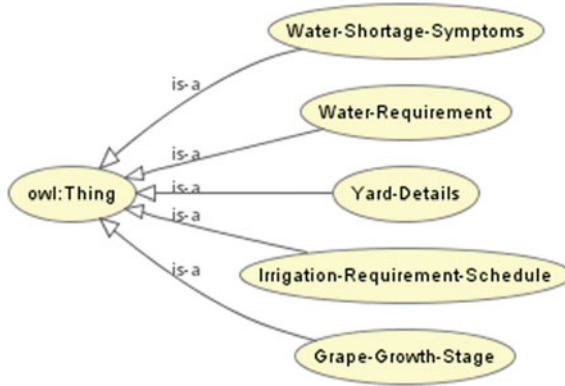


Fig. 2 Classes in vineyard ontology

Fig. 3 Example of irrigation schedule defined in vineyard ontology as data and object property assertions

shortage of water. Specific information about yard like root depth, soil type and available water resources are covered by Yard-Details class. All stages of vine growth are stored as individuals of Grape-Growth-Stage class and are referenced by irrigation requirement schedule.

Irrigation-Requirement-Schedule is the most important class for modelling and predicting irrigation requirements. It contains water requirement schedules as individuals detailing days after pruning of schedule, water requirement quantity in litres per day per hectare, growth stage of vines, irrigation period in terms of month and evapotranspiration measure. Figure 3 shows the example of irrigation schedule defined in vineyard ontology.

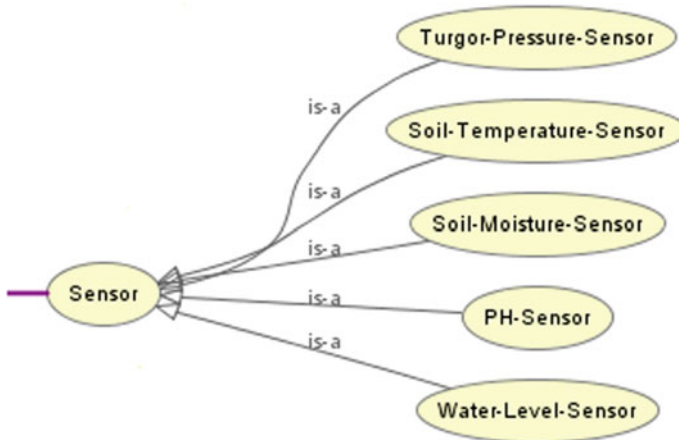


Fig. 4 Sensor class in smart irrigation ontology

2.2 Smart Irrigation Ontology

In order to provide a knowledge base of tools and techniques that can be used for automation of irrigation scheduling, Smart Irrigation ontology is built. It contains all concepts including IoT, for creating and managing smart irrigation system for vineyard. The time and amount of water to apply to vineyard are generally based on four methods as monitoring of soil moisture levels, measuring water status in plants, based on evapotranspiration measurement and based on the estimated vineyard water use rate and soil water storage. As irrigation decision depends on various factors in vineyard, sensors are used for such measurements. Figure 4 shows types of sensors that can be used for smart irrigation. Unit-of-measure, type-of-sensing, locating-of-sensing and sensor-description are data properties of the Sensor class. The ontology provides all support for storing information about IoT-based tools. It also provides data and object properties for storing measurements read by such tools. It comes under devices and communication techniques classes and their sub-classes. Knowledge about irrigation techniques that can be used under vineyards and possible water resources comes under Irrigation-Techniques and Water-Resources classes. Observations class contains sub-classes, data and object properties for storing measurements taken by sensors. System-Design and User-Interface are the classes suggesting options for building irrigation automation system using available resources. As irrigation management depends on climate and weather details in specific region two classes, namely, Climate and Weather-Details, are added in the ontology. Annual precipitation and annual rainfall are data properties of climate class. Rainfall, humidity and temperature are parts of Weather-Details class. Figure 5 shows all classes under smart irrigation ontology.

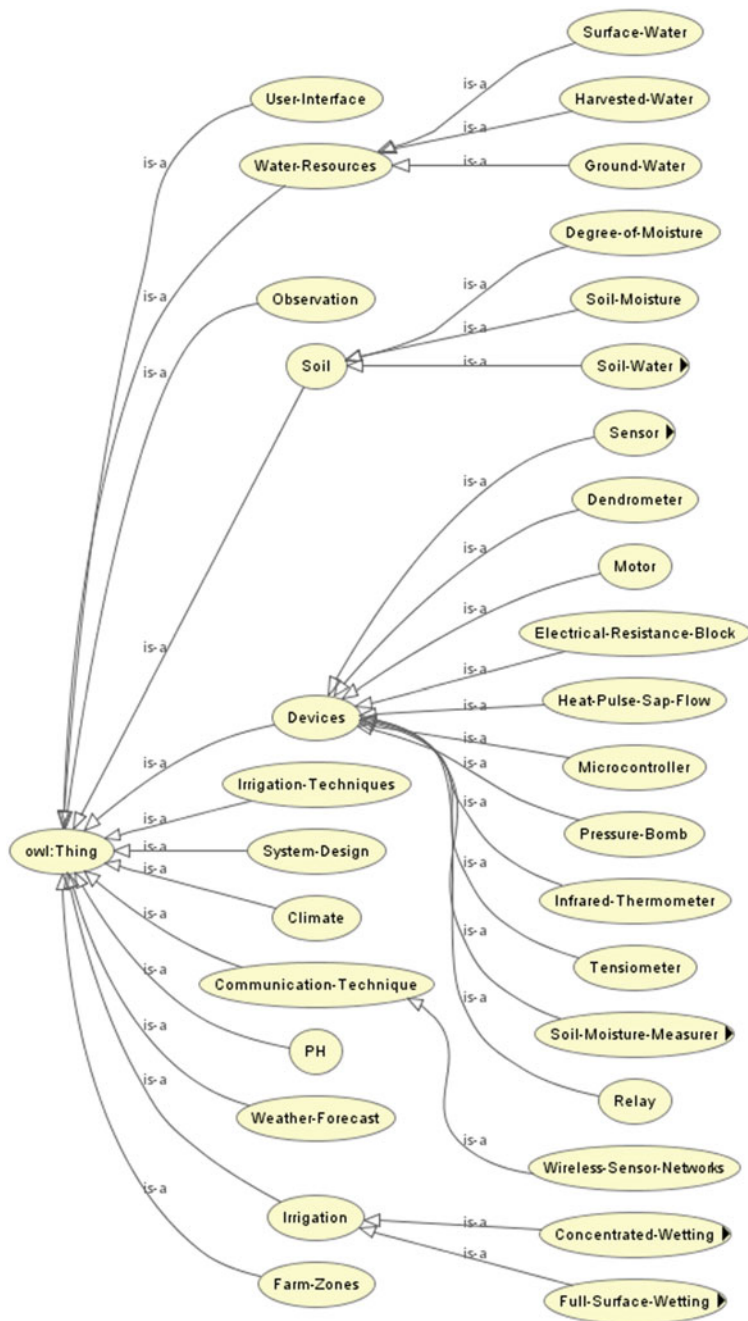


Fig. 5 Classes in smart irrigation ontology

3 Conclusion

Ontology plays an important role in IoT-based automation of agricultural systems. The paper demonstrated how ontology can be built by using natural language processing techniques and formal concept analysis. Paper demonstrated how irrigation details about vineyards can be very well represented using ontology. The information about sensors can also be stored in ontology and used for automated systems building, which is shown in smart irrigation ontology. Based on given water requirements, the automated irrigation system can be built using knowledge from vineyard ontology and smart irrigation ontology as proposed in the paper.

References

1. Cornejo, C., Beck, H. W., Haman, D. Z., & Zazueta, F. S. (2005). Development and application of an irrigation ontology. *2005 EFITA/WCCA Joint Congress on IT and Agriculture*, Vila Real, Portugal, 25–28 July 2005.
2. AGROVOC Thesaurus. <http://aims.fao.org/agrovoc#.VF29AvmUc2U>.
3. TFIDF Algorithm. <http://en.wikipedia.org/wiki/Tf-idf>.
4. Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3), 130–137.
5. OpenNLP. <https://opennlp.apache.org/>.
6. Obitko, M., Snasel, V., Smid, J. (2004). Ontology design with formal concept analysis. In V. Snasel & R. Belohlavek (Eds.), *CLA 2004*, pp. 111–119, ISBN 80-248-0597-9. VSB—Technical University of Ostrava, Department of Computer Science.
7. Ganter, B., & Wille, R. (1999). *Formal concept analysis, mathematical foundation*. Berlin: Springer Verlag.
8. Protege-OWL API Programmer's Guide. https://protegewiki.stanford.edu/wiki/ProtegeOWL_API_Programmers_Guide [Accessed on 17 May 2017].
9. National Research Centre for Grapes. <http://nrcgrapes.nic.in>.

An Intensive Review of Data Replication Algorithms for Cloud Systems



Chetna Dabas and Juhi Aggarwal

Abstract Cloud computing is a technological paradigm that facilitates a universal platform for hosting and accessing services and data by means of a large number of shared resources by an equally large number of users across the globe. Since a large amount of data is involved, it becomes crucial to make the data available to the users on the go without compromising with the integrity of data. Data replication is used to make copies of the data and files such that they can be accessed without delay and also act as a backup in case of any failure. Here we have discussed crucial data replication algorithms, namely, adaptive data replication strategy (ADRS), high QoS-first replication (HQFR), build time algorithm, cost-effective incremental replication strategy (CIR) and dynamic replica control strategy (DRCS).

Keywords Quality of service · Hadoop file system (HDFS) · Replication factor

1 Introduction

Cloud has become increasingly popular as a data storage unit as many business organizations and companies have decided to store their data in cloud data centres. Some of the well-known cloud storage providers are the Amazon S3 [1], Google cloud storage [2] and Microsoft Azure [3]. These storage providers meet the needs of the companies by securely storing and analysing data while providing 99.9999% durability in case of Amazon S3. The benefit of using such a system for storing data is that data can be retrieved from anywhere at any time with help of a web application or mobile phone. The services provided by these cloud service providers include Software as a Service (SaaS), Platform as Service (PaaS) and Infrastructure as a Service (IaaS). These services are provided as pay per use services, i.e. the customer pays according to the amount of resources she/he uses of the cloud. Some of the resources which are used by the client are bandwidth, storage space, networks and servers, to name a few.

C. Dabas (✉) · J. Aggarwal
Department of CSE, Jaypee Institute of Information Technology, Noida, India
e-mail: chetna.dabas@jiit.ac.in

The cloud infrastructure faces many challenges owing to a lot of factors such as a large number of users, highly dispersed data centres, large numbers of data access requests, heterogeneity of data and load balancing, among others [4]. Secure storage of data is one of the major challenges faced by the cloud environment. Since the data is stored in the public cloud, issues like data integrity, data confidentiality and data privacy are of great concern. To deal with the aforementioned issues, cloud platforms use data encryption algorithms to encrypt the data before storing it in cloud as is done in [5] using Shamir's secret sharing algorithm. Data availability and reliability challenges are also related to cloud storage. Data availability means that the data should be available with its agreed quality of service requirements irrespective of any failure. Data reliability considers that the data is complete and error-free whenever the client retrieves it. These issues of data availability and reliability are well handled by two schemes: (1) erasure coding and (2) data replication. Erasure coding [6, 7] partitions a data entity into k uniformly sized chunks in addition to m extra chunks to hold the parity bit. The total ($k + m = n$) chunks are placed on different nodes such that in case of failure data can be recovered using these parity bits.

Data replication improves data availability and performance of the cloud storage system by serving data request by various replicas placed at different locations either at the same site or different sites. Replicating data onto different sites also provides as a means for fault tolerance. Failures can be of the type disk failure, site failure or node failure. Data replication provides the benefits that there is more access to the shared data. Since availability is improved therefore the waiting times are reduced, and hence, the latency is also reduced. For data replication, there are several factors which need to be considered like how many copies of a data item should be maintained, how often should the replication process be called, in case of failure from where will the data be retrieved, for how long should the copy of a data unit be maintained. The main objective of this entire data replication scheme should be to enhance the performance while keeping the number of replicas as minimum, as more number of replicas will increase the storage overhead.

In this work, we have analysed five different data replication algorithms for replica creation, replica placement, replica replacement and replica management. Following are algorithms that we have reviewed (1) adaptive data replication strategy (ADRS), (2) high quality of service-first replication (HQFR), (3) build time algorithm, (4) cost-effective incremental replication strategy (CIR) and (5) dynamic replica control strategy (DRCS). These algorithms take different approaches for generating best replicas while optimizing the time constraints.

2 Related Work

Data reliability has been explored in various literatures [8, 9] which use concepts like Markov models, Bayesian approaches, Monte Carlo method and graph theory for modelling service reliability. However, in [8], the optimal resource scheduling

strategies have been missed out for multiple computing intensive services. In [9], load influence of the nodes has not been performed. In [10], however, data reliability is examined in terms of variable disk failure rate. Herein, a proactive replica checking for reliability (PRCR) strategy is applied such that reliability is achieved by minimizing the number of replicas. This algorithm provides a reasonable solution for creating minimum replicas as opposed to the traditional three-replica strategy used by Hadoop file system [11]. In [10] the concept of location replicas needs to be further addressed. The reliability of cloud storage over a period of time is expressed as an exponential distribution with the failure rate λ which has its own disadvantages. It is assumed that the reliability of data adopts memory-less property, i.e. the reliability of data is same at time $t + t_1$ as it was at time t . If the storage duration of a data item is small then no replication is needed but if it is long enough then PRCR is applied. Here the storage nodes are periodically scanned and checked by a unit called the replication manager. After the scanning operation, it is decided which replica has to be sent for checking. This bears the issue of overhead.

Some other studies have discussed data replication in terms of bandwidth consumption and energy consumption. Energy consumption in cloud storage data centres is immensely big since a huge amount of data are stored and the systems run continuously to provide availability. In [12], database replication is done according to the popularity of data. The architecture adopted in this study is a level architecture having a central database at the top most level; data centres at the middle level and racks at the bottom most level. The system is modelled for performing data replication for minimizing the bandwidth utilization, energy utilization and network delay. Energy consumed by the servers while performing its operation an even when sitting idle is considered along with the power consumption of different types of switches. Bandwidth is utilized during data transfer operations, i.e. when the request for a data replica is invoked by the client to the server and when the server produces the response to the request in the form of a replica. Overhead is a consistent issue with the approach used by the authors. Moreover, in [12], the test bed implementation is not addressed in the proposed work of the authors.

In the work of [13], a graph-based approach is used for data replication. This work employs a locality replica manager (LRM) which coordinates the tasks of the system and takes care of the quality of service (QoS) requirements. Herein, data replication considers the physical neighbourhood of blocks for the query. With each inquiry request, a full graph is constructed for the blocks for which the inquiry is raised and these graphs are hosted by some data nodes. The decision as to who will host the inquiry graph is made based on parameters like the QoS requirement of the inquiry, delay of the graph, probability of a graph being available, storage space and load of the node on which graph is to be hosted. In this paper, a genetic algorithm based approach is used for maintaining the availability and delay of the system at desired level but this approach lacks efficiency. In [13], no real-time implementation of the proposed work has been performed.

3 Data Replication Algorithms

This section presents an insight into five data replication algorithm providing data replication and management functionalities. The main focus of these algorithms is to ensure data availability and reliability by providing a minimum number of quality of service satisfied replicas.

3.1 Adaptive Data Replication Strategy (ADRS)

Author, namely, Najme Mansouri in [13] has proposed a replica placement and replica replacement strategy for improving the availability of data with low cost. For the purpose of replica placement, five parameters are considered, namely, mean service time, failure probability, load variance, latency and storage usage. The aforementioned parameters are calculated as below:

$$\text{Mean Service Time (MST)} = \sum_{j=1}^m \left[\text{ST}(i, j) * \frac{\text{Acc}(i, j)}{\text{AR}(i)} \right] \quad (1)$$

In Eq. (1), $\text{ST}(i, j)$ is the expected time a file serves on a data node, $\text{Acc}(i, j)$ is the access rate of read requests coming from a data node asking for a file on it and $\text{AR}(i)$ is the mean access rate.

$$\text{Load variance (LV)} = \text{Acc}(i, j) * \text{ST}(i, j) \quad (2)$$

$$\text{Latency} = 1/r_i * \sum_{j=1}^m d(i, j) * \frac{\text{Size}(i)}{B(j)} * \text{Acc}(i, j) \quad (3)$$

Using the above equations, a cost function is calculated for each site. Smaller the value of this cost function, the better is its fitness value. The new replica is placed at a site with minimum cost function value.

$$\text{Cost function (CF)} = w_1 \times \text{MST} + w_2 \times \text{LV} + w_3 \times \text{SU} + w_4 \times \text{FP} + w_5 \times L \quad (4)$$

where SU is the storage unit and FP is the failure probability and w_1, w_2, w_3, w_4, w_5 are weight which are randomly assigned.

For replica replacement, the parameters considered are availability of file, last time the replica was requested, number of accesses of the replica and file size of replica. Based on these parameters, a V value is calculated and the replica whose V value is the least is replaced. The V value is calculated as below:

$$V = (w1 * NA + w2 * P)/(w3 * (CT - LA) + w4 * S) \quad (5)$$

where NA is the number of access, CT is the current time, LA is the last request time of replica and P is availability of file [14]. The working of this algorithm is further explained conceptually and in detail in Fig. 1.

In this work, the number of times the replication process need to be called is reduced because only when there is not enough space on the best site, then a replica is created. Response time to access the data is reduced because it is available in the best site and is easily available. Also only read-only data facility is provided which limits the capability of the user to write any changes to the data. No file partitioning is done thereby causing entire data to be stored at one storage location which can lead to high storage space cost.

3.2 Build Time Algorithm

This algorithm considers that data has already been placed at the right data centres by using some existing replication algorithm. Data in this algorithm is considered to be datasets distributed over data centres. The data sets are classified into two types: free flexible dataset (FFD) and constraint flexible dataset (CFD). The size of the storage space decides which category the data set belongs to. If the size of the data set is less than storage space it belongs to free flexible dataset category; otherwise, it belongs to constrained flexible dataset. The storage unit itself is divided into two, i.e. original data space and replicated data space. The dependency between two datasets and access frequency are the parameters considered for this algorithm. The dependency between two datasets is given as

$$\text{Dep}(D_i, D_j) = \text{Count}(T(d_i) \cap T(d_j)) \quad (6)$$

In the above equation, $T(d_i)$ represents the number of tasks which belong to that dataset. Access frequency is frequency with which a dataset is accessed, which is defined as follows:

$$\text{Acc}_{fdi} = \text{AccNum}_{di} / \text{time}_s \quad (7)$$

Here, AccNum_{di} represents the number of times a dataset is accessed and time_s is the time interval of data transfer [15]. Figure 2 gives the working flow of this algorithm in a conceptual manner.

The strategy is compared with a normal data replication scheme and observed a substantial decrease in total cost. Transfer cost is reduced because a dataset is placed at a place where the task it is used for is located. The algorithm does not discuss how the tasks are distributed among different data centres but it only discusses the datasets.

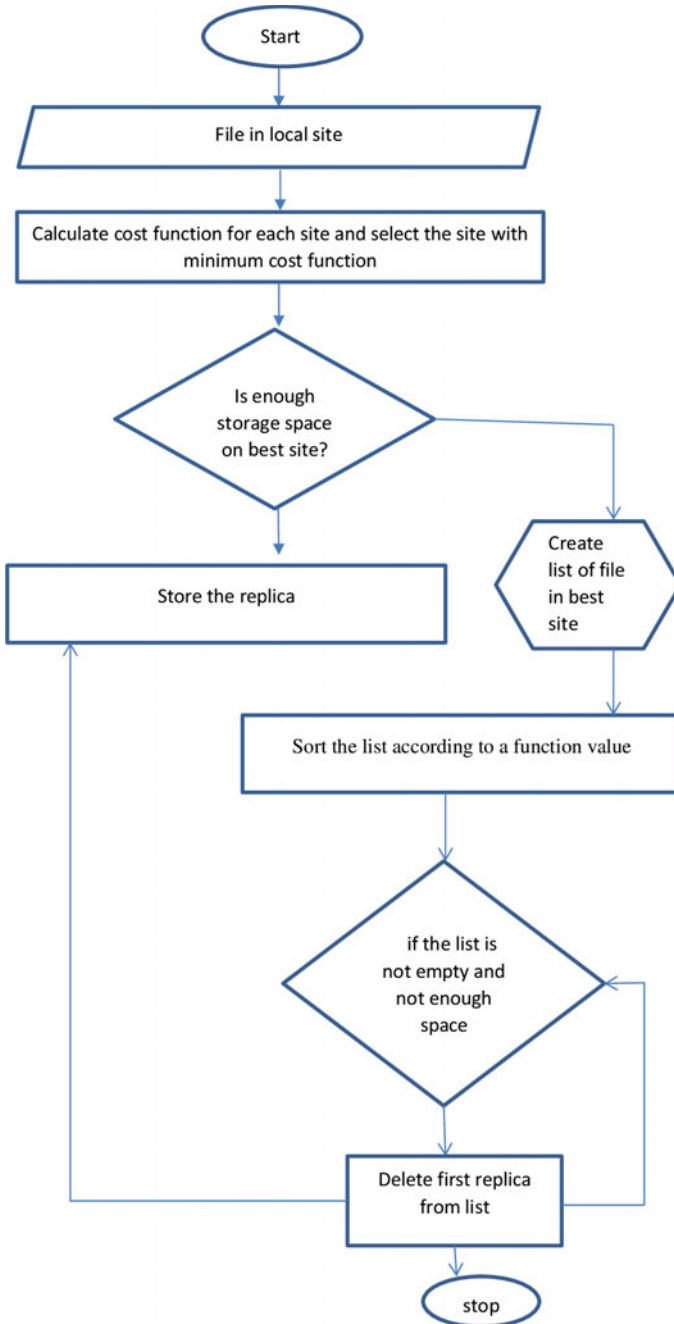


Fig. 1 Working of ADRS algorithm

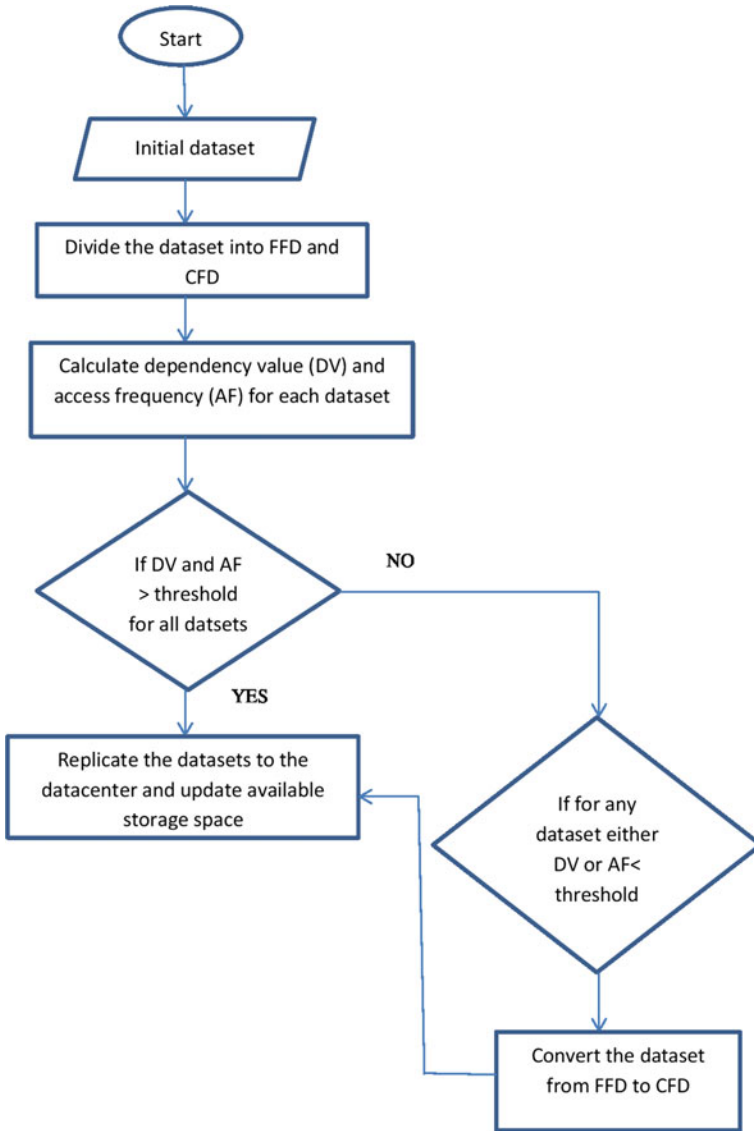


Fig. 2 Working of build time algorithm

3.3 High QoS-First Replication Algorithm

In the QoS aware data replication (QADR) problem [16] the HQFR algorithm is used for producing QoS aware data replicas. QoS can be defined differently according to the user's requirements; here it is taken to be the access time. This work also aims at minimizing the count of those replicas that violate QoS requirement. This algorithm is based on the HDFS [11] architecture and each data block has two copies of the original data. A replication request is generated when an application running on the disk attempt to write a data block. For the nodes to fulfil the replication request, they should be QoS qualified. To qualify as QoS nodes following two conditions should be met:

1. The requested node and the node on which data is to be placed should not be in the same rack so as to avoid any possible rack failure.
2. The time to access a node q_j , for a requested node r_i , should be less than the time required to access a QoS requested node.

The working of this algorithm is explained in Fig. 3 with the help of a conceptual approach.

Minimum cost and minimum number of replicas can be achieved in polynomial time.

Also, the system is highly scalable and can ensure that QoS requirements are fulfilled for a large number of nodes. Every time the higher quality of service node will be preferred so the load distribution is less.

3.4 Dynamic Replica Control Strategy

The algorithm proposed by Wei in [17] finds a cost-effective replica placement strategy which is based on capacity and blocking probability of nodes containing data. Capacity of the node is defined as the number of sessions it can allow during a time and period. Blocking probability is the probability of a request for a session being blocked by the data node as its capacity exceeded. The algorithm imposes a constraint on the bandwidth that can be used for session requests that it should be less the total bandwidth available for the network.

$$BW_{\text{net}} > \sum_{i=1}^c S(i)/T(i) \quad (8)$$

where BW_{net} is the network bandwidth, c is the upper bound on the number of sessions of a node S and T is the permissible delay. The blocking probability is calculated as

$$BP_i = (\lambda_i \tau_i)^c / c! * \left[1 / \sum_{k=0}^c (\lambda_i * \tau_i)^k / k! \right] \quad (9)$$

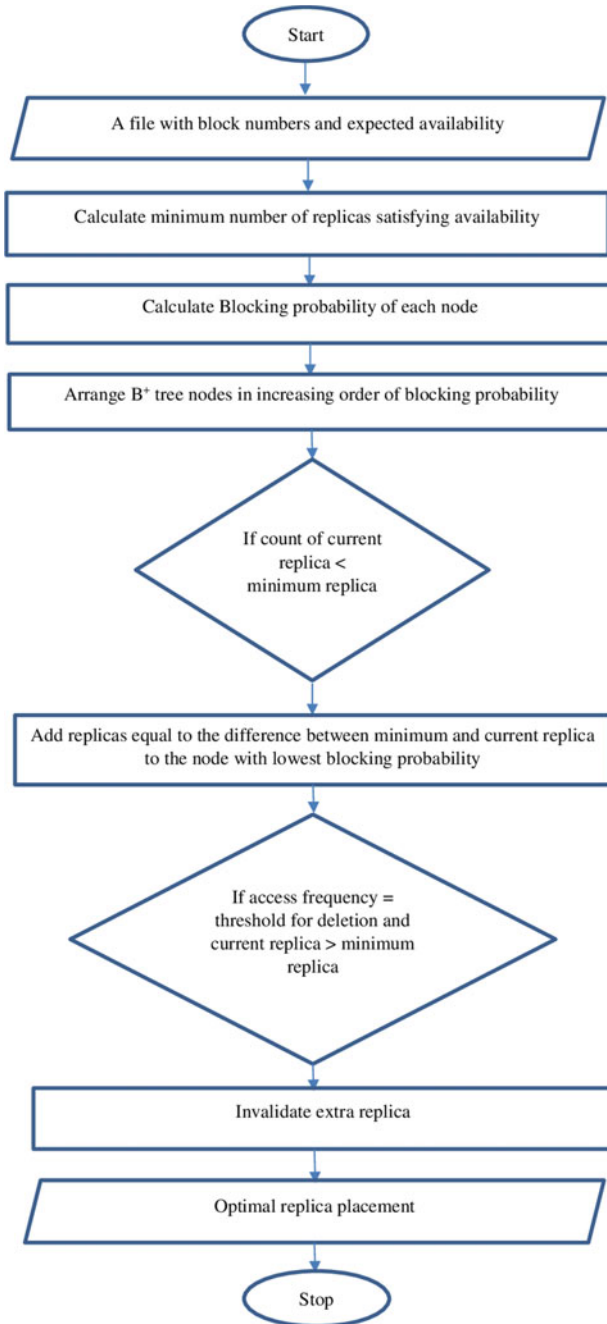


Fig. 3 Working of HQFR algorithm

where λ is the arrival rate of requests and τ is the service time. The replica will be stored at the node with the least value of blocking probability. The conceptual workflow of the algorithm is explained in Fig. 4.

The availability of the data is increased when the number of replicas is kept small. A few sessions are supported simultaneously which may lead to a lot of requests getting blocked and thus consume more bandwidth and time.

3.5 Cost-Effective Incremental Replication (CIR)

The CIR algorithm is proposed by Li in [18] works by creating replicas in incremental steps for the purpose of increasing data reliability. Data reliability is based on many factors such as the time data duration for which the data is stored, number of replicas to be stored and rate of failure of a node. Data reliability is calculated as below:

$$X = 1 - \prod_{i=1}^k (1 - e^{-\lambda_i * T_k}) \quad (10)$$

Here, k represents the number of replicas, λ_i is the failure rate and T_k is the storage duration. This algorithm finds the storage reliability for up to three replicas. Working flow of this algorithm is explained in a conceptual manner in Fig. 5. This algorithm is able to cater to very large data requirements and works well for data-intensive applications. As the replica is created only when the replica creation point is reached therefore number of replicas is reduced which further improves the storage capacity. While the reliability of data is discussed, the reliability of the metadata table which contains the information about actual replicas is not considered.

4 Analysis

A comparative analysis of the five replication algorithms discussed above is presented in Table 1. We observe that the major objective of all the algorithms discussed thus far is to maintain the availability and reliability requirements of the data while distributing the load evenly in a cost-efficient way. The ADRS algorithm optimizes various parameters like mean service time, failure probability, load variance and storage usage. It produces a better response time when the number of nodes is increased as compared to CIR, build time and DRCS algorithms.

The HQFR algorithm gives less computational complexity when calculating the number of replicas. The CIR algorithm performs better than build time algorithm in terms of network utilization. This algorithm also provides a very high reliability.

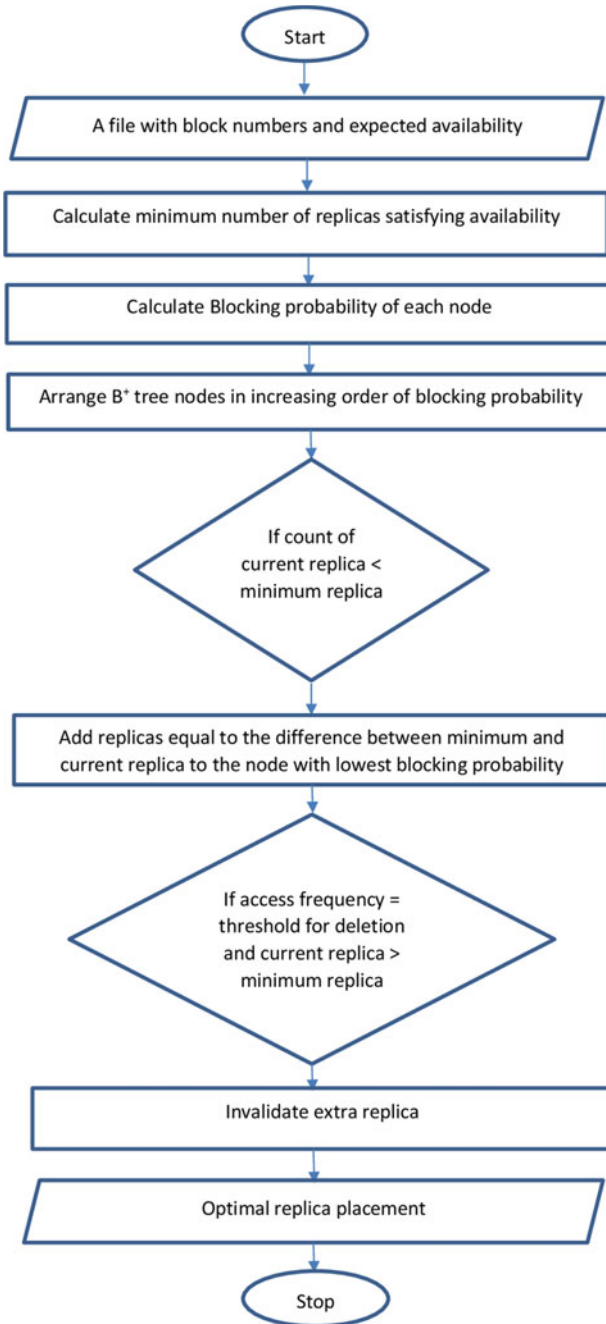


Fig. 4 Working of DRCS algorithm

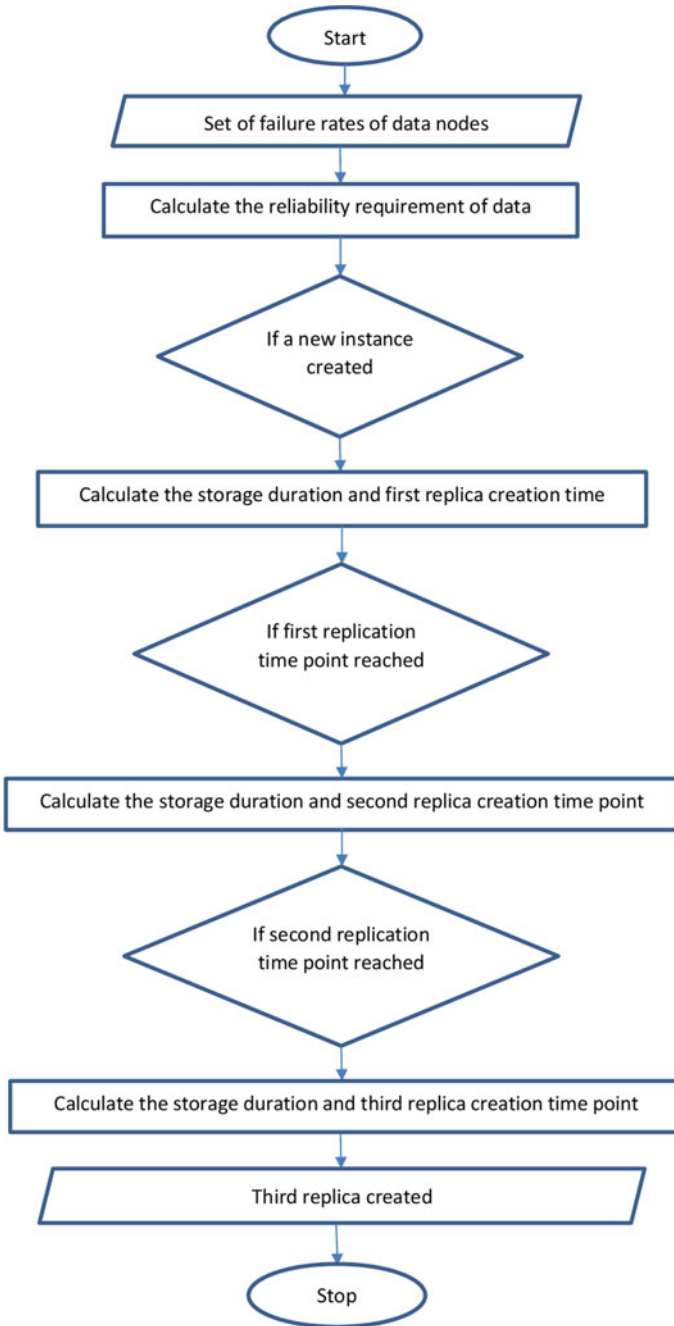


Fig. 5 Working of CIR algorithm

Table 1 Comparison of data replication algorithms

Algorithm	Author	Objective	Approach	Parameters considered	Advantage
ADRS	[14]	To minimize response time and make load balanced cloud storage	Multi-objective optimization	Mean service time, failure probability, load variance, storage usage	Load variance, response time, network usage is improved as the number of files increases
HQFR	[16]	To continuously support quality of service requirement of a data after corruption	Maintaining quality of service requirement	Access time of data block, disk access latency, network latency	Less computation complexity
DRCS	[17]	To maintain load balancing while placing replicas in a cost-efficient way	Placing the replicas with nodes having lowest blocking probability	Blocking probability, replication factor	Improves system's availability
CIR	[18]	To reduce storage cost and improve reliability	Incremental replication	Storage duration, failure rates	Provides 99.9999% reliability with three replicas
Build time	[15]	To reduce the cost of data storage and transfer for work applications	Divide data storage space and datasets into parts	Access frequency, dataset dependency, storage capacity, size of dataset	Greatly reduces the cost for data management

5 Conclusions and Future Scope

Data replication for cloud computing systems has become popular because of its availability to provide high data availability and ensure reliability. Replication is done at file level as well as block level. The algorithms discussed in this paper have particularly focused on maintaining the quality of service of the replicas stored in the data nodes. Quality of service is maintained by ensuring that the time and bandwidth required to retrieve a replica should not be more than the time to retrieve

the original copy. Apart from this, the creation of minimum number of replicas for an original data is taken into account. By taking all these parameters into consideration, a cost-effective reliable data replication can be performed. In future, the authors wish to overcome the problem of creating too many replicas by restricting the replica creation procedure. The work studied so far focuses mainly on read-only data thereby giving no importance to coordination among replicas. The future work will take into consideration consistency among replicas along with the best practices.

References

1. Amazon S3. (2017 March). [Online]. Available: <http://aws.amazon.com/s3/>.
2. Google Cloud storage. (2017, March). [Online]. Available: <https://cloud.google.com/storage/>.
3. Microsoft Azure. (2017, March). [Online]. Available: <https://azure.microsoft.com/en-in/overview/what-is-azure/>.
4. Kapoor, S., & Dabas, C. (2015, August). Cluster based load balancing in cloud computing. In *2015 Eighth International Conference on Contemporary Computing (IC3)* (pp. 76–81). IEEE.
5. Feng, K., & Zhang, J. (2017, April). Improving availability and confidentiality of shared data under the multi-cloud environment. In *2017 IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, (pp. 6–10). IEEE.
6. Mansouri, Y. (2017). Brokering algorithms for data replication and migration across cloud-based data stores (Doctoral dissertation).
7. Li, J., & Li, B. (2013). Erasure coding for cloud storage systems: a survey. *Tsinghua Science and Technology*, *18*(3), 259–272.
8. Luo, L., Li, H., Qiu, X., & Tang, Y. (2016, August). A resource optimization algorithm of cloud data center based on correlated model of reliability, performance and energy. In *2016 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, (pp. 416–417). IEEE.
9. Liu, Y., Li, X., Kang, R., & Xiao, L. (2016, October). Service reliability modeling of the IT infrastructure of active-active cloud data center. In *Prognostics and System Health Management Conference (PHM-Chengdu)*, 2016 (pp. 1–7). IEEE.
10. Li, W., Yang, Y., & Yuan, D. (2016). Ensuring cloud data reliability with minimum replication by proactive replica checking. *IEEE Transactions on Computers*, *65*(5), 1494–1506.
11. Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010, May). The hadoop distributed file system. In *2010 IEEE 26th symposium on Mass storage systems and technologies (MSST)*, (pp. 1–10). IEEE.
12. Boru, D., Kliazovich, D., Granelli, F., Bouvry, P., & Zomaya, A. Y. (2015). Energy-efficient data replication in cloud computing datacenters. *Cluster Computing*, *18*(1), 385–402.
13. Sookhtsarai, R., Artin, J., Ghorbani, A., Faraahi, A., & Adineh, H. (2016). A locality-based replication manager for data cloud. *Frontiers of Information Technology & Electronic Engineering*, *17*(12), 1275–1286.
14. Mansouri, N. (2016). Adaptive data replication strategy in cloud computing for performance improvement. *Frontiers of Computer Science*, *10*(5), 925–935.
15. Xie, F., Yan, J., & Shen, J. (2017, August). Towards cost reduction in cloud-based workflow management through data replication. In *2017 Fifth International Conference on Advanced Cloud and Big Data (CBD)*, (pp. 94–99). IEEE.
16. Lin, J. W., Chen, C. H., & Chang, J. M. (2013). QoS-aware data replication for data-intensive applications in cloud computing systems. *IEEE Transactions on Cloud Computing*, *1*(1), 101–115.
17. Wei, Q., Veeravalli, B., Gong, B., Zeng, L., & Feng, D. (2010, September). CDRM: A cost-effective dynamic replication management scheme for cloud storage cluster. In *2010 IEEE International Conference on Cluster Computing (CLUSTER)*, (pp. 188–196). IEEE.

18. Li, W., Yang, Y., & Yuan, D. (2011, December). A novel cost-effective dynamic data replication strategy for reliability in cloud data centres. In *Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on* (pp. 496–502). IEEE.

Integrated Cryptography for Internet of Things Using TBF Approach



Santosh Kumar Sharma, Muppidi Somasundara Rao,
Lavudi Poorna Chandar Rao and Pendyala Chaitanya

Abstract In current trends, Internet-based portable system is on the huge demand, and in coming future, it will be the most demanded technology among the smart society. It is very difficult to say that entire world will use the smart technology but majority of the developed and developing countries will base on smart work, using smart technology. The people can control their smart devices from remote location and can be causation free with security issues, which is associated with their valuables and other belongings. But all these are not so easy to implement practically to solve security pitfall and other issues to maintain reliable usage of smart devices. Here, in our proposed system, we are providing data security on data layer by using the concept of excessive cryptography and implementing the integrated approach of two cryptography techniques, i.e. TBF-transposition and blowfish cryptography techniques which can make powerful security barrier against the vulnerability and can stop the illegal usage of data. Here first of all, the plain text will go through transposition after that blowfish block cipher technique will be applied on consequential value from transposition outcome. This application will use in highly sensitive security purpose over the smart secure communication as we know that security issues are major challenges among smart devices where the data is more sensitive, and if it will go to wrong hands, there will be loss of human life in some cases. Finally, our work will contribute to integrate the crypto techniques for maintaining more security without breach of services.

Keywords Cryptography · IOT security · Transposition · Blowfish · Security services · Block cipher

S. K. Sharma (✉) · M. S. Rao · L. P. C. Rao · P. Chaitanya
Department of MCA, Vignan Institute of Information Technology,
Visakhapatnam 530049, India
e-mail: sharma.santosh83@gmail.com

© Springer Nature Singapore Pte Ltd. 2019
N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_4

1 Introduction

In present scenario, Internet-based technology and resources are reachable to almost every corner of the world. Due to the heterogeneous structure of IOT connectivity, it creates service reliability along with several security challenges such as authentication for every connected objects and privacy of data. Even though it is easy to access Internet-based devices, it invites lots of security challenges when it comes to daily life of human being; later on if the device control will go in wrong hand, then it may create serious damage to the connected objects. IOT is an integrated approach of hardware and software such as sensors, actuators, transistors and processors. IOT needs a high degree of computational aptitude to perform the smart task. IoT is an integrated product of electronic and electromagnetic devices such as sensors, actuators and microcontrollers with internet connectivity which are embedded on high degree circuit to conduct specific task. IoT is having major applications to make the system smart that can be explore in face of smart city, smart health management system, smart farming, smart home appliances, smart displays and smart security. For making the system elegant, there is a need to collaborative the smart technology but it should be mange through permitted users only to avoid the malicious usage and access of resources.

2 Related Work

By studying several scientific journals and articles related to database security and wireless sensor network security is exposing the several security threats to authentic data accessing and transmission. Here Huang [1, 2] has discussed the application of signature-based authenticated technique and the successful test on IOT devices with the help of Burrows–Abadi–Needham logic for security and using of automated tools and applications (AVISPA) tool. Instead of analysing a single-level security, multiple nodes can also be secured, which provides security in communication, application interface and cryptography for IOT security. Situation is not so far and by the year of 2020, IOT devices will go beyond the billions counts, which create the collective issues related to collaboration techniques. Security is a major challenge in presence of many anomalies & malicious user. Kang and Lee [3, 4] has discussed the security at different stages such as communication security, application level security. Where as Kim. S has discussed deep impact of block chain technique on IoT devices. Zahra, Majeed and Mohsin [5–7] has done the risk analysis and projected the framework for security of IoT as future proof and Quality of services is provided by the Maliki, Khanna with trust management approach [8–10]. Nawir, Baldini and Zada Khan [11–13] they have presented several taxonomy for preventing the attacks with security certificate. Metongnon, Nakagawa and Ben [14–16] has kept the heterogeneous behaviour of IoT and how to secure this with the help of agent platform mechanism for social IoT. Zonari, Midi and Urien [17–19] has discussed the different aspects

of homomorphic cryptographic technique and intrusion detection with threat assessment.

With the continuous growing of IOT technology, 'Things' alarming several critical issues subsequently regarding how to access and monitor the authentication management to transmit and access the data in wireless environment security management is difficult in wireless environment which leads the service segment issues towards the devices and human life. It turn out to be so risky once we fail to provide service communication among them, Sklavos [20] has discussed several security issues to solve this. Sharma and Baik [21, 22] have discussed Massey–Omura cryptography approach as a private key encryption technique with its application and strength. M-Omura works on the precept of the prime modulus with exponential system, where the message is generated with secret key and transmits to the receiver side by using prime modulo. Our study found different types of attacks and threats to our valuables along with its possible security measures such as bootstrap modelling, cryptography (symmetric/asymmetric) authentication protocol, security proxy servers and scanners; in spite of all these available security tools, we cannot confident about data security and we need more complex security techniques to safe our things and data. Gong, Hu, Wen and Bose [23–25] then presented security mechanism at physical layer and how the data can be transmit intelligently over the network by focusing the different security techniques in different layers of IoT architecture. Andreas and Atat [26, 27] proposed how to analyse the risk assessment and security certification with Armour project in IoT. Xu and Moon [28, 29] introduced the concept of two-factor authentication protocol whose working principle is to work in multi-server system to provide more security for known attacks. Further, Jongho discussed the concept of online enemy attack with process saver authentication protocol using AVISPA and random Oracle. Jesus, Caminha and Du [30–32] revealed the strategy of the block chain for securing the Internet of things, and Jean discussed the trust management to show the issue of on-off attack for the IoT devices. Zhou, Wang and Li [33–35] worked with the quantum cryptography for analysing the upcoming future security challenges. Chen Wang provided the concept of novel security scheme for IoT by implementing on instant encrypted transmission.

3 Proposed System

3.1 Transposition/Permutation

Transposition work is to produce the transpose of given matrix unless by supporting column values with row values. First, we take a message and change the order of message as a transposition and split this transposition message into four equal halves and named it as 'P(x)'.

EXAMPLE Take a message which is string

Message = "IOT FOR MILITARY SERVICES"

Perform or split this message into transposition and the result will be as

I	O	L	R	R	E
O	R	I	Y	V	Z
T	M	T	S	I	Z
F	I	A	E	C	Z

where 'Z' are taken as dummy values, and the splitting result is

IOTF	ORMI	LITA	RYSE	RVIC	EZZZ
------	------	------	------	------	------

3.2 Blowfish Algorithm

The working principle of blowfish is symmetrically block cipher approach, and it is originated to replace the IDEA algorithm in cryptography. This blowfish algorithm is used to convert plain text into cipher text which is entrenched with 32–448 bit encryption and work with the block size of 64 bit, operational activity for making final cipher is 16 rounds. Operational behaviour of blowfish as follows:

3.3 Algorithm

- Step-1: Initiate the cryptography process.
- Step-2: Let message $(M)^T = P(x)$ //Apply transposition.
- Step-3: Arrange the message in permutation order (transposition). If the permutation order is not filled, place a dummy value.
- Step-4: New confuse matrix $[m]^C M =$ cipher value.
- Step-5: Split the message as blocks in horizontal order so that each block contains 4-values that is taken as $P(X)$.
- Step-6: $P(X) \text{ } \forall 64$ bits.
- Step-7: Divide 64 bit into 2 equal halves (32 bit) and named as left (X_L) and right (X_R).
- Step-8: For $P_n = 1$ to 16 //if $n =$ true then next step otherwise go to Step-11.
- Step-9: Perform operation $X_L = X_L \oplus P_i$.

Table 1 Comparison of available algorithms with blowfish

Cryptography Algorithms	Key size (BITS)	Block size (BITS)	Security	Speed
DES	56	64	Less secure	Low
RC6	128,192	128	Less secure	Low
AES	128,192,256	128	More secure	Fast
RSA	Maximum 1024 Bits	Minimum 512 bits	Less secure	Fast
Idea	128	64	More secure	Fast
Blowfish	32-448	64	More secure	Fast

Step-10: Perform operation $X_R = F(X_L) \text{ XOR } X_R$

$$\{ \text{ Where } F(X_L) = ((s_1[a] + s_2[b] \text{ mod } 2^{32}) \oplus s_3[c]) + s_4[d] \text{ mod } 2^{32}$$

Logically it can be expressed as $F(X_L)$

$$= ((s_1[a] + s_2[b]\%2^{32}) \oplus s_3[c]) + s_4[d]\%2^{32}$$

Step-11: Run Swap() $X_L \rightleftharpoons X_R$.

Step-12: Perform X_L and X_R operations up to i value 16 and swap operation (undo the last swap).

Step-13: Perform operation $X_R = X_R \oplus P17$.

Step-14: Perform operation $X_L = X_L \oplus P18$.

Step-15: Recombine X_L and X_R .

Step-16: Stop.

3.4 Application and How It Works

This article says about IOT security that helps to make secure data services. They are equipped with latest crypto techniques to fight against the vulnerability of services. Instead, constant improvement leveraging advanced technology is needed for the tools to enable security forces edge out the amateurs. Security is a major challenge during the development of IOT application where huge number of unprivileged user try to use the resources illegally. So our work is to create the complexity for malicious users in case of intercepting the confidential data services. TBF creating the two-level encryption technology where first level work is to create confusion for the amateur using transposition, and the second level provides the encryption algorithm to secure the data, after a long course of action making the concrete secure system for the IOT (Fig. 1; Table 1).

DES: Data encryption is a old encryption algorithm which work on the symmetric key where only one key is use for encryption and decryption process which is shared

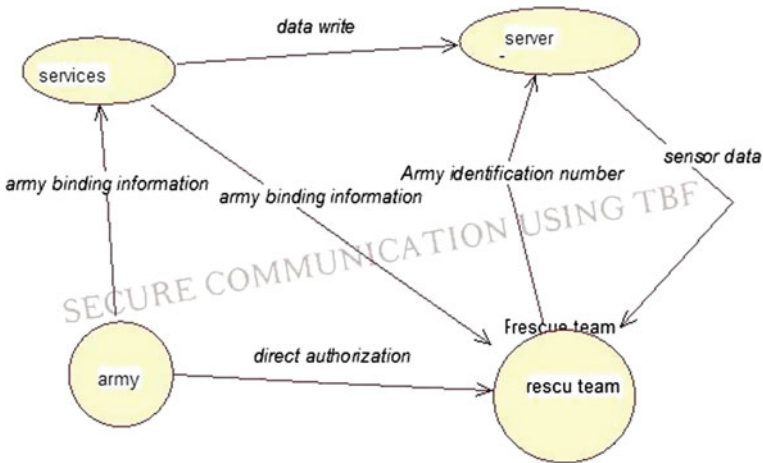


Fig. 1 Secure communication using TBF

by both sender and receiver as a private key. Here we are processing the block size of 64 bit on which DES is applying 56 bit key length.

RC6: RC is stand for Rivest cipher which is available in market from RC 1 to RC6 versions (since 1987 to 1997) which is symmetric key block size 128 bits cipher and support the size from 128, 192, 256 and 2040 bits up to maximum length.

AES: Advanced encryption standard algorithm is the powerful and well known block cipher techniques which implement on block cipher with symmetric key concept. AES is considerably faster than DES with small key size which provides fast and secure communication with the help of strong cryptography process with 16-rounds of encryption process.

RSA: RSA is named on three scientist Rivest, Shamir and Adleman who has given the widely accepted first public key cryptography for secure data transaction . Among all the available public key cryptography it is most popular algorithm which work prime number products and other operation. RSA algorithm is used as a basic algorithm to generate other advance algorithm such as Massy Omura and Diffie hellman algorithm.

IDEA: Apart from all the cryptography algorithm the IDEA (International Data Encryption Algorithm) is having its own unique importance which implement on 64 bit block size with 128 bit key along with 8 number of rounds to transform the plaintext into subsequence strong cipher text.

Blowfish: Blowfish algorithm is also a symmetric block cipher technique which is invented in 1993 to produce big number of cipher sets. Its block size is 64 bit and takes key length from 32 bit to 448 bit and complete the encryption process in number of rounds with fixed size of boxes such as Feistel algorithm.

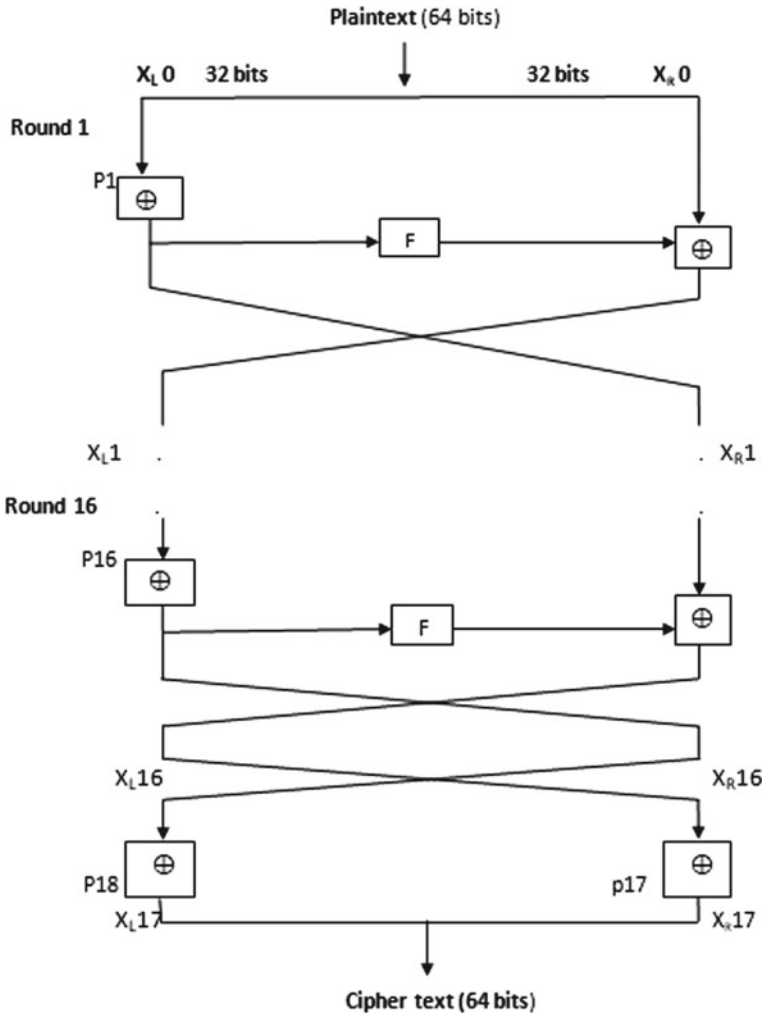


Fig. 2 Encryption process

$$X_L \leftarrow X_R$$

Round 2:

(a) Now perform \oplus (XOR) operation for X_L and p_2

$$\begin{array}{r}
 X_L = 00000000 \ 00000000 \ 00000001 \ 00010010 \\
 \oplus P_2 = 10000101 \ 10100011 \ 00001000 \ 11010011 \\
 \hline
 10000101 \ 10100011 \ 00001001 \ 11000001
 \end{array}$$

$$\begin{array}{cccc}
 F(X_L) = & 00000000 & 00000000 & 00000001 & 00010010 \\
 & \downarrow & \downarrow & \downarrow & \downarrow \\
 & A & B & C & D
 \end{array}$$

Add(Σ) values of A and B

$$\begin{array}{cccc}
 & 00000000 & 00000000 & 00000000 & 00000000 \\
 \Sigma & 00000000 & 00000000 & 00000000 & 00000000 \\
 \hline
 & 00000000 & 00000000 & 00000000 & 00000000
 \end{array}$$

Perform \oplus (XOR) operation of A + B value with C

$$\begin{array}{cccc}
 & 00000000 & 00000000 & 00000000 & 00000000 \\
 \oplus & 00000000 & 00000000 & 00000000 & 00000001 \\
 \hline
 & 00000000 & 00000000 & 00000000 & 00000001
 \end{array}$$

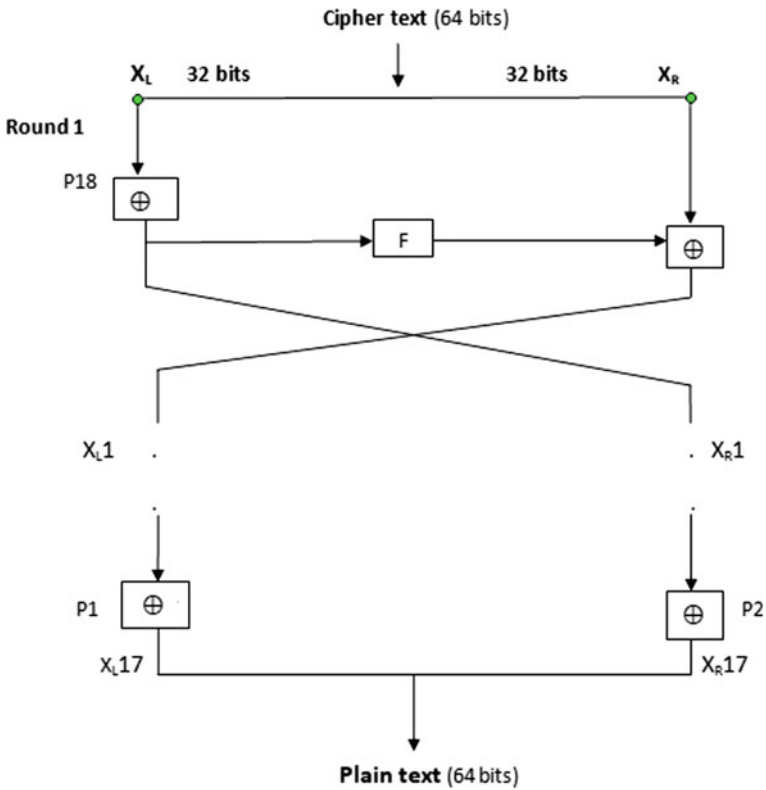


Fig. 3 Decryption process

Add (Σ) C value with D

$$\begin{array}{r} 00000000 \ 00000000 \ 00000000 \ 00000001 \\ \Sigma \ 00000000 \ 00000000 \ 00000000 \ 00000010 \\ \hline 00000000 \ 00000000 \ 00000000 \ 00000011 \end{array}$$

(b) Now perform \oplus (XOR) operation for $F(X_L)$ and X_R

$$\begin{array}{r} F(X_L) = 00000000 \ 00000000 \ 00000000 \ 00010011 \\ \oplus \ X_R = 10000101 \ 10100011 \ 00001000 \ 11010011 \\ \hline 10000101 \ 10100011 \ 00001000 \ 11000000 \end{array}$$

Swap X_L and X_R values so the values of X_L and X_R are interchanged.
Perform above operations up to the P18.

(b) Decryption:

The decryption means converting cipher text to plain text. In the above encryption process, we performed operations as a result we got cipher text. Now convert this cipher text into plain text.

From the above context Fig. 1 is giving the demonstration for secure communication using TBF, Fig. 2 showing the Encryption process and Fig. 3 is presenting the decryption process.

4 Conclusion

By traversing the numerous factors related to security breach, we proposed and concluded that single security cannot be work, and demanded to create the multiple approaches to provide the security at every level to provide the maximum security of data resources. In future work, our proposal is to design the secure communication channel for the military vehicles to protect themselves from any casualties using end-end security mechanisms.

References

1. Huang, H. (2017). Private and secured medical data transmission and analysis for wireless sensing healthcare system. *Citation information: Transactions on Industrial Informatics* 1–10.
2. Challa, S., Wazid, M., Das, A. K., & Kumar, N. (2017). (Member, IEEE), Secure signature-based authenticated key establishment scheme for future iot applications. *IEEE Access*, 1–16.
3. Sain, M., Kang, Y. J., Lee, & H. J. (2017). Survey on security in internet of things: State of the art and challenges, pp. 699–704.

4. Huh, S., Cho, S., & Kim, S. (2017). Managing IoT devices using block chain platform, pp. 464–467.
5. Zahra, B. F., Fatima Zahra, B. (2017). Risk analysis in internet of things using EBIOS.
6. Majeed, A. (2017). Internet of Things (IoT): A verification framework.
7. Mohsin, M., Anwar, Z., Husari, G., Al-Shaer, E., & AshiqurRahman, M. (2016). IoTSAT: A formal framework for security analysis of the internet of things (IoT).
8. Abels, T., Khanna, R., & Midkiff, K. (2017). Future proof IoT: Composable semantics, security, QoS and reliability, pp. 1–4.
9. El-Maliki, T. (2016). Efficient security adaptation framework for internet of things, pp. 206–211.
10. Ben Abderrahim, O., HoucineElhdhili M., & Saidane, L. (2017) TMCoi-SIOT: A trust management system based on communities of internet for the social internet of things, pp. 747–752.
11. Nawir, M., Amir, A., Yaakob, N., & Bi Lynn, O. (2016). Internet of things (IoT): Taxonomy of security attacks, pp. 321–326.
12. Baldini, G., Member, IEEE, SKarmeta, A., Fournere, E., Neisse, E., Legeard, B., & Gall, F. L. (2017). Security certification and labelling in internet of things, pp. 627–632.
13. Zada Khan, W., Mohammed Zangoti, H., Aalsalem, M. Y., Zahid, M., Arshad, Q. (2016). Mobile RFID in internet of things: security attacks, privacy risks, and countermeasures, pp. 36–41.
14. Metongnon, L., Eziny, E. C., & Sadre, R. (2017). Efficient probing of heterogeneous IoT networks, pp. 1052–1058.
15. Nakagawa, I., & Shimojo, S. (2017). IoT agent platform mechanism with transparent cloud computing framework for improving IoT security, pp. 684–689.
16. Ben Abderrahim, O., Elhedhili, M. H., & Saidane, L. (2017). CTMS-SIOT: A context-based trust management system for the social internet of things, pp. 1903–1908.
17. Zouari, J., Hamdi, M., & Kim, T.-H. (2017). A privacy-preserving homomorphic encryption scheme for the internet of things, pp. 1939–1944.
18. Midi, D., Rullo, A., & Mudgerikar, A. (2017). Kalis a system for knowledge-driven adaptable intrusion detection for the internet of things, pp. 656–666.
19. Dorsemaine, B., Gaulier, J. P., & Urien, P. (2017). A new threat assessment method for integration an IoT infrastructure in an information system, pp. 105–112.
20. Sklavos, N., & Zaharakis, I. D. (2016). Cryptography and, security in internet of things (IoTs): Models, schemes, and implementations.
21. Sharma, S. K. (2017). A survey on layered approach for internet of things security. *SERSC, ASTL, SMART DSC-2017* (Vol. 147, pp. 26–33).
22. Sharma, S. K., Baik, N., Khuntia, B. (2018) Encrusted security for internet of things using MAC-OMURA. *IJCA, SERSC-Australia*, pp. 45–54.
23. Gong, B., Wang, Y., Liu, X., Qi, F., & Sun, Z. (2018, February). A trusted attestation mechanism for the sensing nodes of internet of things based on dynamic trusted measurement, pp. 100–121.
24. Hu, L., & Wen, H. (2018, February). Cooperative jamming for physical layer security enhancement in internet of things (pp. 219–228).
25. Bose, R. (2017). Channel-based mapping diversity for enhancing the physical layer security in the internet of things (2017 IEEE).
26. Andreas. (2018, January). Wireless communication and security issues for cyber-physical systems and the internet-of-things, pp. 38–60.
27. Atat, R., Liu, L., Ashdown, J., Medley, M., Matyjas, J., Yi, Y. (2017). A physical layer security scheme for mobile health cyber-physical systems, pp. 1–15 (2017 IEEE).
28. Xu, G., & Li, W. (2018, 15 April). A secure and anonymous two-factor authentication protocol in multi server environment, pp. 1–15.
29. Moon, J., & Youngsook, L. (2017, 27 September). Improving an anonymous and provably secure authentication protocol for a mobile user, pp. 1–13.
30. Jesus, E. F., Chicarino, V. R., & de Albuquerque, C. V. (2018, 8 April). A survey of how to use blockchain to secure internet of things and the stalker attack, pp. 1–27.

31. Caminha, J., & Perkusich, M. (2018, 15 April) A smart trust management method to detect on-off attacks in the internet of things, pp. 1–10.
32. Du, Q., & Song, H. (2018, 8 February). Security enhancement for multicast over internet of things by dynamically constructed fountain codes, pp. 1–11.
33. Zhou, T., Shen, J., Li, X., & Wang, C. (2018, 21 February). Quantum cryptography for the future internet and the security analysis, pp. 1–7.
34. Wang, C., & Ren, T. L. (2018, 17 May) A novel security scheme based on instant encrypted transmission for internet of things, pp. 1–7.
35. Li, Q., Zhu, H., & Zhang, T. (2018, 6 June) Traceable cipher text-policy attribute-based encryption with verifiable outsourced decryption in e-health cloud, pp. 1–12.

Data Governance on Local Storage in Offsite



G. Priyadharshini and K. Shyamala

Abstract The major challenges faced by software industry are how to restrict confidential data and protect copyright or intellectual property information between customer location and delivery center. Typically customer will have multiple vendors spread across various geographical locations. In global delivery model, customer sensitive information will be exchanged between teams and there is possibility of data breach from customer network. Though these delivery centers are firewall segregated or air gap network, it is difficult to restrict end user to store information in local device. Thin client installation at delivery center or Virtual Desktop Infrastructure (VDI) setup at customer location are trivial solution to ensure information is not moved out of network. All traditional offshore development centers may not have thin client set up and they use workstation with local storage. Converting these workstations into thin client is expensive and customer may not be ready to provide VDI setup for offsite location. This paper provides simple solution with zero investment for local workstations to act like Thin Client, and also to ensure that there is no data or information leakage through local storage.

Keywords Data security · Local storage restriction · Group policy object · Thin client · VDI · Data protection

1 Introduction

For any organization, one of the key strategic assets is data. Considering the way data are growing and shared through multiple channels, it is equally important to do data classification. We also need to take utmost care for protecting sensitive, confidential, and restricted data. In information security, we need to secure both tangible and intangible assets to ensure it is trustworthy to use for business. Tangible assets include workstation, access points, servers, router, firewall, switches, etc. Intangible

G. Priyadharshini (✉) · K. Shyamala
Research Department of Computer Science, Dr. Ambedkar Government
Arts College, Chennai 600039, India
e-mail: sushpriya@yahoo.com

© Springer Nature Singapore Pte Ltd. 2019
N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_5

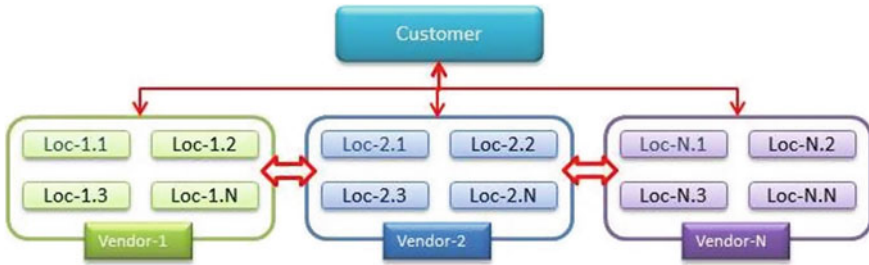


Fig. 1 Sensitive information flow across all locations

asset includes software code, development life cycle documents, customer details, contracts, invoices, statement of works, personally identifiable information, and so on. To secure assets, we also need to consider cost incurred for securing information. Securing intangible asset plays vital role and challenge to keep them with restricted access.

In this digital world, personal and industrial sensitive information spread across the world and are stored in the cloud. Considering vulnerabilities and public threat, there is no guarantee that these information are secured. There are several regulatory bodies and acts established across the globe to guide and product data. These regulatory bodies want to give control over how individual data are used by organization. The organization will comply with these regulatory bodies based on the geography or industry it operates. For example, healthcare industries follow HIPAA (Health Insurance Portability and Accountability Act) which is established in 1996. The organization which stores credit card information is following payment card industry data security standard (PCI DSS).

United Kingdom (UK) came up with The Data Protection Act in 1998 [1] to protect personal data stored on computers or in an organized paper filing system. Europe Union (EU) establishes GDPR (Global Data Protection Regulation) which will apply for all EU states from May 2018 (Fig. 1).

A customer may have multiple vendors for their operations like information technology, marketing, administration, human resources, etc. Each vendor will operate from different locations spread across customer site, nearshore, and offshore. Customer data will flow across all these locations in the form of document, data, and source code which include sensitive detail.

As per “Infowatch Analytic center research report on confidential data leakage” [2], 66% of leaks are from employees who include intentional as well as accidental. The only way to prevent such data leakage is not to store any data locally and monitor local storage on periodic intervals (Fig. 2).

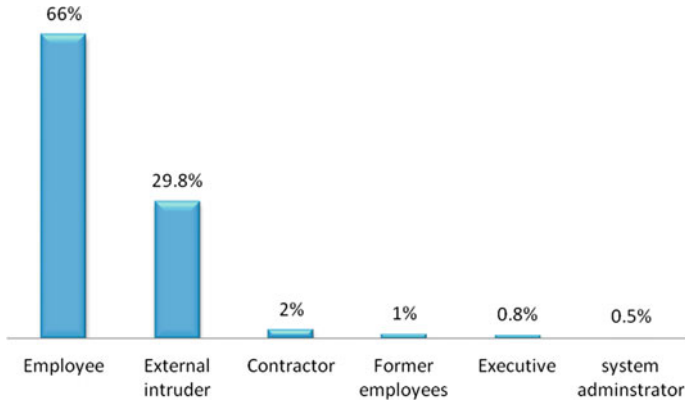


Fig. 2 Sources for data leaks

As per data protection act, data [3]:

- (a) is being processed by means of equipment operating automatically in response to instructions given for that purpose,
- (b) is recorded with the intention that it should be processed by means of such equipment,
- (c) is recorded as part of a relevant filing system or with the intention that it should form part of a relevant filing system,
- (d) does not fall within paragraph (a), (b) or (c) but forms part of an accessible record as defined by Section 68, or
- (e) is recorded information held by a public authority and does not fall within any of paragraph (a) to (d).

Personal data [3] means data which relate to a living individual who can be identified:

- (a) from those data, or
- (b) from those data and other information which is in the possession of, or is likely to come into the possession of, the data controller,
- (c) any expression of opinion about the individual and any indication of the intentions of the data controller or any other person in respect of the individual.

2 Related Work

There are several tools available in the market to prevent data leakage from local network. Most of leading products are licensed version and expensive to implement. These software demonstrate the need for a high-profile acceptable use policy to prevent data leakage and give practical guidance on how to allot funds for IT security.

The challenges of protecting sensitive data from internal users, preventing from insiders and monitoring data leakage are the most difficult area in information security. Security personnel finding it difficult to identify potential insider threats by investigating suspicious activities in a person's physical behavior. Malicious insiders continue to steal data from network leaking personally identifiable information (PII). Whatever the approach, researchers are struggling to define exact problem and find the solution for the problem for all scenarios. The research of Huth et al. [4] gives a different solution to address components of the problem, with the goal of minimizing the potential malicious insiders can inflict on an organization.

It is not easy to identify insider who is trying for security breach data and reason for data leakage through the privilege and access given for their regular task. There are few solutions derived through ontological approach, and utilize the notion of semantic associations and their discovery among a collection of heterogeneous documents. New approaches have been documented seeking to address components of the problem, with the goal of reducing the harm malicious insiders can inflict on an organization. An Ontological Approach to the Document Access Problem of Insider Threat [5] describes the research and prototyping of a system that takes an ontological approach and is primarily targeted for use by the intelligence community.

David Clark et al. did research on Quantitative Analysis of the Leakage of Confidential Data [6] in which the team uses information theory to analyze the sensitive and confidential information which may be leaked by software applications written in very simple conditional statements and improper iterative statements. This analysis helps to determine the bounds on the quantity of information leaked.

Data leakage detection model has been developed [7] for data protection. This research did a detailed study on watermarking technology and proved that this technique will not help for data protection. This study also gives the difference between watermarking technology and data leakage detection model.

None of these studies provide the solution on preventing data leakage from local storage through a potential insider. This research focuses on a simple solution for such a scenario without any additional investment. This solution is fully automated and there is no need for any manual intervention.

3 Problem Statement

When the vendor is working on customer's data which is sensitive or contains personal detail like personal Identifiable Information (PII), vendor must comply with data protection act based on Industry or geography. When an organization subcontract or outsource their work, they need to ensure that vendor also support data protection act and data is not misused or leaked through subcontractor or vendor organization. As per the report published by breachlive [8], 59 records are stolen every second.

As per global leakage report 2017, published by infowatch [2], data leakage through instant messaging, network is increased in the last 1 year. The share of data leaks resulting from theft/loss of equipment, as well as through removable media

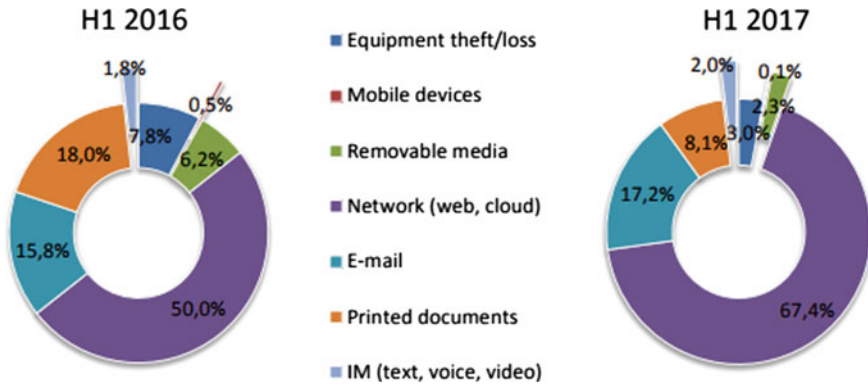


Fig. 3 Distribution of Leaks by channel

and paper documents must be reduced [2]. It is necessary to find a solution to avoid data leakage through network (Fig. 3).

Breachliveindex [9] research report presents top ten data breaches in the last 5 years which are listed in Table 1. Sources for these breaches are malicious outsider and malicious insider.

The printing area is 122 mm × 193 mm. The text should be justified to occupy the full line width, so that the right margin is not ragged, with words hyphenated as appropriate. Please fill pages so that the length of the text is no less than 180 mm, if possible.

To secure customer data, few organizations prefer to establish the connection with the vendor through VDI (Virtual Desktop Infrastructure), citrix receiver or forcing vendor to have air gap offshore development center. In air gap ODC or shared network, several security controls like providing shared folder, spot check are established to avoid local storage.

VDI [10] effectively addresses and resolves key problems in terminal server-based approaches to server-based computing. VMware ESX Server allows multiple user desktops to run as separate virtual machines while sharing underlying physical hardware resources such as CPU, memory, networking, and storage. This isolates users from each other, giving each user their own operating system, allowing granular resource allocation and protecting users from application crashes and operation system faults caused by the activities of other users.

VDI interacts with remote virtual machine and local desktop acts as dump terminal. The operating system, application, and data are stored on virtual machine hosted on central server. This saves desktop management cost, software licensing cost and most importantly it improves security and data leakage from customer location. It requires additional investment on customer side. When customer does not want to invest in setting up VDI and keen only on data leakage, we need solution at local desktop used by vendor at various geographical locations.

Table 1 Top 10 data breaches

Organization breached	Records breached	Date of breach	Source of breach	Industry
Equifax	143,000,000	7/15/2017	Malicious Outsider	Financial
Friend Finder Networks	412,214,295	10/16/2016	Malicious Outsider	Entertainment
Anthem Insurance Companies (anthem blue cross)	78,800,000	1/27/2015	State Sponsored	Health care
Home Depot	109,000,000	9/2/2014	Malicious Outsider	Retail
JPMorgan Chase	83,000,000	8/27/2014	Malicious Outsider	Financial
eBay	145,000,000	5/21/2014	Malicious Outsider	Retail
Korea Credit Bureau, NH Nonghyup Card, Lotte card, KB Kookmin card	104,000,000	1/20/2014	Malicious Insider	Financial
Target	110,000,000	11/4/2013	Malicious Outsider	Retail
Adobe Systems, Inc.	152,000,000	9/18/2013	Malicious Outsider	Technology
MySpace	360,000,000	6/11/2013	Malicious Outsider	Other

Users tend to download various data from customer site and refer them for particular release. The data can be in the form of document, manuals, procedures, processes or sensitive information. Most of the users will not delete these files and there is a chance that some sensitive information stored in these folders without user knowledge. In addition to default files, the users will download customer information for project reference. After downloading the file, user will come to know that few files may not be needed for project purpose, but felt good to have in the system. These will lead to data leakage as per customer contract and lead to regulatory compliance issue. To avoid such scenario, it is important to wipe out unwanted files from location storage on regular basis. Table 2 gives some scenarios which lead to keep sensitive data locally without the user's knowledge.

Table 2 Common files stored locally by the user

Scenario	Description
Temporary files	Temporary files created by application
Downloaded	Any files downloaded from Internet will be stored in default location
Instant messages	All individual or group chat transcription and file shared between users will be stored here

3.1 Research Objectives

The objective of this research is to find a low-cost solution to prevent data storage from local drive where data flows between various vendor group and customer. In a scenario where customer does not have VDI, citrix environment, vendor needs to implement multiple controls including monitoring and log review to restrict sensitive data within customer network.

This solution provides following benefits to vendor and customer:

- Restrict local storage with zero cost.
- Does not require any additional infrastructure set up.
- There is no manual process.
- Zero maintenance.
- No need for any additional review or log monitor.
- Easily portable and replicate to multiple locations.
- User group and file type are configurable.
- Policies and groups can be modified in no time.

4 Proposed Solution

Hackers and malware attackers are a major threat to data loss. These attackers target right from global customers (particularly banks, health care, retail, and insurance) to novice users. However, accidental breaches caused by employee error or data breached while controlled by third-party suppliers continue to be a major problem, accounting for 30% of breaches overall.

To avoid data theft/leakage from the network, no information should be stored locally for a longer duration. The data includes user created, system generated, downloaded, temporary files, and cookies. This can be achieved by implementing different Group Policy Objects to various user groups.

Group Policy Object (GPO) is a collection of settings that define what a system will look like and how it will behave for a defined group of users. Microsoft provides a program snap-in that allows you to use the Group Policy Microsoft Management Console (MMC). The selections result in a Group Policy Object. The GPO is associated

with selected Active Directory containers, such as sites, domains, or organizational units (OUs). The MMC allows you to create a GPO that defines registry-based policies, security options, software installation and maintenance options, scripts options, and folder redirection options [11].

To form a user group, we need to do detailed analysis based on nature of work by the end user. The users may belong to any of below categories but not limited to these groups alone:

- Users who are working on heavyweight software which has dependency on local storage for installation and keeping libraries and APIs.
- Users who require more storage for coding, compilation, and testing.
- Users do not require any storage and work on maintenance and production support.

4.1 Group Classification

- Group 1: Those who do not construct any code and does not require any local storage.
- Group 2: Those who develop code, store libraries, and make changes in configuration and settings. These groups can store files for their development purpose, but should not store any documents or references on permanent basis.
- Group 3: Third group includes those who are considered as exceptions and require storage for a longer duration. This group may work on heavy weight applications or working on document management related activities.

4.2 Solution for Each Group

- Group 1: Local storage for this group should have two logical drives. Primary drive should be hidden for this group and used for operating system and software installation. Only secondary drive should be visible to this group, and they are allowed to store their files in secondary drive or desktop. The files stored in desktop are redirected internally to secondary drive. All files stored in secondary drive should be deleted through scheduler on periodic basis.
- Group 2: This group will have access to all drives and they can store files which are needed for their development work. They are not allowed to store any office documents or pdf files. Scheduler will run every day to wipe out all documents and pdf files irrespective of location including temp folder, downloaded or recycle bin.
- Group 3: These are exception groups. These users should be educated to know data classification and security risk on storing sensitive information. These machines will be monitored on a periodic basis and appropriate action should be taken to ensure compliance.

5 Design Approach

5.1 Workstation Analysis

Installed software and required storage size for applications need to be analyzed for all workstations. Table 3 gives minimum information to be collected from each workstation. This will help to map them into different groups.

5.2 Group Classification

As mentioned in the proposed solution, all workstation should be mapped to particular group. The mapping details can have either list of workstations to be included or exempted from each group.

5.3 Group Policy Creation

Workstations belong to Group 1 and will not have any files which are created by the user. All user files will be deleted on scheduled time from the local drive. This can be achieved through the following settings:

Table 3 Settings for group policy I

Field name	Description	Usage
Hostname	Hostname of workstation	This hostname will be included in GPO
Software	All software installed in the workstation	Helps to analyze whether it has dependency on local storage
Additional storage required for executing software	This will have “Yes” or “No”	To know whether software is needed for additional storage for storing Libraries/configuration settings or API
Additional storage required for the user	Is there any additional space needed for the user to store reference files	This will help us to classify correct group
Projected space required for storing user file	Local storage space required in MB	This is needed for future work to calculate required bandwidth to move files to centralized location

- USB, CD, and primary drive will be hidden normal user and accessible only for admin.
- Only secondary drive will be visible and accessible.
- Files can be saved either in Desktop or secondary drive.
- Data stored in the desktop are automatically redirected to secondary drive.
- Scheduler will run from the workstation and all files will be deleted automatically to ensure local storage restriction as part of security requirement.

Workstations belong to Group 2; specific file groups will be wipeout from local storage. This can be achieved through the following settings:

- USB, CD, and primary drive will be hidden for normal user and accessible only for admin.
- By using administrative template, below options should be disabled.
- Auto recover delay.
- Auto recover save location.
- Keep the last autoSaved version of files for the next session.
- AutoRecover time.
- Save AutoRecover info.
- Default file location.

6 Execution Approach

6.1 Group Policy Object-1

The basic requirement to implement this policy is workstation must have two partitions (Primary and Secondary drive). In addition to basic function listed in Sect. 5, the following sub-functions should also be applied. If the below sub-functions are not applied, the user can do workaround to save file in primary location.

- Hide libraries from Windows explorer (documents, downloads, music, videos, and picture).
- Restrict adding files to the root folder of the user.
- Prohibit users from manually redirecting profile folders.
- Remove “My Documents”, “Recycle bin” from desktop.
- Remove “Documents”, “Music”, “Picture” from startup.
- Remove “Pinned program”, “Recent items”, “Network” from startup.
- Remove properties icon from context menu.
- Restrict access to control panel.

6.2 Group Policy Object-2

This group policy is used for Group 2. To implement GPO to restrict on office files, Microsoft Office administrative template files (ADMX/ADML) is required which are available in Microsoft download center [12]. The policy can be implemented using the steps given below.

- Download ADMX from Microsoft download center.
- Import ADMX to Domain Controller.
- Write PowerShell script in Netlogon to remove all file extensions used for MS Office, Open Office, and pdf.
- This file extension can be customized as per user requirement. The sample code given below is used for removing documents and pdf files alone.

```
$objShell = New-Object -ComObject Shell.Application
$objFolder = $objShell.Namespace(0xA)
$winTemp = get-childitem C:\, D:\ - include *.doc,
*.docx, *.docm, *.dotm, *.odt, *.dotx, *.pdf -recurse
Remove-Item -Recurse $winTemp -Force
```

- This can be extended to remove other types of files as well.
- Configure GPO as mentioned in Sect. 5.3.
- Write batch file to clean files and place them in Windows startup script.
- Include workstation details to GPO.
- Apply GPO and enable scheduler to run on a regular interval.

7 Conclusion and Future Work

To avoid data leakage from customer network where we do not have VDI or citrix infrastructure, it can be achieved through GPO implementation. This is easy to manage and scalable to future expansion without any additional investment. By using existing windows group policy settings, administrative template files, and simple PowerShell scripts and batch files, data leakage from the network can be restricted and compliance to security requirement with zero cost. This approach does not require any change on network model and there is no impact on business continuity. This Group policy can be configured to add additional file types.

As the next step, these policies can be enhanced to include exception on retaining list of files so that all types of files can be wipeout. Also if the user renames the file extension to accepted file types, there is no provision to delete those files through this policy. We need to write a script which reads metadata of file to identify the file type and implement same GPO. This is applicable only for Windows operating system environment and does not apply for other operating systems. We need to find a similar solution for other operating systems.

References

1. <http://www.legislation.gov.uk/ukpga/1998/29/contents>.
2. https://infowatch.com/report2016_half#-DataLeaksource.
3. <https://ico.org.uk/media/for-organisations/guide-to-data-protection-2-9.pdf>.
4. Huth, C. L., Chadwick, D. W., Claycomb, W. R., et al. (2013). *Information system Frontiers-springer US-arch*, 15(1), 1–4.
5. Ieman-Meza, B., Burns, P., Eavenson, M., Palaniswami, D., & Sheth, A. (2005). An Ontological approach to the document access problem of insider threat. In: Kantor P., et al. (Eds.), *Intelligence and Security Informatics*. ISI 2005. Lecture Notes in Computer Science, (Vol 3495). Berlin, Heidelberg: Springer.
6. Clark, D., Hunt, S., & Malacaria, P. (2002) Quantitative analysis of the leakage of confidential data. *Electronic Notes In Theoretical Computer Science*, 59(3), Elsevier.
7. Kale, S. A., & Kulkarni S. V. (2012, November) Data leakage detection. *International Journal of Advanced Research in Computer and Communication Engineering* 1(9).
8. <http://breachlevelindex.com>.
9. <http://breachlevelindex.com/top-data-breaches>.
10. https://www.vmware.com/pdf/vdi_strategy.pdf.
11. <http://searchwindowserver.techtarget.com/definition/Group-Policy-Object>.
12. <https://www.microsoft.com/en-in/download/details.aspx?id=35554>.

Extraction of Character Personas from Novels Using Dependency Trees and POS Tags



Nikhil Prabhu and S. Natarajan

Abstract Novels are a rich source of data for extracting interesting information. Besides the plot, the characters of a novel are its most important elements that shape the story and its message. An interesting task to consider is extracting these characters from novels in the form of the personas they embody. In this paper, we define and introduce a method to extract such personas of characters in fiction novels, in the form of descriptive phrases. These personas are divided into three types of description—facts, states and feelings. We show that such a model performs satisfactorily returning an extraction precision of 91% and average classification accuracy of 80%. The algorithm uses universal dependency trees, POS tags and WordNet to capture semantically meaningful descriptions of characters portrayed. The results have the potential to serve as input for future NLP tasks on literature fiction like character clustering and classification using techniques such as sentence embeddings.

Keywords Universal dependency trees · Part-of-speech tags · WordNet · Natural language processing

1 Introduction

A persona is defined as the aspect of a person's character that is presented to or perceived by others. It can include a person's behaviour in different situations, opinions on different matters, body language and background, among a score of other parameters. In literary fiction, the persona a character displays is much easier to define. Here, the motives, beliefs, behaviour and quirks of a character are explicitly presented, unless intentionally hidden by the author. This paper presents an algorithm that exploits the descriptions of characters given in a novel in order to extract their personas. A persona is defined here as an encapsulation of three facets—facts, states and feelings. Facts include character traits and attributes, (physical, cognitive, or otherwise) as well as indisputable certainties describing the person. Facts could

N. Prabhu (✉) · S. Natarajan
Department of CSE, PES University, Bangalore 560085, India
e-mail: nikhil.prabhu@gmail.com

© Springer Nature Singapore Pte Ltd. 2019
N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_6

be ‘hates exercise’, or ‘She was a squat, smiling woman’. States include physical and emotional states, including ‘howled with laughter’, and ‘ran home as fast as he could’. Feelings refer to directly affective states, both physical and emotional. Examples of valid feelings are ‘awkwardness’, and ‘emptiness’ with phrases such as ‘jealous and angry’, and ‘was hungry’ as descriptions.

2 Background and Related Work

Bamman et al. [1, 2] divided a persona into four categories based purely on typed dependency relations. These are:

- (i) ‘agent’ for verbs that have an (*nsubj*)/agent relation with a character,
- (ii) ‘patient’ for verbs that have a (*dobj*)/*nsubjpass* relation with a character,
- (iii) ‘predicative’ for a noun/adjective that has an (*nsubj*) relation with the character, with an inflection of the word be as a child, and
- (iv) ‘possessive’ for words that have a *poss* relation with a character.

The personas extracted through this mechanism, however, are constituted by only single words, like ‘strangled’, or ‘angry’, including many trivial ones like ‘told’, and ‘came’, which, when they do provide a meaningful pointer to the character’s personality, do so in a very limited sense, lacking context.

3 Proposed System

The philosophy behind the algorithm is that every non-trivial feature worth describing a character is done so with an adjective. The algorithm implements this by finding valid mappings between characters and adjectives in each sentence based on several factors as shown in Step 1. These come together to achieve the goal of the task—extracting concise and comprehensive phrases that serve as descriptions. POS tags and universal dependency trees used are extracted using the Stanford CoreNLP toolkit (Chen and Manning [3], Manning et al. [4], Schuster and Manning [5], Toutanova et al. [6]). The four steps of the algorithm are outlined below.

Step 1: Extracting Valid Mappings

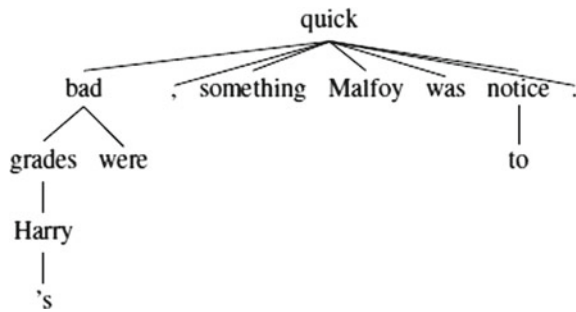
In each sentence containing at least one person and adjective, each possible pair is checked for validity by examining their dependency tree relationships, among other conditions. The four relationships exploited for extraction and classification of phrases are shown in Table 1.

Here, a person is any word with the dependency *nsubj* that refers to a character in the novel. An adjective is any word with the POS tag JJ. In Fig. 1, Harry and Malfoy are persons, while ‘bad’ and ‘quick’ are adjectives. Each mapping that meets the conditions for the relationship it has, as shown in Table 2, is considered valid

Table 1 Valid relationships for a person-adjective mapping

<p>descendant</p> <p>person</p> <p>↓</p> <p>...</p> <p>↓</p> <p>adjective</p>	<p>ancestor</p> <p>adjective</p> <p>↓</p> <p>...</p> <p>↓</p> <p>person</p>
<p>(g)_niece</p> <p>parent</p> <p>↙ ↘</p> <p>... person</p> <p>↓</p> <p>adjective</p>	<p>sibling</p> <p>parent</p> <p>↙ ↘</p> <p>adjective person</p>

Fig. 1 Bare dependency tree for the phrase



and proceeds to the next step. *nsubj* children refer to those nodes that the adjective in a person-adjective mapping actually describes, rather than the person itself. This occurs when an adjective is related to a node as its parent, or as its sibling or niece along with the additional conditions described in Table 3 (Tables 2, 3 and 4 are described in Prolog, some of whose syntax is described in Table 5. These tables are described at the end). Checking for these weed out invalid mappings like that shown in Fig. 1, where the adjective ‘quick’ describes Malfoy instead of Harry, making the extracted phrase ‘quick to notice’ inappropriate for Harry, but apt for Malfoy.

Step 2: Adding Dependents

There are several dependent words which are added to the describing phrase that are vital to the coherence and conciseness of the description returned. These dependents and their conditions of addition are described in Table 6. For example, in the sentence ‘He drifted into an uneasy sleep, thinking about what happened’, adding the parents of the person and the adjective extracts ‘drifted into an uneasy sleep’ instead of just ‘uneasy’, which gives a very different meaning. One particular condition checked

Table 2 Procedure for ensuring validity and classifying type of person-adjective mapping

```

relationship(adjective, person, X):-
  (sibling(adjective, person),
   non_trivial_dep(adjective) ->
    X = sibling;
   (ancestor(adjective, person),
    not(nsubj_child(parent, adjective, NODE)),
    (no_validSplitters(adjective, person, punctuations);
     not(VB_in_lineage(adjective, person)))) -> X = ancestor;
   (g_nib_desc_condition(parent, adjective) ->
    dep(adjective, dobj)) ->
    ((g_niece(adjective, parent) -> X = (g_niece);
     ancestor(parent, adjective) -> X = descendant)).

(g_nib_desc_condition(parent, adjective):-
  (non_trivial_dep(adjective), not(nsubj_child(parent, adjective, NODE))).

VB_in_lineage(A, B):-
  (ancestor(A, B), ancestor(V, B), ancestor(A, V), pos(V, 'VB')).

non_trivial_dep(A):-
  not(dep(A, det); dep(A, case)).

sibling(A, B):- parent(adjective, PA), parent(person, PA).

(g_niece(A, B):- ancestor(B, X), sibling(X, A).

ancestor(A, B):- parent(A, B); parent(A, X), ancestor(X, B).

```

Table 3 Definitions of sub-procedures and terminology used

```

nsubj_child(person, adjective, NODE):-
  (not(same_name(person, NODE)), dep(NODE, nsubj), dep(person, conj)),
  not(parent(person, NODE), (parent(NODE, adjective);
   ((relationship(NODE, P, sibling); parent(NODE, P)),
    (pos(P, 'VB'), no_validSplitters(person, NODE))),
   present_after(person, NODE), parent(adjective, P))).

no_validSplitters(X, Y, PUNCT LIST):-
  (word_in_between(X, Y, Z),
   not(conjunction_splitter(Z); punctuation_splitter(Z, PUNCT LIST))).

conjunction_splitter(NODE):-
  is(NODE, conjunction), adjacent_words_condition(NODE).

punctuation_splitter(NODE, PUNCT LIST):-
  (is(NODE, punctuation), not(descriptive_comma(NODE)),
   adjacent_words_condition(NODE), not(descriptive_pair(PUNCT LIST))).

descriptive_comma(NODE):-
  (punct_type(NODE, comma), dep(NODE, D), member(D, [amod, conj, advcl]),
   pos(NODE, 'RB'); pos(NODE, 'JJ')).

descriptive_pair(PUNCT LIST):-
  (length(PUNCT LIST, 2), type(PUNCT_LIST[0], P1), type(PUNCT_LIST[1], P1)).

adjacent_words_condition(NODE):-
  (previous_word(NODE, P), next_word(NODE, N), not(pos(P, 'RB'), pos(P, 'JJ'), pos(N, 'RB'),
   pos(N, 'JJ')), dep(P, DP), not(dep(N, DP))).

same_name(person, NODE):-
  name(NODE, NM), name(person, NM)).

```

Table 4 Procedure for classifying extracted phrases into fact, state or feeling

```

facet(phrase, X):-
  (person_condition(person) -> X = Fact;
   relationship(adjective, person, sibling) ->
    (feeling_condition -> X = Feeling;
     parent(adjective, P), pos(P, 'VB') ->
      (word_in_between(person, adjective, 'in'),
       no_validSplitters(person, adjective, punctuations);
       root(sentence, adjective) -> X = State; X = Feeling);
     sibling_state_condition -> X = State);
  relationship(adjective, person, ancestor) ->
    (previous_word(adjective, PR), dep(PR, det) -> X = Fact;
     feeling_condition -> X = Feeling;
     desc_state_condition -> X = State; X = Fact);
  (relationship(adjective, person, (g)_niece);
   relationship(adjective, person, descendant)) ->
    feeling_condition -> X = Feeling;
    parent(person, P), pos(P, 'VB') -> X = States; X = Fact).

person_condition(PER):-
  (POS_condition(PER); succeeding_word_condition(PER); have_condition(PER)).

POS_condition(PER):-
  ((parent(PER, P), adjacent_word(PER, A), (pos(P, 'VBZ'); pos(A, 'VBZ')))).

succeeding_word_condition(PER):-
  (succeeding word(PER, S), (inflection(S, 'be'); inflection(S, 'have')),
   (root(sentence, S); parent(S, P), not(pos(P, 'VB')))).

have_condition(PER):-
  ((relationship('have', PER, descendant); relationship('have', P, (g) niece),
   parent(PER, P)), no_validSplitters('have', PER, punctuations)).

feeling_condition:-
  (present_in(adjective, WordNet feelings), (not(dep(P, dobj); pos(P, 'NN')),
   parent(adjective, P)); (inflection(N, feel), nearby word(adjective, N))).

sibling_state_condition :-
  (not(dep(adjective, dep)), child(adjective, C), not(dep(C, nsubj)),
   (dep(adjective; nmod) -> parent(adjective, P), not(pos(P, 'JJ')))).

desc_state_condition:-
  (child(adjective, C), not((inflection(C, 'be'); inflection(C, 'have')));
   root(sentence, adjective); no_validSplitters(person, adjective, punctuations),
   word_in_between(person, adjective, 'in'); pos(adjacent words(adjective), 'VBG');
   adjacent_words(adjective, A), pos(A, 'VBD')).

```

Table 5 Prolog syntax

<p>X ; Y => Try X first; if it fails, try Y.</p> <p>X -> Y => If X, then try Y, otherwise fail.</p> <p>X -> Y ; X => If X, then try Y, else try Z</p> <p>X, Y => Try X, then try Y</p>
--

for is the presence of ‘splitters’ in the form of punctuations or conjunctions that split the phrase, eliminating redundant details. A conjunction that splits a phrase into two parts—of which one part describes something other than the ‘person’, is considered as a valid splitter. A descriptive conjunction that has its adjacent words as describers, like ‘and’ in ‘kind and sweet’, is invalid. A punctuation may similarly split a phrase but will be considered invalid if it is either a descriptive comma, an aside-describing identical pair, or fails the conditions for a conjunction splitter described in Table 3. Examples of invalid punctuation splitters are present in ‘good, honest, thoughtful man’, and ‘Bob, sitting on the chair, laughed’, with the first containing descriptive commas and the second being a descriptive pair.

Step 3: Refinement

The words extracted till now may not form a contiguous phrase, so any gaps are filled. The resulting phrase is divided into sub-phrases by valid punctuation splitters, if present. The sub-phrases that satisfy any of the following conditions are then added to the description:

- (i) It contains the adjective of the mapping.
- (ii) It immediately follows the person of the mapping.
- (iii) The first word in the sub-phrase is the person.

Again, any gaps present are filled and any stop words present at the ends are removed, giving the final description.

Step 4: Classification

The procedure for classification shown in Table 4 follows from validity checking, where three of the initial conditions for classification follow from the type of relationship between the person and adjective as found in Table 1, along with a new condition *person_condition(PER)* on words directly related to the person in the mapping. *WordNet_feelings* is a set used in the decision to classify a phrase as a feeling. It is the set of all the hyponyms of the synset *feeling.n.01* present in the WordNet database. Checking if a word is present here is complicated by the fact that adjectives are stored as clusters rather than trees, making a direct search ineffective. Table 7 describes a procedure that uses the presence of lemmas present for each word to check if a word is present in this set.

Table 6 List of dependents to be added to the descriptive phrase

1	All the words between the person and the adjective if the dependency of the adjective is <i>xcomp</i>
2	All negative words that are either <ol style="list-style-type: none"> i. child or sibling of the adjective, or ii. are present between the person and the adjective in the sentence
3	<ol style="list-style-type: none"> i. All words with dependency <i>nsubj</i> that don't represent a character and do not have POS tag based on any of PRP, NNP, WP, DT ii. All words with POS tags VBZ, VBD or VBG that are adjacent to the person iii. All nodes with the POS tag JJ and the dependency <i>advcl</i> or <i>acl:relcl</i>, if they are either siblings or children of the adjective or the person, respectively
4	All valid prepositions—children of the adjective with POS tag IN that don't have the dependency mark
5	<ol style="list-style-type: none"> i. All adverb children of the adjective ii. All siblings or children of the adjective that have either <i>advmod</i> or <i>det</i> dependency which are preceded by a word having an an inflection of 'be' or 'have'
6	<ol style="list-style-type: none"> i. All <i>nmod</i> nodes that are either children, siblings, or parents of the adjective ii. All <i>conj</i> children related in the same way as long as they don't have POS tags VB or VBD iii. In addition, all prepositions that mark these <i>nmod</i> or <i>conj</i> words, for each of these found
7	<ol style="list-style-type: none"> i. The parent of the person if it has POS tag VB and there are no valid punctuation splitters between it and its parent. (Table 3, <i>punctuation_splitter(X, Y, PUNCT LIST)</i>) ii. The parent of the adjective if the adjective's dependency is any of <i>nmod</i>, <i>conj</i>, <i>xcomp</i>, <i>dobj</i>, <i>amod</i>, <i>advmod</i> and there are no valid splitters between it and its parent
8.	<ol style="list-style-type: none"> i. All valid prepositions that are either siblings or children of the adjective, and ii. all words with dependency <i>cc</i> that are either <ol style="list-style-type: none"> (a) related to the adjective in the same way or (b) are the parent of any node added to the list till now, as long as the words are present after the person and the adjective in the sentence
9	All punctuations as long as the only punctuations extracted are not an aside-describing identical pair, (Table 3, <i>descriptive pair(PUNCT LIST)</i>) as well as all 'as...as' comparative pairs
10	<ol style="list-style-type: none"> i. All words with dependency <i>neg</i> that are parents of any node in the list, and ii. All <i>neg</i> words present before the adjective as long as the adjective has no valid splitters between it and its parent, and the person in the mapping is not the first person described in the sentence
11	All children nodes of nodes in the list that have either <ol style="list-style-type: none"> i. dependencies <i>xcomp</i>, <i>ccomp</i>, <i>nmod</i>, <i>conj</i>, <i>dobj</i>, <i>cc</i> or <i>aux</i>, or ii. have POS tags VB, RB, WRB, extracted recursively until there are no more to be added

Table 7 Procedure for checking if a word is present in WordNet_feelings

1	Get the synsets of the word, and convert each to its corresponding lemma and store them in a list
2	For each lemma, get its derivationally related forms and store them along with the existing lemmas in the list
3	Convert each lemma in the list to its corresponding synset, if it exists
4	If any of the synsets are present in <i>WordNet_feelings</i> , the word is considered present. If not, extract synsets of words

Table 8 Performance of the algorithm

Task	Precision	Recall	Negative predicate value	Specificity	F1 score	Accuracy
Validity of phrase	0.91	0.62	0.39	0.77	0.74	0.67
Classification as fact	0.66	0.53	0.85	0.91	0.55	0.80
Classification as state	0.75	0.86	0.73	0.58	0.79	0.73
Classification as feeling	0.60	0.48	0.91	0.95	0.53	0.88
Average classification	0.67	0.62	0.83	0.81	0.63	0.80

4 Results and Inferences

The algorithm was evaluated using an unbiased human evaluator who marked a randomly generated subset containing both valid phrases (extracted by the algorithm) and invalid ones (phrases filtered out by the conditions of the algorithms as in Table 4) for the books ‘A Picture of Dorian Grey’ and ‘The Adventures of Huckleberry Finn’. Each phrase was marked as valid or invalid, and if valid, it was marked as one class out of Fact, State or Feeling. The total number of phrases evaluated amounted to 985. To perform the evaluation, one must be able to distinguish whether a phrase is describing a given person or not. In addition, one must be able to confidently state whether such a description falls under the category of a fact, state or feeling as clearly distinguished in the Introduction. As this task would be simple for a person with sufficient command of the English language and would bring very few ambiguities that need to be resolved, one evaluator was considered sufficient for the purpose of evaluation. Table 8 shows the results of the evaluation of validity of phrase extraction and phrase classification. The precision of phrase validity was 91% while the average accuracy of classification was 80%. Table 9 shows a handpicked subset of phrases that represent the persona of the character Dorian Grey from the novel ‘The Picture of Dorian Grey’, as extracted by the algorithm.

Table 9 Sample phrases from the persona of the character ‘Dorian Grey’

Facts	Is never more present in my work than when no image of him is there As a rule, he is charming to me Now and then, however, he is horribly thoughtless Seems to take a real delight in giving me pain Is my dearest friend Has a simple and a beautiful nature Does not think his natural thoughts, or burn with his natural passions Was brilliant, fantastic, irresponsible But he has been most pleasant; is far too wise not to do foolish things now and then
States	Burying his face in the great cool lilac-blossoms Saw it creeping into the stained trumpet of a Tyrian convolvulus; murmured, flushing at his own Boldness Had recognized himself for the first time Stood there motionless and in wonder, dimly conscious that Hallward was speaking to him; was Reclining in a luxurious arm-chair Will tell you more Answered in his slow melodious voice Replied, touching the thin stem of his glass with his pale, fine-pointed fingers Had grown years older Grew pale as he watched her
Feelings	Far too charming But he felt afraid of him, and Ashamed of being afraid Seemed quite angry Was puzzled and anxious Looked pale, and proud, and indifferent Felt that he was on the brink of a horrible danger

5 Conclusion and Future Work

Extracting semantic summaries from the text is a useful task, and the results of the algorithm described can be used as a basis for further literary and computational analysis. Skip-thought vectors (Kiros et al. 2015) [7] are representations of sentences in the form of embeddings similar to word embeddings. These represent sentences in the form of vectors where a sentence is represented by the sentences surrounding it. They have been used to successfully perform tasks of semantic relatedness, paraphrase detection and classification. This model can be used on the personas extracted for the purpose of a variety of tasks. These can include clustering similar personas of characters across novels, classifying character personas into a set of archetypes.

Acknowledgements We would like to thank Ms. V. Sruthi for her help in evaluation of the 985 phrases present in the test set for the purpose of determining the performance of the algorithm.

References

1. Bamman, D., O'Connor, B., & Smith, N. A. (2014a). Learning latent personas of film characters. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)* (p. 352).
2. Bamman, D., Underwood, T., & Smith, N. A. (2014). A Bayesian mixed effects model of literary character. *ACL, 1*, 370–379.
3. Chen, D., & Manning, C. D. (2014). A fast and accurate dependency parser using neural networks. In: *EMNLP* (pp. 740–750).
4. Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & Mc-Closky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *ACL (System Demonstrations)* (pp. 55–60).
5. Schuster, S., & Manning, C. D. (2016). Enhanced English universal dependencies: An improved representation for natural language understanding tasks. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*.
6. Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. Association for Computational Linguistics* (pp. 173–180).
7. Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Skip-thought vectors. In *Advances in neural information processing systems* (pp. 3294–3302).

A Multifactor Authentication Model to Mitigate the Phishing Attack of E-Service Systems from Bangladesh Perspective



Md. Zahid Hasan, Abdus Sattar, Arif Mahmud and Khalid Hasan Talukder

Abstract A new multifactor authentication model has been proposed for Bangladesh taking cost-effectiveness in primary concern. We considered two-factor authentications in our previous e-service models which were proven to be insufficient in terms of phishing attack. Users often fail to identify phishing site and provide confidential information unintentionally, resulting in a successful phishing attempt. As a result, phishing can be considered as one of the most serious issues and required to be addressed and mitigated. Three factors were included to form multifactor authentication, namely, user ID, secured image with caption, and one-time password. Through the survey, the proposed multifactor model is proven to be better by 59% points for total users which comprises 55% points for technical users and 64% points for non-technical users in comparison to traditional two-factor authentication model. Since the results and recommendations from the user were reflected in the model, user satisfaction was achieved.

Keywords Phishing attack · E-banking · E-service · Online banking

1 Introduction

At present, e-services can be considered as the most substantial concerns in our day-to-day life. The efficient utilizations of computerized knowledge and tools expand along with our requests. However, the ideas and tools will not be adequate to upkeep novel technologies with the hi-tech devices in modern e-service system.

Md. Zahid Hasan (✉) · A. Sattar · A. Mahmud · K. H. Talukder
Daffodil International University, 102 Sukrabadh, Mirpur Road, Dhaka 1207, Bangladesh
e-mail: zahid.cse@diu.edu.bd

A. Sattar
e-mail: abdus.cse@diu.edu.bd

A. Mahmud
e-mail: arif.cse@diu.edu.bd

K. H. Talukder
e-mail: cse.khalid@gmail.com

We proposed models for two e-services, namely, e-healthcare [1] and mobile payment [2] system that guarantees accomplishing the objectives and requirements, as supported by our previously suggested framework models [3]. Besides, the models can have an important role in e-commerce sector which can lead to a giant accomplishment in terms of upgrading of e-care services. However, the suggested models should be secure and reliable in terms of prevention of phishing attack which has become a concern now a day.

Phishing is such kind of attack which exploits the weakness of end user of a system [4]. A huge number of internet users cannot able to differentiate phishing and original site, and they ignore passive warnings like toolbar indicator [5]. The aim of phishing is to steal sensitive information like password, credit card information from Internet user, etc. Therefore, authentication needs to be ensured in order to reduce phishing attack.

Most of the e-services in Bangladesh use two-factor authentications (2FA) which are insufficient to defend the phishing attack. As a result, multifactor authentication needs to be implemented. Numerous security factors are being used for authentication in economically advanced countries but it is not viable in a developing country like Bangladesh where the cost-effectiveness in authentication is a challenging task.

The main purpose of a multifactor authentication solution is to verify the identity of the user by other means apart from the traditional username and password authentication. These factors are usually combined with a PIN code that generates one-time passwords (OTP). Based on the earlier papers, we have investigated the login behavior of users and how the user compromises their login credentials with phishing site through the survey. Along with this, we proposed a new authentication mechanism in the context of developing country like Bangladesh to support our previously developed models. The specific objectives of this research can be listed as follows:

- To propose a new multifactor authentication mechanism being supported by previously proposed e-service model.
- To exploit the login behavior and to figure out the user's experience after applying the proposed approach.

This model can be applied to stop deceitful emails, texts, and copycat websites which share valuable information such as social security numbers, account numbers, login IDs, and passwords. Besides, it will be beneficial in prevention of stealing money and identity or both.

This paper is designed in the following way: Sect. 2 illustrates the background; Sect. 3 clarifies the previously proposed models; Sect. 4 describes the proposed MFA model; Sect. 5 illustrates the results; the discussions and conclusions are added in Sect. 6.

2 Background

Phishing can be considered as the social engineering attack which targets at manipulating the limitation of the system procedure created by system users [4]. As for illustration, a system might be technically safe against the attack like password theft but passwords might get leaked from the end user when an attacker insists to update their respective passwords through a fake HTTP link. So, this phenomenon threatens the complete system security. In addition, we can take DNS cache poisoning, for example, which can also be utilized by the attackers to create more influence on social engineering messages [4].

According to Weider [6], the major motives behind phishing attack are as follows:

- i. Financial benefit: Phishers can utilize the users' banking credentials to gain financial profits.
- ii. Identity hiding: Phishers can advertise the users' identities in the market and can be used later by criminals who hide their authentication and activities from the authority.
- iii. Reputation and disrepute: Phishers can attack the victims for the peer recognition.

Due to abovementioned reasons, phishing attack can be considered as serious one and an effectual lessening will be required.

Multifactor authentication involves the use of two or more independent security factors to authenticate an information system and to lessen phishing attack. Unfortunately, several technological, economical, and usability limitations exist that resist sector-wide acceptance of MFA.

3 Previous Model

We have anticipated frameworks for e-care systems in our earlier works those minimize the misinterpretation about design, development, and communication systems and develop the harmonization among data, service, and management. E-services can be considered as one of the most important factors, and the projected framework ensures the ease of access of services anytime, anywhere without being delimited by any periphery. These two models are described below in brief.

3.1 Contextual Mobile Payment System Implementation

Five players or entities are included in the projected system shown in Fig. 1 named user, network operator, merchants, bank, and cell phone manufacturer that are placed in the management level of ICTization model. Integration of mobile payment system

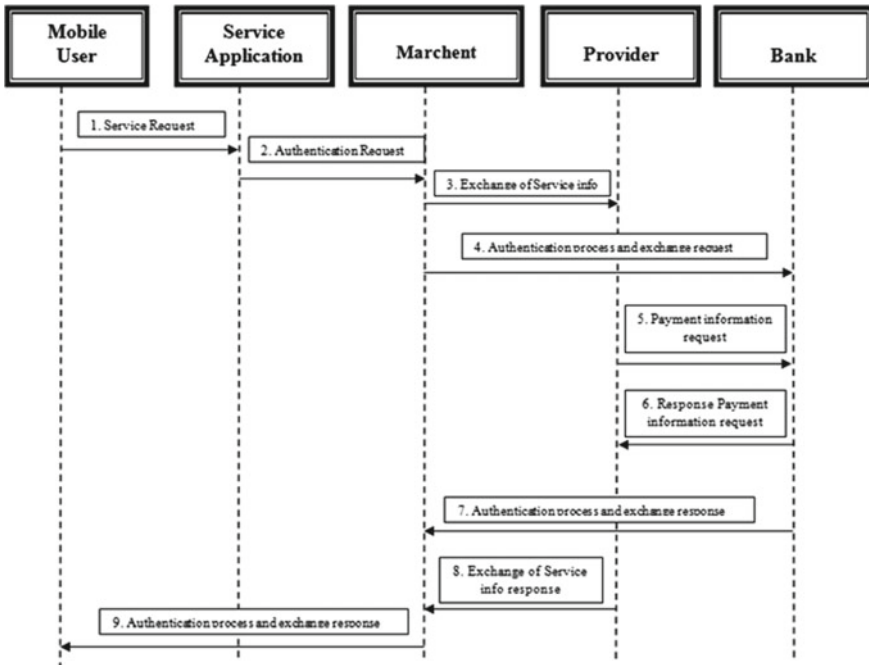


Fig. 1 Contextual mobile payment system [3]

entities is the significant factor to develop a global open solution model in lieu of a closed system given a small scope. The functionalities of these entities are as briefly followed:

- Customers or mobile users are capable to purchase the products from merchants and can make transaction even with others through mobile phones.
- The merchant play the role of a middleman who provides the requested services for the users.
- The network operators are responsible for the network connectivity in mobile devices and manage the user’s authenticity.
- Bank plays the role of a financial institute that is responsible for payment procedure and monitors the flow of transaction.
- Device manufacturers are responsible to develop a generic ground for payment functions as the demand of users.

The projected model allows subscribers to buy products, balance inquiry, and money transfer to bank accounts through mobile SMS. It will enable users to check the transaction status and history such as the amount is left to use, the amount has been spent in the last day, week, and month, transaction acknowledgement, balance notification, etc.

For instance, when a subscriber desires to buy a transportation card, he/she can use mobile phone to pay directly from his/her own bank account. According to the steps in Fig. 8, user plans to obtain any services and/or goods utilizing mobile devices with projected application from different kinds of services integrated with the application. The apps/icons which are included in the mobile phones will be used to provide the desired facilities to the consumer. Users pay for the applications or services and proceed to advance activities. The procedure of these sending and receiving services is happened between the user and the provider where merchant acts as the third party who becomes connected with both ends and ensures authentication.

If a consumer desires to pay for the products and/or services, merchant will inform the bank about the particular client and provide support to accomplish the financial transaction. Afterward, the bank will conclude the financial transaction with consumer and provider and inform the merchant to give the specific products or services to the client. At last, the merchant will offer the services or products to the consumer and the consumer can consume the products and/or services without any disturbance.

3.2 Contextual E-Healthcare Model

The anticipated model supports ambient healthcare services as supported by our formerly planned model frameworks. Five entities are included in our system as shown in Figs. 3 and 5 named as patient, merchants, home network, network operators, and healthcare centers. The functions of these entities are given below in brief:

Sensors are integrated into patients and patients can utilize their home networks for requesting and receiving the desired services. The home network can be considered as the combination of mobile devices like cell phone, laptop, and PDA with Internet functionalities. The merchants plays the role of a middleman to define and provide the services as requested by the patients. They will also communicate with banks regarding financial transactions. The network operators like mobile operators and ISPs will provide the network connectivity to mobile devices and manages the subscriber's authenticity (Fig. 2).

The healthcare service centers can be considered as the collection of hospitals and diagnostics center together with expert physicians and researchers. These centers are scattered in several cities and countries as well. To be noted, the patient information (located in databases) is shared among these centers. As for illustration, patients can appeal the services through their home network both manually and automatically. If we consider the automatic service, the sensors are used to capture the raw data like heart bit, pulse rate, motion, etc. and transmit the data to the home network. On the contrary, patients can utilize their home networks to demand explicit services manually where home network forwards the data to the merchant in order to verify the validity and authenticity of the client. The data can be sent in two ways such as IP traffic and mobile traffic as chosen by the patient. As a result, user can demand the services both using email and cell phones.

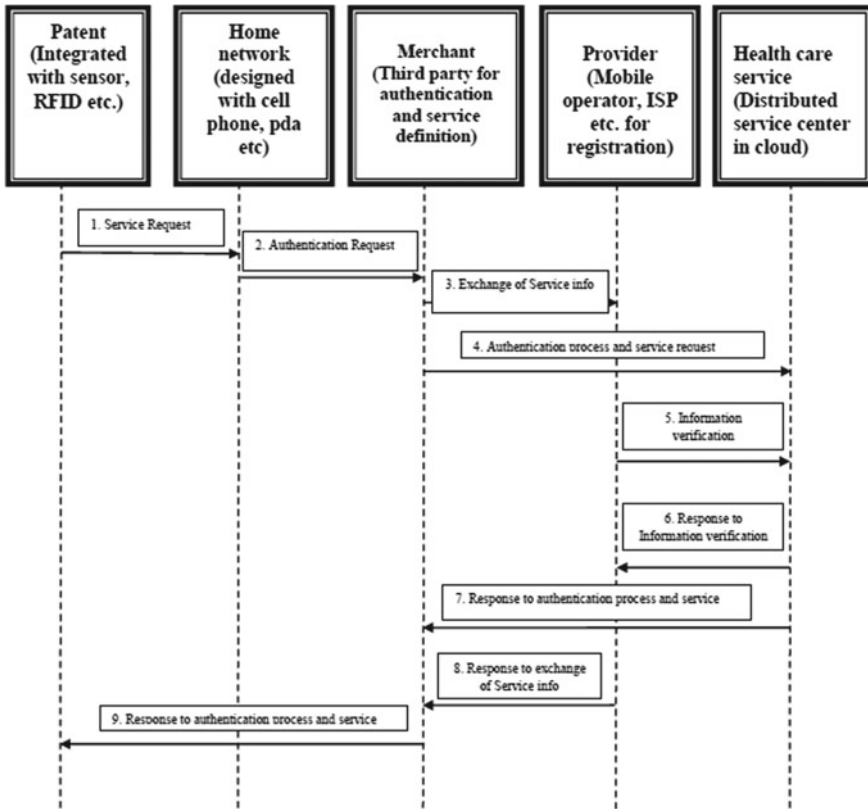


Fig. 2 Contextual e-healthcare model [2]

The procedure of sending and receiving services and application happened between consumer and provider, and the merchant acts as the third party that communicates with both sides and ensure authentication. The merchant can communicate with banks to meet financial transactions as well. As for illustration, if a consumer needs to pay for any registrations or services, merchant can inform the bank that the specific user wants to accomplish the financial transaction. Afterward, the bank will conclude the financial transaction with client and provider and inform the merchant in order to grant the specific service to the consumer. Next, merchant will inform the consumer. To be noted, the healthcare centers sustain an interaction with both service provider and merchant in order to confirm the legitimacy of the client. At the end, the demanded services will be received to the consumer through the merchant.

To be noted, we considered 2FA in these proposed models which is proven to be insufficient in terms of security. However, phishing attack is one of the main concerns for these models which need to be addressed and mitigated. In addition, additional cost to ensure authentication will affect the utilization of these models.

4 Proposed Multifactor Model

Most of the e-services in Bangladesh use 2FA by combining knowledge base and one-time password (OTP) base together. Here, we have combined three factors together to authenticate a user as shown in Fig. 3. An extra factor, security image with caption for defending the phishing attack was included.

Figure 3 describes the authentication procedure between a user and authentication server. At first, user puts the email ID on the site. According to user ID, server replies his/her security image and caption. If the client verifies the image and caption as given through registration time, password is put. Phishers can send similar login page via email, SMS, or other media, but they cannot send security image and caption. When server confirms a valid password, OTP is provided to the client. After the successful OTP reply, authentication process will be completed.

Figure 4 shows the process flow of the authentication approach. When a user needs to login into the site, he/she enters the username at first. According to the username, the site displays the predefined image and caption which was set by the user during the registration phase. If the image and caption match with previous one, the password is put afterward.

Fig. 3 Authentication procedure flowchart

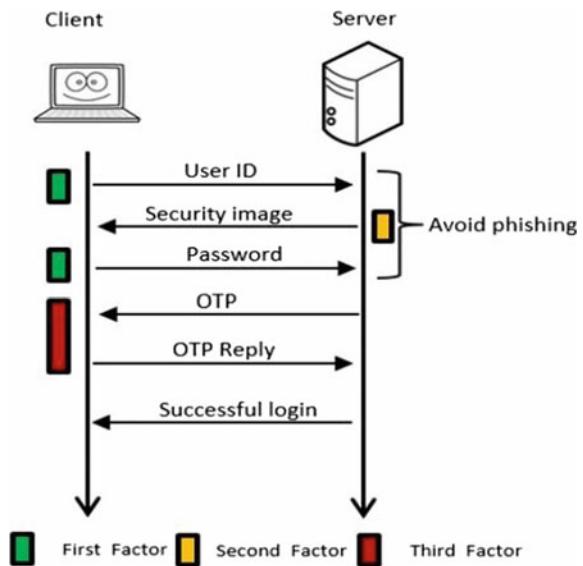
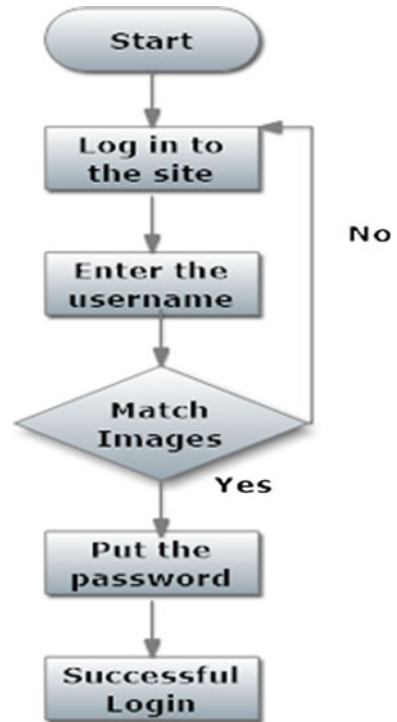


Fig. 4 Authentication procedure



5 Result

As a part of our research, we made the survey on two types of users: technical and nontechnical.

- Technical: The students who have knowledge on computer applications
- Nontechnical: The students who do not have much knowledge on computer applications.

52 students were randomly selected from different faculties where 27 of them were technical students and 25 nontechnical students. Two different look-a-like sites for login were provided where one is valid site and another is phishing site with different URLs. After that, data were collected from user who logged into the phishing site or who denied to login. We made the survey in two different phases: (1) Before registration and (2) after registration. Before registration, as can be seen from Fig. 5, 72% of the total students were unable to identify the phishing site which comprises 62% technical and 80% nontechnical students.

After that, users have to complete the registration. From Fig. 6, we can see users have to register themselves with their name, email ID, and password. Most importantly, users have to select their own image and caption which will be used for authentication purpose.

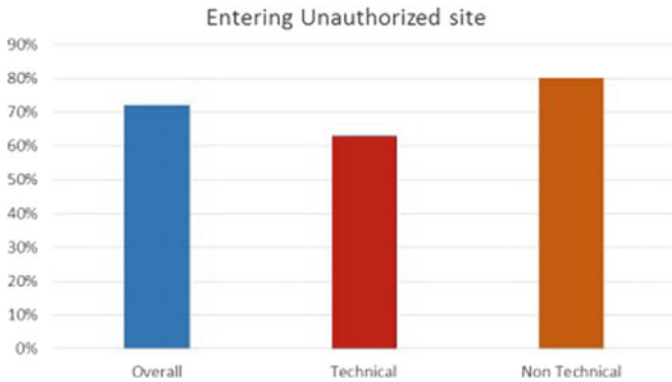


Fig. 5 Phishing result with 2FA model

Fig. 6 Registration procedure

The registration form is titled 'Register Form' in green. It contains the following fields and elements from top to bottom: a text input for 'name', a text input for 'email address', a text input for 'password', a text input for 'Caption', and a file upload section. The file upload section includes the text 'Image size (130*130)', a 'Browse...' button, and the text 'No file selected.'. At the bottom of the form is a large green 'CREATE' button. Below the form, there is a link that says 'Already registered? Sign In'.

Thereafter, the security image is verified in three steps.

Step 1: The registered user gives his/her user ID, as shown in Fig. 7.

Step 2: Server returns his/her security image and caption (Fig. 8) according to the email ID.

Step 3: The user confirms security image and password will be required to enter the desired site.

After registration, it can be seen from Fig. 9 that only 13% of total students failed to identify the phishing which includes 7% technical and 16% nontechnical students.

Fig. 7 Enter only user ID

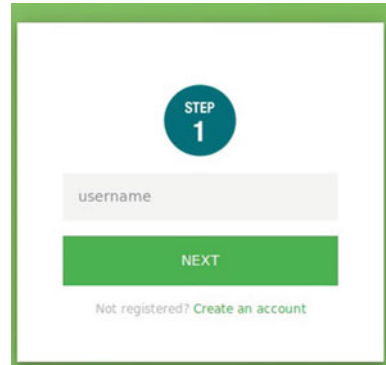
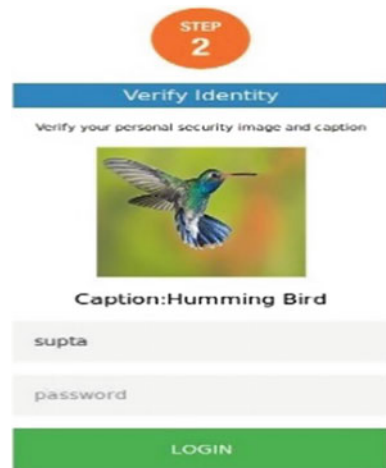


Fig. 8 Security image



If we compare our multifactor authentication model with tradition 2FA model, the result is improved by 59% points for total students, 55% points for technical and 64% points for nontechnical students.

Next, we made an another survey based on two factors:

- i. Agreeable: Users find this model effective and would like to continue.
- ii. Annoying: Users find this model complicated and denied to continue.

Based on Fig. 10, it can be stated that most of the users find this model an efficient one to prevent phishing attack. 88% of total students agreed to continue where only 11% denied. In addition, the difference between agreeable and annoying is 84% for technical and 66% for nontechnical students.

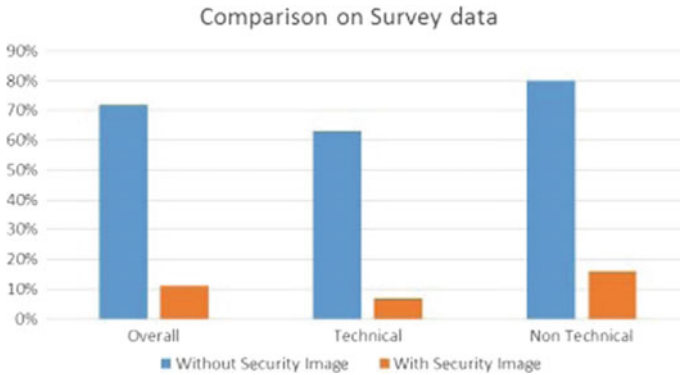


Fig. 9 Comparison between 2FA and MFA model

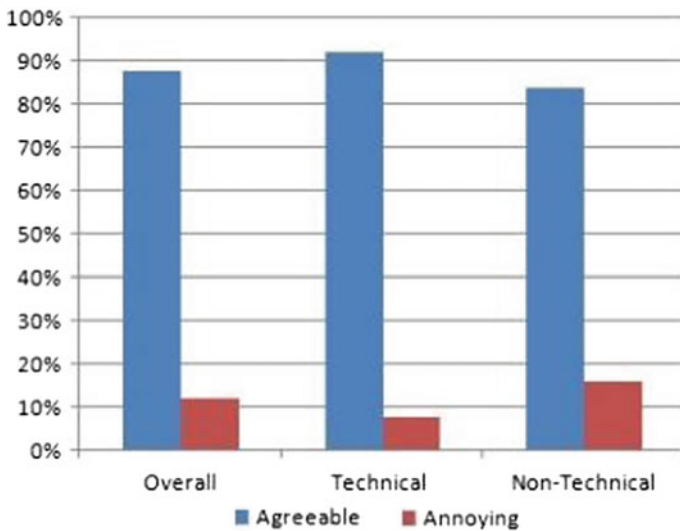


Fig. 10 Comparison between agreeable and annoying

6 Discussion and Conclusion

This paper is developed on previously developed e-service model which can extend the effort on enlightening depiction of ICT setup. Taking the business into account, this paper will assist in providing a system to categorize detailed collection of ICT infrastructural features to achieve a secured authentication. On contrary, from the social standpoint, this paper will offer a collective framework with a goal to obtain secured citizen concerned services. We used 2FA in our earlier proposed e-service model. To be noted, password is the key element for single-factor authentication which cannot be considered as secures in the present e-commerce and Internet world.

Attackers can use sophisticated applications to find out easy guessing password like name, birthdate, phone number, etc. 2FA techniques are currently being used by most of the e-services in Bangladesh. The combination of knowledge base and possession base factor like one-time password (OTP) also cannot avoid phishing attack. In this paper, by carrying out survey, we collect login behavior of 52 students with phishing site. Using survey data, we have found that 72% of total students compromise their password with phishing site and then we propose three-factor authentication mechanisms by combining existing 2FA with security image.

The proposed multifactor model is experienced to be better than traditional 2FA model by 55 and 64% points in case of technical and nontechnical students. Importantly, most users have found this model effective and have decided to continue with the suggested approach. In short, this paper aimed to validate previously proposed models in terms of authentication without adding any extra cost taking developing country like Bangladesh in concern. Besides, we also addressed the user participation in accepting the multifactor authentication model. Therefore, surveys were conducted to compare the traditional 2FA models with the proposed multifactor authentication model. The outcomes and suggestions are set in the designed model where the cost-effectiveness was one of the primary concerns. Besides, the execution of this model can play noteworthy part in m-commerce that will have an enormous success in terms of authentication of e-service system in Bangladesh.

References

1. Mahmud, A., & Sattar, A. (2013). 'ICTization framework': A conceptual development model through ICT modernization in Bangladesh, Published. In *Advanced Computer Science Applications and Technologies (ACSAT), 2013 International Conference, Malaysia*. 23–24 Dec. 2013, 19 June 2014, 978-1-4799-2758-6, Publisher: IEEE.
2. Mahmud, A., & Sattar, A. (2014). Deployment of contextual mobile payment system: A prospective e-service based on ICTization framework from Bangladesh perspective. In *Proceedings of the International Conference on Advances in Computer Science and Electronics Engineering—CSEE 2014*, Copyright © Institute of Research Engineers and Doctors. All rights reserved. ISBN: 978-1-63248-000-2.
3. Mahmud, A., & Sattar, A. (2016) Deployment of contextual E-healthcare system: A prospective e-service based on context aware conceptual framework and ICTization framework model. In *2016 IEEE 11th Conference*, Hefei, China, 5–7 June 2016/24 October 2016, Electronic ISBN: 978-1-4673-8644-9, 978-1-5090-2605-0, Publisher: IEEE.
4. Khonji, M., Iraqi, Y., & Jones, A. (2013). Phishing detection: A literature survey. *IEEE Communications Surveys & Tutorials*, 15, 2091–2121.
5. HUANG, H., TAN, J. & LIU, L. (2009). Countermeasure techniques for deceptive phishing attack. In *International Conference on, 2009. New Trends in Information and Service Science*, NISS'09. IEEE, pp. 636–641.
6. Weider, et al. (2008). A phishing vulnerability analysis of web based Systems. In *IEEE Symposium on Computers and Communications*. ISCC 2008. 2008. IEEE, 326–331.

Study on Energy-Efficient and Lifetime-Enhanced Clustering Algorithm-Based Routing Protocols in Wireless Sensor Network



Ishita Banerjee and P. Madhumathy

Abstract The sensor nodes in wireless sensor network (WSN) come with limited battery life, thus energy usage in WSN is to be handled with great care. The routing protocols are the major research area to work on the QoS of the network such as network lifetime, scalability, energy consumption, packet overhead, etc. Since sensor node possesses limited battery life, to use nodes efficiently yet not losing connectivity in the network becomes a major issue in designing the routing protocols. To achieve energy efficiency and better network lifetime, grouping of sensor nodes into small clusters and selecting one cluster head seems to have more advantages in comparison with other network models to get better scalability, robustness and end-to-end data delivery. In this paper, we have discussed and studied the different energy-efficient routing protocols for clustering of sensor nodes in wireless sensor network, its merits, demerits and applications.

Keywords WSN · Energy-efficient routing protocols · Clustering algorithms

1 Introduction

Wireless sensor network (WSN) is a widely used area of communication in recent days that faces few networking issues while implementation such as different application fields, limited resources availability and functionality, packet dimensions, and multi-hop transmissions for mobile users in dynamic scenario. Applications of WSN are widely spread and hardly left any area untouched such as environmental and habitat monitoring [1] such as water quality [2], river/flood monitoring [3], fire detection and rainfall observation in the field of agriculture, and also extends its applications in the field of military [4], coal mines [5], education [6], health monitoring [7] as well as medical diagnostics, and many more. Wireless sensor network basically consists of nodes which might have sensors to gather data. A common working of such WSN

I. Banerjee (✉) · P. Madhumathy
Department of ECE, Dayananda Sagar Academy of Technology and Management,
Bangalore 560082, India
e-mail: paul.ishita@gmail.com

is collection or generation of data in the sources and sending the data to the intended sinks through wireless communication. This whole network is based on few intermediate nodes. The wireless communication that takes place here is through radio which is energy-expensive method. Effective use of this kind of transmission causes the nodes to loose energy and eventually resulting in data collection loss and loss or delayed data delivery. Though loss of few nodes is tolerable, loss of some highly dependable or critical node may cause serious harm to the end-to-end connection.

Routing protocol classifications are based on functioning mode, participation style and network structure. In the study of Li et al. [8], on the basis of functioning mode, it can be subdivided into proactive routing protocols, reactive routing protocols and hybrid routing protocols [9]. In the first case, i.e. proactive routing protocol, data is sent to the base station through predefined path. In reactive routing protocol, path is established on demand. Hybrid types already establish path initially but modify according to the need for the improvement of the communication. On the basis of network structure, it can be subdivided into data-centric routing protocol, hierarchical routing protocol and location-based routing protocols. The data-centric routing protocols are sink initiated. Hierarchical routing protocols [10] are cluster-based, energy-efficient routing protocols where low-energy nodes capture data and high-energy nodes process and transmit data. Location-based routing protocols forward data at each hop to the neighbour node which is geographically located nearest to the sink. Here, the location of the nodes is to be known using GPS to find the optimum path. Depending on participation style, the protocols are classified as direct communication, diffusion-based and clustering-based. In the study of Kaur and Kad [11] and Yi [12], closely classifying, following are the protocols that are used according to the way the nodes participate in transferring data to the destination.

- Direct communication (base station and nodes can communicate to each other directly).
- Diffusion-based (this type of algorithm uses location data).
- E3D (majorly diffusion-based, uses location data as well as residual energy, node utilization).
- Random clustering (chose cluster head randomly, collect all node data present in that cluster and forward to base station).
- Optimum clustering algorithm (based on physical location and residual energy, cluster head formation strategy changes after few rounds of communication for optimum cluster formation).

The major issue in designing any wireless sensor architecture is limited battery life of the sensor nodes which link the communication process. For effortless end-to-end data delivery, increased lifetime of the network is required which could be achieved by grouping few nodes and selecting one cluster head rather than randomly sending the data directly through available nodes to the base station. As soon as a cluster is formed and a cluster head is selected, the cluster head is in terms of maintaining the data of each node in that cluster and is responsible for packet transmission to the destination or sink node. In this paper, we shall discuss few energy-efficient routing protocols for cluster-based network of nodes in WSN.

2 Related Work

The few popular clustering-based routing protocols such as LEACH [13], PEGASIS [14] and HEED [15] possess their own advantages and disadvantages in terms of load balancing, lifetime of network, scalability and fields of application. LEACH and HEED routing protocols are the clustering protocols which work in homogenous WSN, whereas BEENISH works in heterogeneous platform. BEENISH utilizes four different advanced energy levels improvising the previous protocols such as the two-level heterogeneous ones like stable election protocol (SEP), DEEC [16] and DDEEC [17], and also the three-level ones like EDEEC [18]. If network lifetime or throughput is considered, then iBEENISH performs better than BEENISH. MBEENISH and iMBEENISH give better results than BEENISH and iBEENISH if we consider sink mobility model facilities [19]. An energy-efficient routing protocol (EERP) is implemented to maximize network lifetime of WSNs using an optimal aggregated cost function and A* algorithm [20]. In this paper, the authors have selected optimal and shortest path between source and sink based on residual energy of sensor nodes, free buffer of the nodes and link quality between sensor nodes. Here, A* algorithm [21] is implemented to select optimal path to process the above-mentioned parameters. Another improved routing protocol named EE-LEACH [22] yields better performance in comparison with the LEACH by increasing packet delivery ratio, reducing end-to-end delay, optimizing energy consumption and thus increasing network lifetime. Cluster-based ACO BAN [23] creates a routing table from where it selects minimum cost function value to find the optimal path in between source and sink and provides less overhead, minimum number of packets delivered within intermediate nodes and high data transmission rate. The protocol PDORP [24] uses the characteristics of efficient gathering of sensor information system. It also uses PEGASIS and DSR [25] routing protocols for reduction of energy consumption and network overhead and faster response time, and ensures better connectivity of nodes. The hybridization of genetic algorithm along with bacterial foraging optimization [26, 27] used by PDORP proposes the optimal path. The E2HRC [28] routing protocol effectively balances the energy usage of wireless sensor network, thus decreasing energy consumption of nodes as well as reduced numbers of control messages. Load-balancing cluster-based protocol (LCP) [29] is a modified version of HEED protocol to increase the energy efficiency of the network. A reliable, energy-balancing, multi-group routing protocol (REM) [30] is implemented in the field, and data at critical time is to be transmitted to the fire station real time with least delay to save the crisis situation. Another routing protocol discussed here is PECE [31] that promises better energy balancing of the network.

3 Study of Different Routing Protocols for Clustering-Based Design of Sensor Nodes

Here, we have discussed few clustering-based algorithm-based routing protocols for wireless sensor network which all promise to be energy-efficient and also increases network lifetime to provide better connectivity.

3.1 *BEENISH*

In this paper, BEENISH, iBEENISH, MBEENISH and iMBEENISH protocols for heterogeneous wireless sensor networks (WSNs) are implemented. Balanced energy-efficient network integrated super heterogeneous (BEENISH) selects four different energy levels of sensor nodes and selects cluster heads (CHs) based on average energy level of the entire network and residual energy levels of each node, whereas improved BEENISH or iBEENISH uses different CHs selection schemes in an very efficient manner by increasing the network lifetime. Here, a mathematical sink mobility model is created which is implemented on BEENISH (Mobile BEENISH or MBEENISH) as well as iBEENISH (Improved Mobile BEENISH or iMBEENISH). Finally, simulation results show that BEENISH, MBEENISH, iBEENISH and iMBEENISH protocols perform better than other contemporary protocols for stability period, lifetime of network and throughput. The selected four energy levels of the nodes are as follows: normal energy level, advanced energy level, super energy level and ultra-super energy level. The initial energy levels for the nodes are least for normal and gradually most for ultra-super level. Here, CH is not fixed but rotated amongst the nodes by the probability, i.e. the ratio of residual energy level of each node in the cluster to the average energy level of the network. Here, the nodes which have higher energy levels are chosen as CHs which means the nodes with lower energy levels are unused and ultra-super, super and advanced node have higher probability to be chosen as CHs. Here, iBEENISH comes to an approach of solving this problem of choosing only the higher energy nodes as CHs, and thus both BEENISH and iBEENISH together claim to improve the network lifetime. If sink mobility is considered, then iBEENISH and BEENISH perform better rather than the non-sink mobility.

3.2 *EERP Using A* Algorithm*

The energy-efficient routing protocol (EERP) is implemented to increase the network lifetime of WSNs using an optimal aggregated cost function along with A* algorithm. A* algorithm helps to select the optimal path from the source node to the destination node taking into consideration of residual energy of nodes, packet reception rate and node buffer state. Here, the sink node is having awareness of criteria of each node

to find the optimal path. Thus, at the initial state, all the nodes are supposed to send the parameters to the sink node. Now while sending data to the sink node later, the parameters of that particular node will be appended to data packet that is to be sent to the sink node. Depending on the parameters that the sink node already gathers, A* algorithm finds the optimal path. As the residual energy level of a node falls below the predefined threshold level energy, that node can no more be considered as a part of the optimal path, thus ensuring a better network lifetime. This node whose energy has fallen below the threshold level need not send its parameters to the base station/sink node anymore. Here, the network load is balanced depending on the threshold value of energy of each node.

3.3 *EE-LEACH*

In this paper, EE-LEACH is implemented. Generally, the nodes which have maximum residual energy if participate in any routing protocol yield energy efficiency and better network lifetime to the network, better packet delivery ration and less energy utilization, better end-to-end delay. This makes only the highest residual energy containing nodes to send their data to the base station. To implement EE-LEACH probability of coverage is calculated from Gaussian distribution function. A list of neighbour node is created with the residual energy data information of those neighbouring nodes using one sorting algorithm. If the sink node is not near to the source node, data ensemble source node cluster is formed in a relatively small area. Taking into account the concentration degree of the source nodes and the residual energy, the optimal cluster head is selected for this purpose. Though energy consumption is reduced here, data confidentiality and integrity are not considered to greater extent.

3.4 *Cluster-Based ACO BAN*

In the field of health care, WSN plays a major role. The routing protocol named cluster-based ACO BAN that is to be discussed here helps to monitor patient data efficiently while sensitively takes care of the network lifetime and energy consumption of the nodes. Here, cluster caves are formed, where nodes send their data to the cluster head and cluster head sends its aggregated data to the net cluster head. Sensitive and important data are passed to the destination effectively and reliably in critical situations. Here, clusters are meant to send only critical information rather than continuous information. At need, the critical information will be shared with the corresponding healthcare department through only the cluster heads. A probabilistic function is created to select the cluster heads in each level. In healthcare field since the monitoring of physical parameters are required, link failure becomes a common occurrence. Due to dead node in the transmission path, packet loss occurs and that misuses the bandwidth and energy. To improve this drawback, minimum numbers of

nodes are required to forward critical data to destination. This also reduces network overhead. A method using CH-ACO algorithm enhances the accuracy of finding such path. This is mainly based on ANT colony algorithm. The algorithm finds its path from source to destination by connecting the neighbours and also updates data in the routing table by collecting data from the nodes regarding their residual energy. To avoid network overhead in this process, clustering concept is used which balances the load of the network, reduces the number of intermediate nodes and finally increases network lifetime.

3.5 PDORP

In wireless sensor network, several routing protocols are used to deal with issues like reliability of the connection/ network, energy consumption, shortest path, delay, network overhead, resource utilization, etc. PDORP, known as directional transmission-based energy-aware routing protocol, is directed towards solving the above-mentioned issues. In case of dynamic source routing (DSR) for small energy density as the node switches from active to sleep, the data packet waits initially which increases the end-to-end delay of packets and waiting time, thus decreasing the efficiency of the network. This drawback increases energy consumption of the network. In PDORP, the goal is to choose the dead nodes and a different path is identified by conserving less energy. This protocol uses the characteristics of efficient gathering of sensor information system, also PEGASIS and DSR to reduce energy consumption, overhead, response time and connectivity issues. Hybridization of genetic algorithm and bacterial foraging optimization helps to find the optimal path. PDORP creates a trust list of the nodes that transmit information. This type of trust list that contains the node parameters and which is updated after every transmission is useful at the time of aggressive transmission and helps to prevent existing route damage. The use of directional transmission decreases the path distance, and thus less energy is consumed. Overall, this protocol provides less BER, less time to packet delivery, less energy utilization, better throughput, ultimately leading to better QoS as well as enhanced lifetime of the network.

3.6 E2HRC

There are several clustering algorithms and routing protocols in the field of wireless sensor network in order to encounter the problem of dying out the nodes due to energy consumption which in turn breaks the communication channels between the nodes. The E2HRC routing protocol effectively balances the energy usage of wireless sensor network, thus decreasing the energy consumption of nodes as well as reduced numbers of control messages. E2HRC is experimented in a ring-domain heterogeneous communication topology. A topology control model is created in a

ring domain where nodes are divided into separate levels as compared to their positions and after that ring domains are also divided. While sending and receiving data, next hop node is selected with the optimal direction angle, thus reducing the energy consumption. Based on the node residual energy and relative node position in the cluster, the network is divided into different heterogeneous clusters and along with this cluster head rotation mechanism helps to balance the node energy consumption. E2HRC routing protocol uses two parameters as optimal direction angle along with energy to balance energy utilization of the network.

3.7 LCP

The energy-aware distributed and dynamic load-balancing cluster-based protocol (LCP) is a modified version of HEED protocol. HEED protocol's clustering part is modified to increase the energy efficiency of the network. The clustering method has two phases known as the setup phase and the steady-state phase. Features of LCP are as follows: The selected cluster heads advertise message within its own cluster range. Selection of cluster heads after every round, i.e. re-clustering, is one of the energy-consuming jobs which is eliminated in this protocol discussed here by setting an initial time interval at the starting stage of every cycle; this helps to delay in re-clustering message that is to be received from the base station. As the nodes do not receive base station messages such sooner, the cluster head keeps rotating the authority among the existing member nodes by selecting the one that contains maximum remaining energy. LCP promises network lifetime increase in a considerable figure in comparison with the existing protocols such as LEACH, HEED, RHEED, etc.

3.8 REM

A variety of routing algorithms are designed in the field of wireless sensor network that promises better performance in terms of delay, packet dropping, intrusion, energy consumption, network lifetime, etc. A major real-life scenario arises in the field of firefighter operation majorly known as firefighter communication where sensors are implemented in the field and data at critical time is to be transmitted to the fire station real time with least delay to save the crisis situation. A very fast dependable and robust communication network is required in this situation. Design of a specific routing protocol meets the specific need to save the crucial situation. A reliable, energy-balancing, multi-group routing protocol (REM) is designed for reliability and energy-balancing communication by selection metric-based CH, cluster rotation and imposing a routing algorithm. Sensor nodes are assigned metric value, and the nodes are chosen as cluster heads according to the highest metric value. The node with highest residual energy and number of connections is assigned to have a higher

Table 1 Analysis of literature survey of the above-mentioned protocols

S. No.	Routing protocol	Domain/class/feature	Advantage	Disadvantage/drawback	Application
1	BEEENISH, MBEENISH, iBEEENISH and iMBEENISH	Heterogeneous wireless sensor network, cluster-based routing protocol, multi-level energy based	Better stability, increased network lifetime, more number of packets/messages transmitted to base station	Four levels of energy level induce more complexity; ultra-super, super and advanced nodes are more tortured	Military, traffic transportation, environmental monitoring, mobile communication, etc.
2	EERP	Multi-hop scheme	Energy-efficient, reduced packet retransmission, load balanced	Minimum hop count to find optimal shortest path needs high link quality	Micro-electro-mechanical systems (MEMS), wireless communications, target tracking, environmental monitoring and battlefield applications
3	EE-LEACH	Effective data ensemble and optimal clustering	Energy-efficient data gathering, reduced energy consumption, increased network lifetime	Lack of data security and integrity	Military, object tracking, habitat monitoring
4	Cluster-based ACO BAN	Wireless body area network, optimal path selection, a meta-heuristic search algorithm combines ANT colony optimization (ACO) using clustering	Better network connectivity, better throughput, robustness	Multiple path selection algorithms are to be run for backup, acknowledgement segment with every packet caused network overhead	Body area network, health monitoring

(continued)

Table 1 (continued)

S. No.	Routing protocol	Domain/class/feature	Advantage	Disadvantage/drawback	Application
5	PDORP	Energy optimization using hybridized algorithm, cache and directional transmission concept	Less buffering, delay is reduced, less energy utilization and better throughput, consistent QoS and more lifetime of network	Sink mobility is not considered, initial creation and updation of trust list at each round puts overhead to the network	Battlefield surveillance, habitat monitoring and underwater monitoring
6	E2HRC	A heterogeneous ring-domain communication topology, clustering algorithm, event-driven cluster head rotation mechanism	Reduced energy consumption, number of control messages, reduced signal attenuation for long-distance communication and decreased packet loss ratio	The packet delivery ratio gradually decreases as number of nodes increases in wireless sensor network	IPv6 routing protocol for low power and lossy networks
7	LCP	Distributed and dynamic clustering protocol, inter-cluster approach, cluster head rotation approach	Energy-efficient, better load balancing, increased network lifetime	Compromised end-to-end delay, packet delivery is done hop by hop	Extreme environment like dense forest, earthquake zone, volcanic area, etc.
8	REM	Cluster-based hierarchical approach, metric-based CH selection and CH rotation	Reliable, less delay, energy balancing, better packet delivery ratio	Delay variation occurs with increased numbers of firefighters	Firefighter communication network (FCN)
9	PECE	Cluster formation, stable data transfer, BCO	Reduced cost of communication, energy-efficient, load balancing, increased network lifetime	When the number of clusters is more than 6, the communication between node and cluster both increase in number	Industry, military, environmental monitoring and medical field

metric value. When the cluster head forwards data to the base station, it also takes account of minimum hop to the base station along with residual energy and number of connections of the nodes. This protocol offers low latency, more reliability, good energy balancing and longer network lifetime.

3.9 PECE

The routing protocol named predictive energy consumption efficiency (PECE) for WSN consists of the steps: cluster formation and stable data transfer. First, cluster formation is done based on forming an energy-saving routing algorithm which takes into account the node degree, residual energy of the nodes and distance of the route, i.e. path between the nodes. Cluster head is selected according to the node degree and node relative distance to provide better coverage and to minimize communication cost within the cluster. These formed clusters determine the number of CHs and each cluster size, and also reduce path cost between member nodes and CH within the same cluster, increasing the overall cluster performance. The stable data transfer stage in PECE data transmission is designed with bee colony optimization (BCO). Here, two different bee agents are used to predict the optimized route according to the energy consumption, neighbour hops and delay in the network. This procedure improves the energy balancing of the network (Table 1).

4 Conclusion

Growing demand in the field of WSN accelerates the research scope in this field. There is hardly any domain where communication is not addressed through WSN. Since sensor nodes contain limited battery life, proper energy balancing and utilization is major concern in designing any routing protocol. Clustering routing protocols show better energy efficiency, reliability, load balancing and network lifetime in comparison with direct communication or diffusion-based routing protocols. The logic behind the protocols is epitomized along with the advantages and areas that could be improvised. We have also mentioned the possible application domains of the protocols discussed.

References

1. Ding, X., Yang, H., & Sun, S. (2014). A design of ginseng planting environment monitoring. *Sensors and Transducers*, 166(3), 80–83.
2. Geetha, S., & Gouthami, S. (2017). Internet of things enabled real time water quality monitoring system. Springer open.
3. Hughes, D., Ueyama, J., Mendiondo, E., Matthys, N., Horr , W., Michiels, S., et al. (2011). A middleware platform to support river monitoring using wireless sensor networks. *Journal of the Brazilian Computer Society*, 2011(17), 85–102.
4. Ge, Y., Wang, S. & Ma, J. (2018). Optimization on TEEN routing protocol in cognitive wireless sensor network. *EURASIP Journal on Wireless Communications and Networking*.
5. Li, M., & Liu, Y. (2009). Underground coal mine monitoring with wireless sensor networks. *ACM Transactions on Sensor Networks*.
6. Hao, J. (2015). Image processing and transmission scheme based on generalized Gaussian mixture with opportunistic networking for wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*.
7. Depari, A., Ferrari, P., Flammini, A., Rinaldi, S., Rizzi, M., & Sisinni, E. (2014). Development and evaluation of a WSN for real-time structural health monitoring and testing. *Euroensors, Procedia Engineering*, 87, 680–683.
8. Li, M., Li, Z., & Vasilakos, V. (2013). A survey on topology control in wireless sensor networks: taxonomy, comparative study, and open issues. *Proceedings of the IEEE*, 101(12), 2538–2557.
9. Sohrabi, K. (2000). Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, 7(5), 16–27.
10. Liu, F.A., Zhang, C.H., & Wu, N. (2014) Adaptive hierarchical routing algorithm for WSN based on energy and distance. *Appl. Res. Comput.*, 31(11), 3434--3437.
11. Kaur, P., & Kad, S. (2017) Energy-efficient routing protocols for wireless sensor network: a review. *International Journal of Scientific & Technolgy Research*, 6(12), ISSN 2277–8616 92.
12. Yi, S. (2007). PEACH: Power-efficient and adaptive clustering hierarchy protocol for wireless sensor networks. *Computer Communications*, 30, 2842–2852.
13. Heinzelman, W. R., Chandrakasan, A., & Balakrishnan, H. (2000) Energy-efficient communication protocol for wireless microsensor networks. In *System sciences, Proceedings of the 33rd Annual Hawaii International Conference on*, 2(4–7), 10, 2000.
14. Lindsey, S., & Raghavendra, C. S. (2002). PEGASIS: Power-efficient gathering in sensor information systems. In *Proceedings IEEE Aerospace Conference* (pp. 3–1125–3–1130). MT, USA: Big Sky.
15. Youni, O., & Sonia, F. (2004). HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *Mobile Comput IEEE Trans.*, 3(4), 366–379.
16. Qing, L., Zhu, Q., & Wang, M. (2006). Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor network. *ELSEVIER Comput. Commun.*, 29, 2230–2237.
17. Elbhiri, B., Saadan, R., El-Fkihi, S., & Aboutajdine, D. (2010). Developed distributed energy-efficient clustering (DDEEC) for heterogeneous wireless sensor networks. In *5th International Symposium IV Communications and Mobile Network (ISVC)* (pp. 1–4).
18. Saini, P., & Sharma, A.K. (2010). E-DDEEC—enhanced distributed energy efficient clustering scheme for heterogeneous WSN. In *Parallel Distributed and Grid Computing (PDGC), 1st International Conference on* (Iss. 28–30., pp. 205–210).
19. Akba, M., Javaid, N., Imran, Md., Amjad, N., Khan, M. I., & Guizani, M. (2016). Sink mobility aware energy-efficient network integrated super heterogeneous protocol for WSNs. *EURASIP Journal on Wireless Communications and Networking*.
20. Ghaffari, A. (2014). An energy efficient routing protocol for wireless sensor networks using a-star algorithm. *Journal of Applied Research and Technology*, 12, 815–822. Yeung, K. Y., et al. (2001, April 1). Validating clustering for gene expression data. *Bioinformatics*, 17, 309–318

21. Yao, J., Lin, C., Xie, X., Wang, A., & Hung, C.-C. (2010). Path planning for virtual human motion using improved A* star algorithm. In: *Information Technology: New Generations (ITNG), Seventh International Conference* (pp. 1154–1158).
22. Arumugam, G. S., & Ponnuchamy, T. (2015) EE-LEACH: Development of energy-efficient LEACH Protocol for data gathering in WSN. *EURASIP Journal on Wireless Communications and Networking*.
23. Srinivas, M. B. (2016). Cluster based energy efficient routing protocol using ant colony optimization and breadth first search. *Procedia Computer Science*, 89, pp. 124–133.
24. Brar, G. S., Rani, S., Chopra, V., Malhotra, R., Song, H., & Ahmed, S. H. (2016). Energy Efficient Direction-Based PDORP Routing Protocol for WSN, Special Section on Green Communications and Networking for 5G Wireless. *Digital Object Identifier*, 4, 3182–3194.
25. Johnson, D., Hu, Y. & Maltz, D. (2007). The dynamic source routing protocol (DSR) for mobile Ad Hoc networks for IPv4. IETF, document RFC 4728.
26. Chen, H., Zhu, Y., & Hu, K. (2009) Cooperative bacterial foraging algorithm for global optimization. In *Proceeding IEEE Chinese Control and Decision Conference*, (vol. 6, pp. 3896–3901). Guilin, China.
27. Kim, D. H., Abraham, A., & Cho, J. H. (2007). A hybrid genetic algorithm and bacterial foraging approach for global optimization. *Information Sciences*, 177(18), 3918–3937.
28. Zhang, W., Li, L., Han, G., & Zhang, L. (2017). E2HRC: An energy-efficient heterogeneous ring clustering routing protocol for wireless sensor networks. *Special Section on Future Networks: Architectures, Protocols, and Applications, Digital Object Identifier*, 5, 1702–1713.
29. Eshaftr, M., Al-Dubai, A. Y., Romdhani, I., & Yassien, M. B. (2015). A new energy efficient cluster based protocol for wireless sensor networks. *Proceedings of the Federated Conference on Computer Science and Information Systems*, 5, 1209–1214.
30. Atiq, M. K., Manzoor, K., Hasan, N. ul., & Kim, H. S. (2014). A reliable energy-balancing multi-group (REM) routing protocol for firefighter communication networks. *EURASIP Journal on Wireless Communications and networking*.
31. Zhang, D., Wang, X., Song, X., Zhang, T., & Zhu, Y. (2015). A new clustering routing method based on PECE for WSN. *EURASIP Journal on Wireless Communications and Networking*.

Role of Fog Computing in IoT-Based Applications



Charul Thareja and N. P. Singh

Abstract Internet of Things (IoT) as the name suggests is an interconnection of our daily usage items like smartphones, cars, laptops, tube lights, electric fans, etc. With the inception of IoT, each and every electronic item in this world would have an IP address. It would transform our way of interaction, making our life more autonomous. The essential requirement for the upcoming of this technology includes real-time response, low latency, fast processing and computation capabilities. These features being difficult to be handled by cloud data centres because of their remote location leads to development of a new technology called fog computing, which is an extension of cloud computing services from cloud data centres towards the edge devices. Fog computing possesses some security and privacy threats which are discussed in this paper. There is a need to resolve these threats so that people use fog-based IoT applications without any reluctance. Some of the solutions proposed are also mentioned in this paper.

Keywords IoT · Fog computing · Cloud computing · Applications · Challenges

1 Introduction

Internet of Things (IoT) is the interconnection of billions of devices that can interact and share resources with the help of embedded software, sensors and other electronic equipment [1]. The IoT possess applications in almost all domains like military for battlefield surveillance [2], health care for patient's health monitoring [3], environment for analysing moisture content in soil [4], etc. It is inferred that IoT is going to make our life autonomous, which can be illustrated from one of the most common applications of IoT, i.e. Smart home application [5], enabling lighting system, room temperature, entertainment system and other electronic devices to be controlled automatically, with the help of sensors.

C. Thareja (✉) · N. P. Singh
Department of ECE, NIT Kurukshetra, Kurukshetra 136119, India
e-mail: tharejacharul19@gmail.com

© Springer Nature Singapore Pte Ltd. 2019
N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_9

The above-mentioned IoT applications collect a huge amount of data, which needs to be processed or analysed. Presently, with 23.14 billion Internet-connected devices, the cloud data centres can handle the amount of data gathered by them, but with the rapid growth of Internet-connected devices, which is estimated to increase up to 500 billion by 2025, the data to be processed will also increase with the approximation of about 500 zettabytes by 2019, which is beyond the handling limit of cloud data centres [2]. So, the need of the hour is to have a scenario in which the data is analysed in real time and only some of the data which require huge computations, or which are necessary for storage, need to be sent towards the cloud DCs.

IoT applications possess different characteristics, like data storage, huge computations and real-time responses. For data storage and huge computational task, cloud computing is a good solution. But for real-time response, it becomes difficult for cloud data centres to handle IoT applications, because cloud DCs are remotely located and leads to poor QoS. So, the need is to bring these cloud services from far-away places to near the user devices, which is done through fog computing.

Thus, fog computing can be defined as an extension of the cloud computing for the real-time, latency-sensitive applications, which consists of a combination of heterogeneous, decentralized devices, communicating and cooperating among themselves to perform data storage and various data computation tasks [6].

The remainder of the survey is organized as follows. Section 2 represents the features of fog computing along with its comparison with cloud computing. The architecture of fog computing has been reviewed in Sect. 3. Role of fog computing in different IoT-based applications has been discussed in Sect. 4. Sections 5 and 6 include the various security and privacy threats of this system and measures to secure it, respectively. Finally, the survey is concluded in Sect. 7.

2 Features of Fog Computing

Many of the features of fog computing are similar to that of the cloud computing but some are different. Both have been described briefly. The similar features include the following:

- **Mobility Support** [7]: Mobility is the ability to move. Through fog computing, IoT devices acquire this ability without being disconnected from the fog nodes. Fog services are ubiquitous in nature. Hence, mobility support offered by cloud data centres is limited as compared to the support provided by fog nodes.
- **Heterogeneity** [8]: The data is collected from Internet-connected devices that can be either mobile IoT devices including wearable devices (fitness tracker, smart cameras, etc.) and mobile smart devices (vehicles, smartwatches, etc.) or fixed IoT devices including sensors and RFID tags deployed on specific areas of product. Collected data is then transferred to the fog nodes (basically routers, gateways and switches) which are deployed in diverse environment like on the roof of buildings, in smartphones, in traffic lights and so on, making the whole architecture

heterogeneous. Same heterogeneity is seen in cloud computing but with the difference that data instead of being collected by fog nodes is collected by cloud data centres which are remotely located.

- **Online analytics and interplay with cloud [9]:** Fog nodes acts as an intermediate between the IoT devices from which the data is being collected and the clouds where the data is to be sent for storage. This feature of fog computing can be explained with the help of an example of smart e-healthcare centre. The data from the sensors deployed on the patient's body is being collected continuously by fog nodes, which is being analysed in real time to detect emergent event. If everything seems to be normal, and then the data is sent to the cloud for storage. Otherwise, some necessary steps are taken to help patients like ringing alarms and sending alert messages to the doctor. Until the doctor arrives, sensors on the patient's body take active actions to give them intensive care. It can be said fog layer is an interface between the device layer and the cloud layer. Similar analysis is performed by the cloud data centres but with some delay and with only two layers (Device layer and cloud layer).

Some of the distinguishing features from the cloud computing are enlisted below:

- **Location Awareness [10]:** The location of the fog nodes can be tracked easily to provide real-time services to the user devices. Based on the knowledge of location of fog nodes, device region can also be identified easily.
- **Geographic Distribution [10]:** The fog nodes that are geographically distributed may be on the roof of the building, on smartphones, on top of the vehicle and at the point of interest so that it is able to provide services to IoT devices, i.e. collect high data stream from IoT devices for processing.
- **Low Latency [9]:** Fog nodes being deployed nearer to the user devices and possessing the necessary capability of storing and computing data are able to provide services with much less time as compared to that taken by the cloud.
- **Large-Scale IoT Application support [8]:** Large-scale IoT application includes large number of sensors, and thus a large number of data for computation and processing to be handled in real time, which is a cumbersome task for cloud computing but can be easily handled by fog nodes which are ubiquitous and has the expertise to manage huge number of IoT devices.
- **Decentralization [8]:** Fog computing is a decentralized network. No central server is present to provide services and resources. Fog nodes arrange itself automatically to provide real-time services to the IoT devices.

Table 1 shows the comparison of different features of cloud and fog.

3 Architecture

The fog computing architecture is illustrated in Fig. 1. It is a three-tier architecture.

Tier 1: The bottom tier known as the device layer consists of the different smart devices including mobile IoT devices, i.e. the devices carried by their users like fitness

Table 1 Comparison of different features of cloud or fog [2]

Features	Cloud	Fog
Latency	High	Low
Mobility	N/A	Yes
Architecture	Centralization	Decentralization
Service access	Through core	At the edge/on handheld device
Location awareness	N/A	Yes
Geographic distribution	N/A	Yes
Scalability	Average	High
Availability	High	High
Number of users or devices	Tens or hundreds of millions	Tens of billions
Content generation	Central location	Anywhere
Content consumption	End devices	Anywhere
Software virtualization	Central corporate facilities	User devices/network equipment

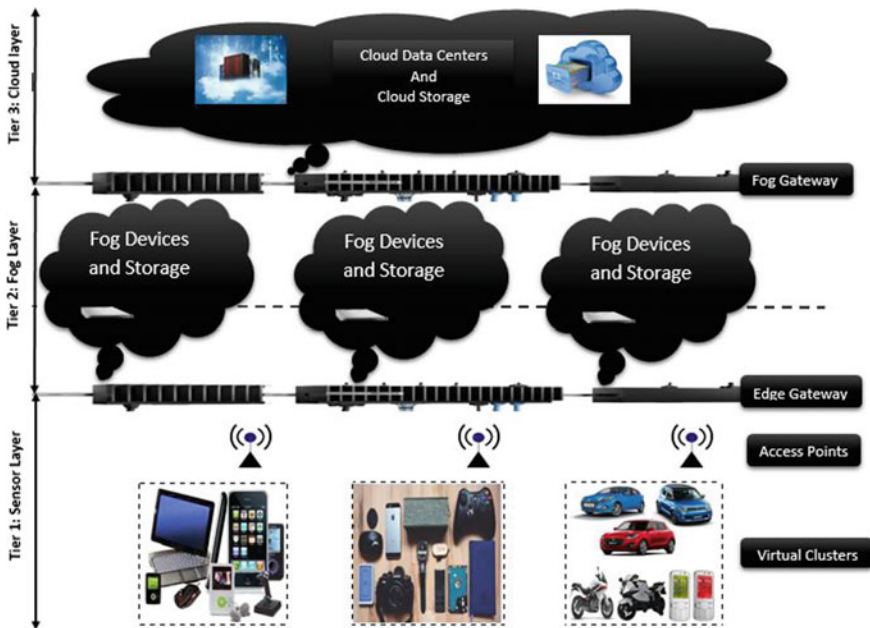


Fig. 1 Fog computing architecture

trackers, bracelets and smart devices like smartphones, vehicles, smartwatches, etc. and the fixed IoT devices including RFID and sensor nodes which are attached on the items to perform some specified task. These heterogeneous devices transmit data to the upper tier.

Tier 2: The second tier known as the fog computing layer consists of intelligent devices called fog nodes which are routers, gateways, etc. that possess the ability of data storage, computation, routing and packet forwarding.

Tier 3: The uppermost layer is known as the cloud computing layer, which consists of data centres and big servers. The data which requires large historical data for processing or huge computational tasks is sent to the cloud layer.

Different smart devices also known as the terminal nodes join together to form virtual clusters (VCs). Different VCs together form an edge virtual private network (EVPN). The data from the VCs is sent to the fog instances (FIs) via edge gateways for computing, storage, etc. The fog layer is divided into two parts, one is fog abstraction layer, which manages the fog resources, enables virtualization and provides user's privacy. Another part is fog orchestration layer, consisting of a software agent called foglet, which checks the state of device. Within the fog instances, data is processed and analysed, whether it requires real-time service, temporary storage or permanent storage. For permanent storage or long-term storage, data is sent to the data centres or cloud layer but for short-term evaluation, data is processed in fog layer only. Thus, the fog computing architecture allows the efficient utilization of cloud Layer [11].

4 Applications

Fog nodes play four important roles, namely, real-time service for fog, transient storage for fog, data dissemination for fog and decentralized computation for fog. Based on these roles, some of the important applications have been discussed below.

4.1 Real-Time Services for Fog

Cloud data centres being far away from the IoT devices takes a lot of time in analysing and computing the results of data, because first the data is sent to the cloud, then in DCs it takes some processing time and after that the analysed result comes back to the device. So, three different times are being added here: time taken by data to reach the cloud (T_c), processing time (T_p) and time taken by the analysed data to reach back to the device (T_r) [12]. So, the expression for total time (T_t) can be written as

$$T_t = T_c + T_p + T_r$$

But in case of fog computing, fog nodes being nearer to the fog devices, the data after the analyses reaches within microseconds to the required destination. Though all the three times which are added in cloud computing scenario are added in fog computing also, the time taken by data to reach to fog nodes is much less as compared to that of the cloud. Thus, for real-time applications like in Battlefield Surveillance, it is really important for us to get the information regarding the entry of an intruder into our land, so that some necessary steps can be taken to avoid any kind of uncertainty in our country. But, if we depend upon cloud data centres for this information, then the security personnel will get the data when the intruder has already entered the land. So, for latency-sensitive applications, fog nodes act as a private cloud for the IoT devices.

Some applications of fog nodes which offer real-time control and fast decision-making capability are smart traffic light, decentralized vehicular navigation, home energy management, e-healthcare services, etc.

- **Smart Traffic Light [13]:** In this application, the fog nodes are deployed on the traffic light pole and takes necessary actions according to the data collected by the sensors, which are installed on the road, on traffic signal pole and many more places for covering the whole view of the Carrefour. The light sensors when detecting the flashy lights of ambulance and police vans send the data to fog nodes, which then take urgent steps to let these emergency vehicles go first. Some sensors send the details of the speed of pedestrians and the bicyclers crossing the road towards the fog nodes, and accordingly the traffic signals are arranged by giving them enough time to cross the roads. Not only this but fog nodes also try to avoid collisions by sending warning messages to the vehicles which are approaching with high speed. The sensors deployed on roads continuously gather data regarding the speed of vehicles or send it to fog nodes for further analyses. Thus, this application, with the help of fog computing, is going to reduce the number of accidents.
- **Healthcare and Activity Tracking [14]:** The scenario can be understood in three layers: in the first layer, all the sensors are included which are deployed on the body of the patient for collecting data from all the ICUs. In the second layer, fog node, i.e. Private cloud of hospital, is present to deliver fast responses after analysing the data provided by sensors like sending messages to doctor or starting alarms during emergency situation and giving necessary instructions to the sensors to take preventive actions until the doctor arrives. The third layer consists of a cloud, which is just for storing all the details of the patient. If there is a need to check medical history of patients, then this layer comes in use.

4.2 *Transient Storage for Fog*

When the data is to be stored temporarily, i.e. for about 1–2 h, or if the data is to be accessed frequently then instead of storing it in clouds (because storage space and bandwidth would be wasted), it can be stored in fog nodes and when the necessary

updates have been done on the data then it can be either discarded or sent to the clouds for permanent storage. When some files which need to be accessed continuously for real-time application are stored in cloud, then delay is added to the response.

This role of fog node is useful in various applications including content caching, shopping cart management, software and credential updating, etc. Some of them are discussed below.

- **Content Caching [8]:** The data which is accessed by the user for the first time comes from the cloud and gets stored into the fog node. The second time when some other user wants to use the same information then there is no need to go to the tier 3 again as the data has already been cached from cloud to the fog node. Hence, the second user gets faster response as compared to the first one. Its working is same as that of the cache memory in computers. This application can be used in interstate buses by employing a fog node on the top of the bus providing free Wi-Fi as a social service, and then all the passengers can access the data using that fog node.
- **Shopping Cart Management [8]:** Traditionally, while e-shopping, the data of e-cart needs to get updated in the cloud, which used to add delay to the services of the customer. But with the advent of fog computing, the cart information gets cached into the fog nodes, where the necessary updates are done by the customer after that when the customer logs out his/her account, then the data is sent back to the cloud for storage. This procedure reduces the amount of delay in services and even increases customer's satisfaction.

4.3 Data Dissemination for Fog

Being the middle tier in the fog computing architecture, fog nodes act like an interface between the IoT devices and the cloud data centres. The fog nodes either collect data from IoT devices and send it to cloud or collect data from cloud data centres and send it to IoT devices. Fog node acts as a router, helps in packet forwarding, can work as a data aggregator, and can also perform simple processing for choosing the proper audiences to whom the information should be sent.

The applications including data dissemination for fog includes energy consumption collection, local content distribution and fog-based malware defence. Some of them are discussed below:

- **Local Content Distribution [15]:** Local fog nodes provide useful information to the users passing by, e.g. users are given updates about the traffic, hotels, restaurants, gas stations, while they are travelling and if simple processing is done, then the messages of nearby hotels, restaurants, etc. would be sent to a person who is sensualist and about amusement parks to a person who is adventurous, etc. This application can also be used in parking navigation service [16], by suggesting the drivers the right place to park, with the help of various sensors and cameras being deployed in the parking area.

- **Fog-based Malware Defence [8]:** If malware defence hardware and software are installed in each of the devices, then it would waste a lot of storage space and would require timely processing (more battery backup). So to avoid it, if the malware defence system is installed on the fog nodes, then it would act as shared resource for all the IoT devices to detect the infected files on compromised devices, and then clean them.

4.4 Decentralized Computation for Fog

Fog nodes provide decentralized computation services, which aid both the ends, i.e. user devices and cloud centres from heavy computations. Like in mobile phones, all the computations and processing cannot be done because it will consume battery, so to avoid it, computational task is transferred to the fog nodes, which sends analysed data back to device, after processing. Likewise, for every small computation, earlier the mobile devices used to offload data to cloud DCs, adding latency to the process, wasting the bandwidth and even wasting the computational resources, which is now performed by fog nodes.

The applications supporting the above feature are computation offloading, aided computation and big data analytics. Some of them are discussed below.

- **Computation Offloading [17]:** Though clouds possess huge computational capability, it consumes a lot of energy and adds latency in providing services to the devices. Fog computing brings these computational services towards the user devices. The IoT devices can fully exploit the computing resources of the fog nodes even for simple processing and it also reduces the computational burden on clouds.
- **Big Data Analytics [18]:** As the number of IoT devices is increasing, the amount of data collected by these devices is also increasing. Analysing huge amount of data at cloud DCs would be difficult; instead, the task can be distributed among different fog nodes. Like, when finding a missing person, instead of analysing all video recordings collected at cloud DCs, the task is divided into each fog node to check the records. Thus, performing local analytics at the fog nodes makes process fast and reduces delay.

5 Threats of Fog Computing

The threats which have restricted the growth of cloud computing and fog computing are security and privacy threat. These are discussed below.

5.1 Security Threat

Fog computing is more secured than cloud computing because of two main reasons. First, the data is transferred locally between the data sources and the fog nodes, and the dependency on Internet connection is reduced. Since the data is being stored, analysed and transferred locally, it becomes difficult for hackers to access user's data. Second, the information exchange between the cloud and the smart devices does not take place in real time which does not let the tracker find critical information of the user.

Fog computing being more secure than cloud computing cannot be said to be fully secured; as the user's data reaches to the employees employed at fog nodes, they may misuse their data and this even leads to privacy leakage. Some of the security threats are as follows:

- **Forgery and Spam:** Some attackers may make their fake profiles and may even add fake information about them on it, to mislead other users. This also causes wastage of storage, bandwidth and energy. Spam refers to the redundant or unwanted information, which unnecessarily wastes the storage resources.
- **Tampering:** Some attackers try to interfere with the data sent by the user. They can either alter the data or drop the data; the purpose is that the real data should not reach the destination. Being wireless communication, it becomes difficult to detect whether this drop occurred due to transmission failure or by some intruder.
- **Jamming:** Some attackers generate a huge amount of redundant or fake messages, so that no other user can utilize the computation and routing services of fog. The purpose is to jam the communication channels. It can also be said as Denial-of-Service, when some legitimate user is not able to use the services of fog because it is being flooded with fraudulent data. It is common in fog because of its limited resources.
- **Eavesdropping:** Some attackers continuously listen to the communication channel, to get the important details. This type of attack is common when data encryption techniques are not used.
- **Collusion:** Two or more parties can combine together to fraud a legal user. The two parties combined can be IoT devices, fog and IoT device or cloud and fog, etc. This is going to double the attack capability.
- **Man-in-the-Middle:** There may be an attacker sitting in between the two parties, listening to their conversations, who is secretly modifying the data being exchanged between them and the two parties do not even get to know it.
- **Impersonation:** There can be a situation in which an illegal user can act as a legitimate user to utilize the services provided by the fog nodes or it may happen a fake fog is providing services to the legal users and maliciously utilizing the information provided by them.

5.2 *Privacy Threat*

It is also an important issue as the user's data is being processed, shared and collected, and no user wants his/her data to be leaked. There are four kinds of privacy threats which includes identity, data, location and usage privacy.

- **Identity Privacy:** User's identity includes his/her name, address, Aadhar card number and some more personal details which are easily disclosed during checking the authenticity of a person while they are availing for the services of the fog.
- **Data Privacy:** When users utilize the fog services, some of the information about user like its occupation, health status, preferences are exposed to some third party which maintains the user's data at the fog node.
- **Usage Privacy:** Usage pattern tells when the users utilize the fog node services, which gives the information about the living pattern of the user including, when they sleep, when they wake, at what time they go to office and many more details, which are really dangerous if it gets into wrong hands.
- **Location Privacy:** Location privacy is something which every user has to sacrifice in order to enjoy different online applications. If a user utilizes some fog node services, his/her location can easily be accessed by finding the fog node to which it is connected and an attacker can even detect the trajectory of a user.

6 **Fog Security**

The threats mentioned above leads to the unwillingness of users to utilize fog-based IoT applications. So, this generates a need to secure fog computing. In this section, the security and privacy challenges have been further discussed and the measure which should be taken to avoid these challenges is also discussed. It is divided into four parts similar to the applications studied above.

6.1 *Challenges and Solutions on Real-Time Services*

Fog computing provides various fog-assisted IoT real-time applications, which due to security and privacy threats has not been utilized fully. Some of the challenges faced in these applications with the solutions to them are enlisted in Table 2.

6.2 *Challenges and Solutions on Transient Storage*

Transient storage capability provided by fog allows users to maintain their data provisionally. The users will be using these services only if they are sure that their

Table 2 Security challenges and solutions in real-time services application of fog computing

Role	Security challenges	Security solutions
Real-time services	Identity authentication	Identity authentication [19] Cooperative authentication [20] Anonymous authentication [21]
	Access control	Role-based access control policy [22] Attribute-based access control policy [23]

Table 3 Security challenges and solutions in transient storage role of fog computing

Role	Security challenges	Security solutions
Transient storage	Sensitive data Identification and protection	Symmetric encryption [24] Asymmetric encryption [25]
	Secure data sharing	Proxy re-encryption [26] Attribute-based encryption [27] Key-aggregate encryption [28]

Table 4 Security challenges and solutions in data dissemination role of fog computing

Role	Security challenges	Security solutions
Data dissemination	Secure data search	Symmetric searchable encryption [29] Asymmetric searchable encryption [30]
	Secure content distribution	Secure service discovery [31] Broadcast encryption [32] Key management mechanism [33] Anonymous broadcast encryption [34]

data is safe, and there would be no privacy leakage. The challenges faced in transient storage and the solutions which can be used to provide security are enlisted in Table 3.

6.3 Challenges and Solutions of Data Dissemination

Fog nodes act as an intermediate between the IoT devices and the clouds. So, it requires the information which is being passed between the two ends that should be accurate. The challenges which are faced and the measures which can be taken for efficient data transmission are listed in Table 4.

6.4 Challenges and Solutions of Decentralized Computation

Fog nodes instead of storage capabilities also possess computational capabilities. These computations can be hacked by an attacker, and he/she may be controlling

Table 5 Security challenges and solutions in decentralized computation role of fog computing

Role	Security challenges	Security solutions
Decentralized computation	Verifiable computation	Privately verifiable computation [35] Publicly verifiable computation [36]
	Secure-aided computation	Server-aided verification [37] Server-aided encryption [38] Server-aided key exchange [39]

them and misguiding user. Not only this but there are some more problems which can be faced by user and various solutions to resolve these challenges that are listed in Table 5.

7 Conclusion

Fog computing is not a replacement of cloud computing, but it complements it. Both the technologies when used together form a new breed of technology that serves various IoT applications which may deal with computing, or storing temporary data, or acting as an interface between two layers, or serving real-time applications. This new technology in spite of making our life autonomous is surrounded by some security and privacy threats which need to be resolved to enjoy fog computing. Some of the techniques to resolve the challenges have been mentioned, but this is an open research issue and needs more inputs.

References

- Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645–1660.
- Farooq, M. J., & Zhu, Q. (2017). Secure and reconfigurable network design for critical information dissemination in the Internet of battlefield things (IoBT). In *2017 15th Int. Symp. Model. Optim. Mobile, Ad Hoc, Wirel. Networks, WiOpt 2017* (vol. 17, no. 4, pp. 2618–2632).
- Verma, P., & Sood, S. K. (2018). Fog assisted-IoT enabled patient health monitoring in smart homes fog assisted-IoT enabled patient health monitoring in smart homes (vol. 5, no. February, pp. 1789–1796).
- Tarushi Wasson, P. K., Choudhury, T., & Sharma, S. (2017) Integration of RFID and sensor in agriculture using IOT—IEEE Conference Publication. In *Smart Technol. Smart Nation (SmartTechCon), International Conference* (pp. 217–222).
- Lee, Y. T., Hsiao, W. H., Lin, Y. S., & Chou, S. C. T. (2017). Privacy-preserving data analytics in cloud-based smart home with community hierarchy. *IEEE Transactions on Consumer Electronics*, 63(2), 200–207.

6. Vaquero, L. M., & Rodero-Merino, L. (2014). Finding your way in the fog: Towards a comprehensive definition of fog computing. *ACM SIGCOMM Computer Communication Review*, 44(5), 27–32.
7. Munir, A., Kansakar, P., & Khan, S. U. (2017). IFCIoT: Integrated Fog Cloud IoT: A novel architectural paradigm for the future Internet of Things. *IEEE Consumer Electronics Magazine*, 6(3), 74–82.
8. Ni, J., Zhang, K., Lin, X., & Shen, X. (2017). Securing fog computing for internet of things applications: Challenges and solutions. *IEEE Commun. Surv. Tutorials*, (no. c, pp. 1–1).
9. Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing—MCC'12*, (p. 13).
10. Shropshire, J. (2014). Extending the cloud with fog : Security challenges & opportunities. *Americas Conference on Information Systems* pp. 1–10.
11. Sarkar, S., Chatterjee, S., & Misra, S. (2015). Assessment of the suitability of fog computing in the context of internet of things. *IEEE Transactions on Cloud Computing*, 7161(c), pp. 1–1.
12. Liu, L., Chang, Z., Guo, X., Mao, S., & Ristaniemi, T. (2017). Multi-objective optimization for computation offloading in fog computing. *IEEE Internet of Things Journal*, 4662(c), 1–12.
13. Liu, J., et al. (2018). Secure intelligent traffic light control using fog computing. *Future Generation Computer Systems*, 78, 817–824.
14. Akrivopoulos, O., Chatzigiannakis, I., Tselios, C., & Antoniou, A. (2017). On the deployment of healthcare applications over fog computing infrastructure. In *2017 IEEE 41st Annual Computer Software and Applications Conference*, pp. 288–293.
15. Oteafy, S. M. A., & Hassanein, H. S. (2018). IoT in the fog: A roadmap for data-centric IoT development. *IEEE Communications Magazine*, 56(3), 157–163.
16. Lu, R., Lin, X., Zhu, H., & Shen, X. (2010). An intelligent secure and privacy-preserving parking scheme through vehicular communications. *IEEE Transactions on Vehicular Technology*, 59(6), 2772–2785.
17. Wang, C., Liang, C., Yu, F. R., Chen, Q., & Tang, L. (2017). Computation offloading and resource allocation in wireless cellular networks with mobile edge computing. *IEEE Transactions on Wireless Communications*, 16(8), 4924–4938.
18. Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet Things Journal*, 3(5), 637–646.
19. Chandrasekhar, S., & Singhal, M. (2015). Efficient and scalable query authentication for cloud-based storage systems with multiple data sources. *IEEE Transactions on Services Computing*, 1374(c), 1–1.
20. Lin, X., & Li, X. (2013). Achieving efficient cooperative message authentication in vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 62(7), 3339–3348.
21. Lu, R., Lin, X., Luan, T. H., Liang, X., Member, S., & Shen, X. S. (2012). Pseudonym changing at social spot: An effective strategy for location privacy in VANETs. *IEEE Transactions on Vehicular Technology*, 61(1), 1–11.
22. Salonikias, S., Mavridis, I., & Gritzalis, D. (2016). *Critical information infrastructures security* (vol. 9578, pp. 15–26).
23. Lewko, A., & Waters, B. (2011). Decentralizing attribute-based encryption. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (vol. 6632 LNCS, pp. 568–588).
24. Agrawal, M., & Mishra, P. (2012). A comparative survey on symmetric key encryption techniques. *International Journal of Computer Science and Engineering*, 4(5), 877–882.
25. Tarakdjian, G. (1993) *A Survey* (pp. 74–81).
26. Ateniese, G., Fu, K., Green, M., & Hohenberger, S. (2006). Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security*, 9(1), 1–30.
27. Alrawais, A., Alhothaily, A., Hu, C., Xing, X., & Cheng, X. (2017). An attribute-based encryption scheme to secure fog communications. *IEEE Access*, 5, 9131–9138.

28. Chu, C., Chow, S. S. M., Tzeng, W., Zhou, J., Deng, R. H., & Member, S. (2014). Supplementary material for key-aggregate cryptosystem for scalable data sharing in cloud storage. *IEEE Transactions on Parallel and Distributed Systems*, 25(2), 1–4.
29. Rizomiliotis, P., & Gritzalis, S. (2015). ORAM based forward privacy preserving dynamic searchable symmetric encryption schemes. In *Proceedings of the 2015 ACM Workshop on Cloud Computing Security Workshop—CCSW' 15* (pp. 65–76).
30. Hwang, Y. H., & Lee, P. J. (2007). Public key encryption with conjunctive keyword search and its extension to multi-user system. In *Proceeding of Pairing* (pp. 2–22).
31. Wu, D. J., Taly, A., Shankar, A., & Boneh, D. (2016). Privacy, discovery, and authentication for the internet of things. In *Proceedings of ESORICS* (pp. 301–319).
32. Park, J. H., Kim, H. J., Sung, M. H., & Lee, D. H. (2008). Public key broadcast encryption schemes with shorter transmissions. *IEEE Transactions on Broadcasting*, 54(3), 401–411.
33. Boneh, D., Gentry, C., & Waters, B. (2005). Collusion resistant broadcast encryption with short ciphertexts and private keys. In *Proceedings of CRYPTO* (pp. 258–275).
34. Libert, B., Paterson, K. G., & Quaglia, E. A. (2012). Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model. In *Proceedings of PKC* (pp. 206–224).
35. Gordon, S. D., Katz, J., Liu, F.-H., Shi, E., & Zhou, H.-S. (2015). Multi-client verifiable computation with stronger security guarantees. In *Proceedings of TCC* (pp. 144–168).
36. Elkhyaoui, K., Önen, M., Azraoui, M., & Molva, R. (2016). Efficient techniques for publicly Verifiable delegation of computation. In *Proceedings of ACM CCS* (pp. 19–128).
37. Wang, T., Zeng, J., Bhuiyan, M. Z. A., Tian, H., Cai, Y., Chen, Y., et al. (2017). Trajectory privacy preservation based on a fog structure in cloud location services. *IEEE Access*, 5, 7692–7701.
38. Bellare, M., Keelveedhi, S., & Ristenpart, T. (2014). DupLESS: Server-aided encryption for deduplicated storage. In *Proceedings of Usenix Security* (pp. 179–194).
39. Kamara, S., Mohassel, P., & Riva, B. (2012). Salus: A system for server aided secure function evaluation. In *Proceedings of ACM CCS* (pp. 797–808).

Single Horizontal Camera-Based Object Tracking Quadcopter Using StaGaus Algorithm



Lakshmi Shrinivasan and N. R. Prasad

Abstract This research is a solution to the problem of a hardware platform of object tracking drones. This platform can be used to build further advancements such as selfie drones, follow-me drones for adventure sports and robot pets. StaGaus algorithm has been derived using first principles and is compared against standard algorithms like SRDCF. The algorithm is tested on, an on-board two Android phones for real-time telemetry. This paper demonstrates that StaGaus works even on memory- and performance-constrained devices. In order to standardise the development of object tracking drones' algorithms, we have built a customised ROS-based Gazebo simulator from scratch. This simulator is capable of simulating multiple robots of multiple types. This uses actual physics Open Dynamics Engine which has been compared against the existing simulators. Finally, as a by-product, this design proved that the cost of the proposed physical quadcopter is low.

Keywords Drone · Image processing · Android · Machine learning · Simulator using ROS and Gazebo

1 Introduction

A quadrotor is a rotorcraft propelled by four motors typically using a separate flight controller. One of the non-trivial research problems of machine learning-based image processing that is being tackled with our novelty is the development of object tracking algorithms for an on-board Android controller using a single camera. A single camera is only 2D information unlike the 3D information that is provided by stereo-vision, RADAR and LIDAR. We demonstrate that the accuracy of this 2D information is sufficient enough to track the object without the need of expensive 3D sensors.

Another problem is the standardisation of results of object tracking drones' algorithms, because algorithms like StaGaus employ both image processing statistics and statistics of the controller algorithms. It is essential to simulate both controllers

L. Shrinivasan (✉) · N. R. Prasad
Department of ECE, Ramaiah Institute of Technology, Bengaluru 560054, India
e-mail: lakshmi.s@msrit.edu

© Springer Nature Singapore Pte Ltd. 2019
N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_10

113

with the physics of the quadrotor while maintaining a GUI for image processing. Instead of relying on weather conditions of the locality and relying on hardware which may have to be re-built if they crash, the simulator we have built from scratch using robotics operating system and Gazebo provides a single simulation interface for both controller design and image processing. This simulator can run multiple vehicles of multiple types, and we have made this run on a single Lenovo Y50-70 laptop having Ubuntu Operating System.

2 Problem Definition and Contributions

For both physical and software-based validations (using simulation) of single camera-based object tracking quadcopter, we have come up with an online training StaGaus algorithm and a custom simulator which runs in Gazebo.

Novelty is based on three aspects:

1. The proposed statistical Gaussian algorithm that has high tracking efficiency with a single camera (unlike the additional GPS sensor, LIDAR, RADAR and stereo-vision laden drones).
2. Performance of the algorithm is based on the performance-constrained and battery-optimised devices (as a by-product, this is useful for cost-effective drones).
3. The designed customised Robotics Operating System-based Gazebo simulator that uses actual physics engine to simulate multiple vehicles of multiple types.

3 Related Work

There has been a lot of research work done on object tracking algorithms both in spatial and in frequency domains. But, for a single camera mounted on a quadrotor, the additional problem was with the dynamic backgrounds. In literature, a lot of generic algorithms are available for object tracking using Android phone. Alper et al. describe direct object detection, point-based matching and feature-based and boosting-based machine learning tracking algorithms (like features fed to AdaBoost) [1]. Arnold et al. describe object tracking in videos (mainly with discriminative training) [2]. Hanxuan et al. describe feature matching and online training extensively for algorithms (with performance optimisations like Monte Carlo sampling) [3]. Rui et al. describe an algorithm that uses both spatial- and frequency-domain-based tracking systems [4]. But we found that the next advancement was to avoid position error that would occur in long-term tracking. Also, there were very less or almost no literature for an Android-controlled quadrotor design. Hence, the authors proposed new design to build and validate system, in a standard robotic simulator with accurate physics

and visuals as StaGaus algorithm which uses control along with image processing algorithms. Since many of these surveys use OpenCV, OpenCV is used for simulation of the system.

Literature survey was conducted to test our drone in the simulator that researchers would use and that would simulate both physics and image processing in real time. Aaron et al. describe the popular simulators of the time with the considerations of ease of programming, popularity and the physics of the software simulations [5]. Pablo et al. describe a large number of simulators for robots on the bases of price, ease of development and usage, architectures, distributed computing and the accuracy of the simulations [6]. From these researches, we concluded that ROS and Gazebo provide stable simulations while still letting image processing algorithm work along with LIDAR mounted on quadrotor. While we found simulators built on different software like Unreal Engine, we did not find a stable object tracking-specific simulator on ROS and Gazebo [7]. Also, researches have indicated that researchers and users are very likely to use ROS- and Gazebo-based simulators [8]. We decided to build multi-type, multi-robot simulator for object tracking.

4 Proposed Methodology for Physical Validation

Figure 1 describes the overall architecture. Ultrasonics sensor is used for the obstacle avoidance of obstacles. (HCSR04 has a range of 4 metres with a time limit of 60 milliseconds.) Ardupilot APM 2.6 flight controller, 2200 mAh lithium polymer battery, four 850 kV brushless motors, four electronic speed controller cum battery elimination circuit modules, propellers and two Android phones are used.

Ardupilot APM 2.6 supports USB communication, pulse width modulation of 50 Hz, micro-aerial link commands via telemetry port. It has provisions for external sensors. It has a lot of internal sensors like accelerometer, gyroscope, compass, internal barometer and 16 MB internal memory. It supports up to 18 V and 90 A ESCs [9].

4.1 Proposed OpenCV-Based Image Processing Algorithm

The authors have used Open Computer Vision library because Java is well supported in Android through OpenCV Manager. We use a user interface in Android in which the user just draws a rectangular bounding box by drawing diagonally (user touches on one corner of rectangle, triggers Action Down event, swipes till the diagonally opposite corner and releases thereby triggering Action Up event of Motion Event of Android). For this, GrabCut's filtering is employed for initial image to segregate foreground and background [10]. The posterior Bayesian smoothing provides filtering around the local maximum [11]. If the image does not look fine, the user can just repeat initialisation.

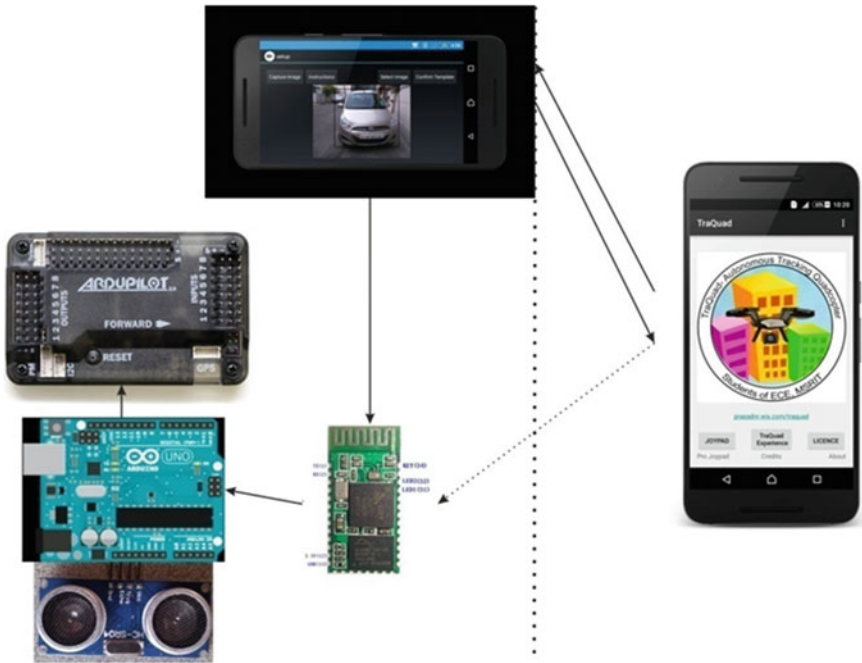


Fig. 1 System architecture. Architecture of the proposed physical quadrotor system

As the processors' performance is generally tuned to work well for hexagonal approximation for colour spaces, the standard hexagonal approximation has been used [12]. The centroid of the blob is calculated on the basis of confidence of the features of the image. The centroid and size of the blob constitute the input vector for the online training algorithm StaGaus. The blob would be derived out of the filtered image of the object of interest derived out of rectangular area enclosed by the rectangular bounding box. During live tracking, feature matching is used against this reference blob. The main advantage of this method is that the users need not enter parameters of the online training algorithm StaGaus. The parameters would be optimised automatically over time. ORB is used for feature matching as it provides the confidence of the features and also the feature matching. It is real time and is suitable for Android phones [13].

A certain number of features has to be detected within the object (so that the centroid would not shift). If the occlusion is higher and the number of features detects gets lesser than the Hamming matcher threshold, then the image would be discarded. For tracking, the position of centroid is used for X- and Z-axes (represented by yaw and throttle, respectively). The root mean square of the distances of the features from the centroid would determine the size of the blob. This size of the blob is used for the third dimension Y-axis (represented by pitch).

4.2 Proposed Linear Control Algorithm

Velocity is chosen as a linear function of the position of the centroid. (Experimental setup proved that choosing acceleration as a linear function of the position of the centroid would lead to transients and object tracking percentage drops drastically.) Width and height represent the width and height of the image respectively. ω_{yaw} represents the angular velocity of the yaw. $v_{throttle}$ represents the velocity of the throttle. v_{pitch} represents the horizontal and forward velocity.

$$\forall \text{ mod } (t) = \begin{cases} -t & |x| < 0 \\ t & |x| \geq 0 \end{cases} \quad \text{step}(t) = \begin{cases} 0, & t < 0 \\ t, & |t| \geq 0 \end{cases}$$

x and z are the horizontal and vertical distances of the left-top corner of the image.

$$v_{yaw} = 1 - \left[\text{step}\left(x - \frac{\text{width}}{4}\right) - \text{step}\left(x - \frac{3 \times \text{width}}{4}\right) \right] \times \left[1 - \left(\frac{4}{\text{width}}\right) \times \text{mod}\left(x - \frac{\text{width}}{2}\right) \right] \quad (1)$$

$$v_{throttle} = 1 - \left[\text{step}\left(z - \frac{\text{height}}{4}\right) - \text{step}\left(z - \frac{3 \times \text{height}}{4}\right) \right] \times \left[1 - \left(\frac{4}{\text{height}}\right) \times \text{mod}\left(z - \frac{\text{height}}{2}\right) \right] \quad (2)$$

Equations (1) and (2) represent the yaw and throttle. For pitch control, the lesser σ (Standard Deviation) means higher speeds. So,

$$df(x) = \frac{-f(x)xdx}{\sigma^2} \rightarrow f(x) = ce^{\frac{-f(x)xdx}{\sigma^2}} \quad (3)$$

$$\mu_{\text{centroid}}(x, z) = \frac{\sum \eta_{\text{individual}} \times \text{centroid}_{\text{individual}}(x, z)}{\sum \eta_{\text{individual}}} \quad (4)$$

$$\forall \text{distance}_{\text{image}} = \left| \left(\frac{\text{width}}{2}, \frac{\text{height}}{2} \right) - \mu_{\text{centroid}}(x, z) \right| \quad (5)$$

The distance determined in Eq. (5) is used for the determination of statistics of the object being tracked.

$$\sigma_{\text{centroid}} = \sqrt{\frac{\sum (\text{distance}_{\text{image}})^2}{n}} \quad (6)$$

Using Eqs. (3), (5) and (6), the equation deduced is used for pitch.

$$v_{\text{forward}} = \frac{v_{\text{forwardMax}}}{\sigma_{\text{centroid}} \sqrt{(2\pi)}} e^{-\frac{1}{2} \left(\frac{\text{distance}_{\text{image}}}{\sigma_{\text{centroid}}} \right)^2} \quad (7)$$

Equation (7) represents the third dimension derived out of two dimensions using statistical analyses. Equations (1) and (2) account for the changing bounding box size. On-field test demonstrated that full linearization of (1) and (2) results in yaw oscillations. So, we have restricted the linearization to 50% exactly.

In StaGaus algorithm, if $\text{distance}_{\text{image}} > \sigma_{\text{centroid}}$, the existing curve of StaGaus is used. But, the $\text{distance}_{\text{image}}$ would be used for the updation of the curve. Else, curve is updated and centroid is used as per Eqs. (1), (2) and (7). This StaGaus algorithm avoids position error that usually occurs in long-term learning. By using the template and re-centering using the control algorithms, the target is not lost until there are occlusions.

4.3 Proposed User Interface

The user interface determined the most of the development of autonomous algorithm. Applications typically use a Ground Control Station (GPS) and are usually partially autonomous (like the usage of GPS waypoints). But, with a single-camera on-board, the quadrotor has an interface which just streams videos back along with the detected object marked with a circle. The only control the user has is the emergency button for an emergency joypad control. Pro Joypad and simple joypad modes are just for initialisation. However, they can also be used for the manual testing of object tracking algorithm. In Pro Joypad mode and simple joypad mode, several manoeuvres can be tried with manual input with detected object while bypassing autonomous control.

As mentioned in the left side of Fig. 2, Pro Joypad mode would stream the video while still maintaining a four-channel control. The four channels values: roll, pitch, yaw and throttle values, are sent. The value of each channel would vary 1000–2000 which corresponds to the RC input values of the flight controller. This mode made StaGaus compute the object in real time as the quadrotor also had to accept input from Pro Joypad mode. Right side of Fig. 2 displays the simple joypad mode. This mode is similar to Pro Joypad. But the controls are simple and are not meant for manoeuvres.



Fig. 2 The Pro Joypad mode in the left and simple Joypad mode in the right

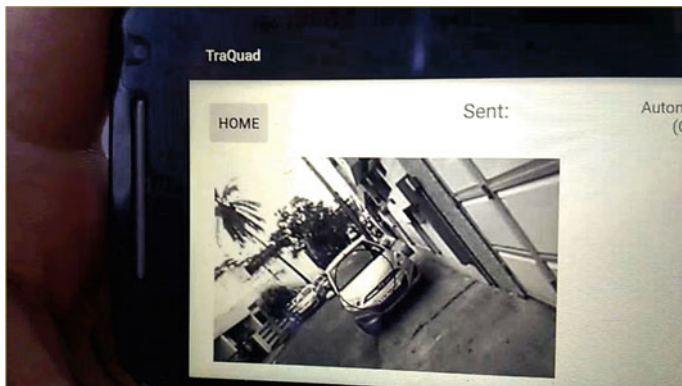


Fig. 3 Object tracking mode indication in the Android application

User swipes across the screen once. User can select an image already stored on phone or use camera (managed through Android intent) to capture a new image and that can be loaded in the setup screen. The object is filtered using the bounding box as the limit and background is removed. ORB feature matching starts.

Figure 3 demonstrates the tracking scenario. The feature matching is done using ORB feature matching using brute-force Hamming matcher. Note that the algorithm is very robust against rotations (both in-plane and out-of-plane). So, a circular pattern reinforced the StaGaus algorithm regarding rotational invariance which boosted the tracking percentage. GrabCut consumes about 0.6 s. This is sufficiently enough as the features would be stored in global class of Android.

5 Proposed ROS-Based Gazebo Simulator

The authors have chosen to build Robotics Operating System-based Gazebo simulator after literature survey indicated that Open Dynamics Engine (ODE) provides accurate physics library and people are likely to use it [14, 6, 8]. We have built our custom Robotics Operating System and Gazebo simulator from scratch using ODE. We used ROS Indigo, Gazebo7 and Ubuntu 14.04 operating system. Pioneer 3DX is used as a rover with a 64×48 resolution running at 5 Hz. It has an angular velocity of 1 rad/s and has a velocity of 0.5 m/s. Copter is the quadrotor using 10 Hz LIDAR (only for validation) and 320×240 resolution camera refreshed at 5 Hz. We have successfully simulated the linear dynamics of the quadrotor with observable dynamics which is as shown in Eq. (8).

$$\begin{bmatrix} F \\ \tau \end{bmatrix} = \begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} a \\ \alpha \end{bmatrix} + \begin{bmatrix} \omega \times mv \\ \omega \times I\omega \end{bmatrix} \quad (8)$$

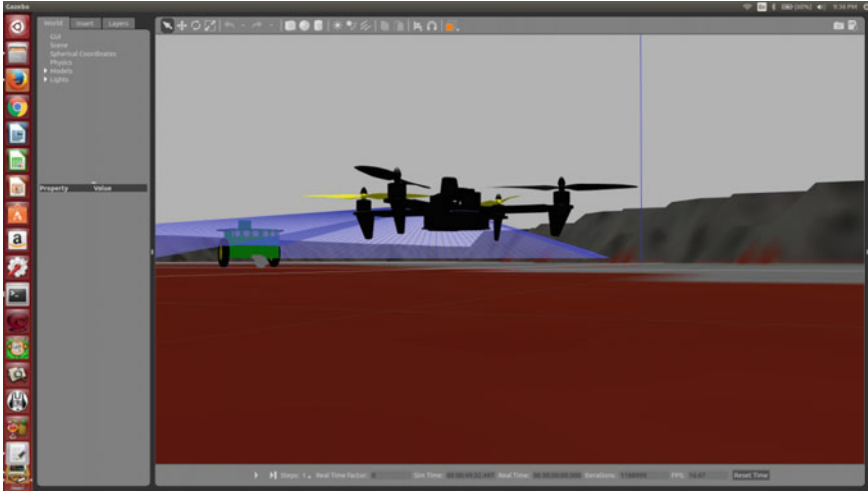


Fig. 4 Object tracking initialisation in ROS Gazebo software in the loop simulation

- F Net force of the quadrotor (thrust of the rotors)
- τ Net torque of the quadrotor (yaw)
- a Acceleration of the quadrotor
- ω Angular velocity vector
- I Inertia
- α Angular acceleration
- m Mass of the quadrotor
- v Velocity of the quadrotor.

Figure 4 demonstrates the initialisation of object tracking using the camera that is also simulated. The height is adjusted, and LIDAR is turned on so as to account for the calculation of tracking percentage and detection percentage. This method helps us test for conditions like occlusion without risking the breakage of physical hardware. Both rover and the quadrotor are being simulated in a single computer.

Figure 5 depicts the physics tests of our simulator. It may be easy to get visually nice looking simulator in Unity or Unreal. The control dynamics of quadrotor landing on Pioneer 3DX has been simulated with aerodynamics.

6 Parameters Setup

Android phone's NV21 images are converted to RGB. MJPEG streaming is achieved by integrating our custom OpenCV components built from scratch (with the help of Seed Studio's Webcam Android application) [15].

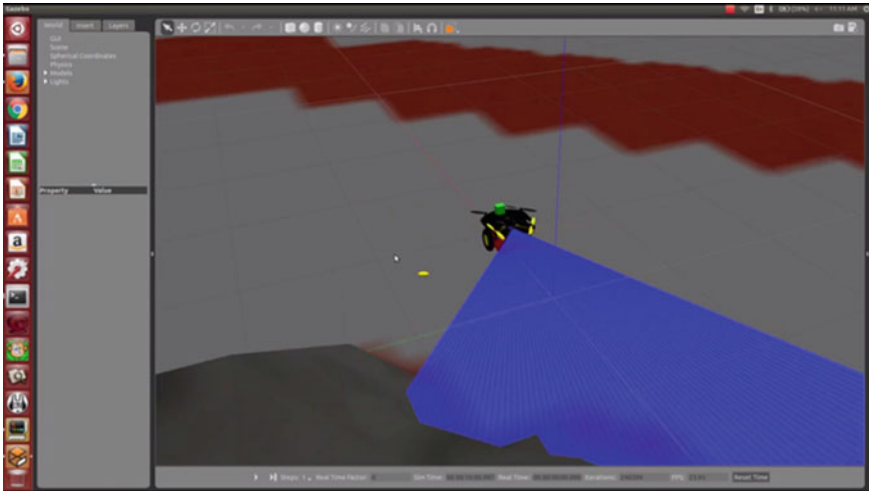


Fig. 5 Physics tests in ROS Gazebo software in the loop simulation

Arduino's servo maps the range of 45° – 135° to the range of 1000 – $2000 \mu\text{s}$ 50 Hz PWM. PWM values of 1400 – 1600 are used with 1500 as the mid-value. ORB feature matching is used along with brute-force Hamming matcher in Android phone for 10 frames of 9×9 kernel images. To avoid the frame rate dropping down to 1 frame per second when internal memory or microSD memory is used, cache variables are used in global class. Sony XPERIA Dual M2 mobile phone is being used with a resolution of 320×240 for input image. This resulted in processing at about 150 frames per second for colour thresholding range of $1/6$. Threshold for the features of 10 frames of 9×9 kernel images is 2 frames.

If γ represents the Hamming matcher threshold,

$$y = \begin{cases} \gamma + 2, & n_{\text{new}} < n_{\text{template}} \\ \gamma - 1, & n_{\text{new}} \geq n_{\text{template}} \end{cases}$$

7 Results

For 640×480 image, detection rate has been close to 97% and tracking rate has been close to 79% in the practical tracking validation (89% in SITL). Our StaGaus algorithm is robust in rotational invariance. It handles both in-plane and out-of-plane rotations of up to 30° on each side as demonstrated in Fig. 6.

The comparison of StaGaus with SRDCF and colour-filtered ORB tracking is shown in Fig. 7. Note that we have also included plain RGB-filtered tracking for reference (it loses track within first 20 s). SRDCF and StaGaus are close in both detection and tracking percentages. The main advantage is that SRDCF achieves

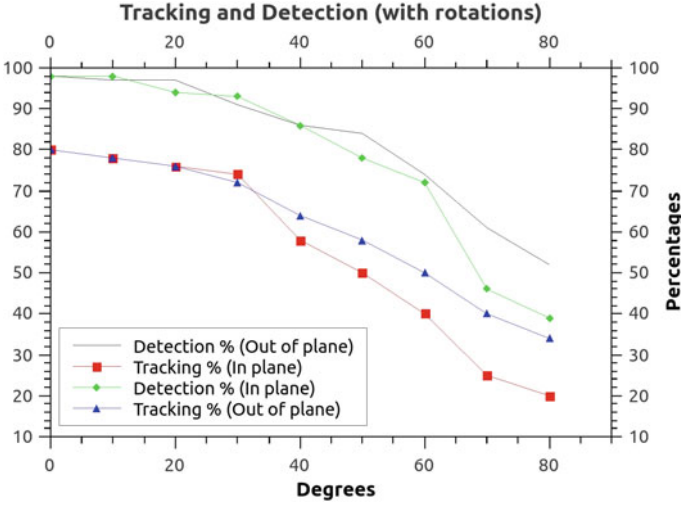


Fig. 6 Detection and tracking % of StaGaus algorithm observed with physical Android setup

detection and percentage through 2D images only without StaGaus controller. But, when control algorithm is used, StaGaus performs much better in real-time performance (by Big O considerations, by several times). Also, StaGaus does not have the position drift that SRDCF suffers. StaGaus uses the template as the reference so that the centroid is re-centred.

As mentioned in Fig. 7, the new designed algorithm is compared against three more algorithms. Some of the earliest attempts included primitive form of colour thresholding followed by blob detection by dilating and eroding (as in RGB thresholding with 1/6 thresholding range without feature matching or using separate controller algorithm that overrides FCU control algorithm). This method results in loss of tracking when the background has a similar colour to that of the object being tracked.

Similarly, for RGB thresholding with 1/6 colour range that is coupled with ORB feature matching with the default controller algorithm that FCU offers, the tracking is significantly better. But, yaw oscillations are observed when the object is directly in front of the quadrotor, thus decreasing the tracking rate slightly. When StaGaus controller algorithm is used, yaw oscillations are dampened making the flight stable. Thus, experiments with both frequency domain and spatial domains lead to StaGaus.

We have defined a metric $\psi = -\log_{10}\left(\frac{\% \text{ tracking}}{\text{memory} \times \% \text{ CPU usage}}\right)$ to assess the algorithms on time complexity, space complexity and tracking % with the same metric.

Table 1 contains the reported time complexity and space complexity of algorithms. For StaGaus and RGB, feature detection requires minimum of 9 pixels and n represents the number of features. Also, as StaGaus requires 10 frames of 9×9 image kernels, the appropriate offset is added. We consider 81 features, $S = 1$, $N_{ne} = 1$ and $K = 0$. Rest of the parameters are as they are (values authored in their paper).

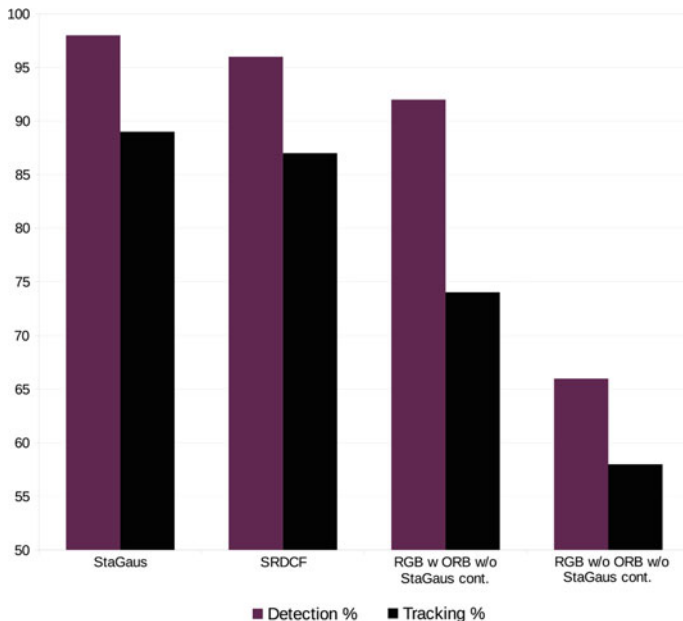


Fig. 7 Detection and tracking rates of various algorithms with reported rates and SITL

Table 1 Comparison of ψ for various algorithms (worst case for StaGaus)

Algorithms	Time complexity	Space complexity	Time	Space	ψ
StaGaus	Width \times height + $n + 9n$	Width \times height + $n + 9n + 10 \times 9 \times 9$	308,010	308,820	9.03
SRDCF	$dSMN \log MN + SMNN_{Ne} + (d + K^2)dMNN_{GS}$	$l^2 + d + \text{width} \times \text{height}$	66,300,583	307,282	11.4
RGB (1/6) ORB (without StaGaus Controller)	Width \times height + $n + 9n$	Width \times height + $n + 9n + 10 \times 9 \times 9$	308,010	308,820	9.11
RGB (1/6) (without StaGaus Controller)	Width \times height	Width \times height	307,200	307,200	9.21

$MN = 50 \times 50 = 2500$ and $N_{GS} = 4$. We ignore trivial constants and variables that do not significantly contribute to performance or memory.

It can be observed that higher ψ means that the algorithm does not adhere to memory and performance constraints of mobile phones. We can see that SRDCF is more than two orders higher than StaGaus on a log scale.

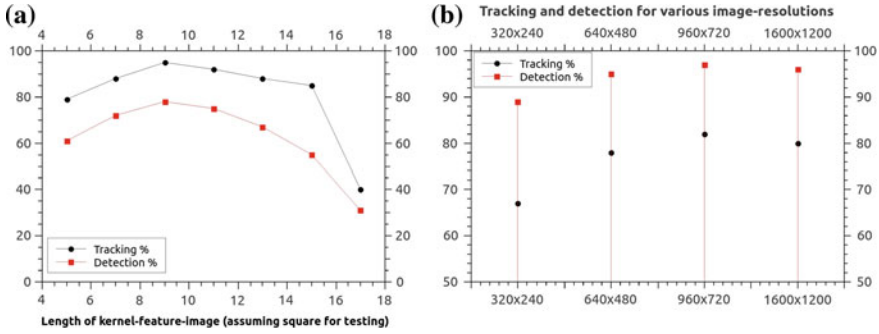


Fig. 8 a Length of kernel image, b image resolution results in right

As shown in Fig. 8a when the size of the kernel is too small, the features are mapped to the same cell resulting in loss of features. When the size of kernel is too large, noisy features will also be mapped resulting in the deterioration of tracking rate. When the image resolution gets higher, the detection rate and tracking rate get higher as depicted in Fig. 8b. However, when Android phone processor’s performance hits the peak, if the image resolution is increased, real-time processing no longer occurs and this results in the slight reduction of tracking percentage.

The authors have relied on Android phone for most of the processing and for real-time control, Arduino has been used. APM 2.6 flight controller has been used for low-level quadrotor dynamics.

Regarding the simulations, Table 2 compares the various simulators meant for object tracking drones. It is observed that our ROS and Gazebo simulator is capable of simulating high-quality visuals while still maintaining physics. We have used the metrics of evaluations as that of used in the popular literature [7, 6, 8]. “~” represents the dependence of implementation (For example, high-speed simulations are possible in V-REP at the cost of approximations in physics). “OS” in the cost’s column stands for open source.

As can be observed from Table 3, the cost of proposed and designed drone has been low (a by-product aspect) due to the StaGaus algorithm that has been used. It reduces the need of expensive sensors, and it is able to run on normal Android phones.

8 Conclusions and Future Work

This drone has been used with a horizontally placed camera. We have compared various image processing algorithms and demonstrated that our algorithm works on memory and performance-constrained devices like Android phones while maintaining a higher tracking efficiency than other popular algorithms. We have also built

Table 2 Comparison of various simulators meant for object tracking drones

Simulator	Cost	Physics	Testing conditions	Simulation pace	CPU %	Memory %	Visuals
V-REP	OS	~	~	Real time	Low	Low	High
MATLAB/Simulink	High	~	~	Slow	High	High	~
Unreal engine (including Airsim)	Free	~	High	Real time	Moderate	Moderate	High
ROS Gazebo (TurtleBot)	Free	✓	Moderate	Slow	High	High	High
ROS Gazebo (Ours)	OS	✓	High	Real time	High	High	High

Table 3 Costs of various object tracking drones

Drone	Tracking principle	Marketing stage	Cost
TraQuad	Imaging based	–	17,000 Rs
AirDog	Bluetooth based	Existing	107,312 Rs
Solo	FPV drone	Shut down	53,656 Rs
DJI	GPS based FPV	Existing	37,420 Rs (min)
Xiaomi	FPV drone	Existing	34,812 Rs
Parrot	FPV drone	Existing	18,700 Rs (min)
Nixie	Proprietary	Pre-order	–
Lily	GPS based	Shut down	61,704 Rs
Zero Zero robotics	Proprietary	Pre-order	–
GoPro	FPV drone	Pre-order	59,872 Rs

a ROS Gazebo simulator and have efficiently built a unified software platform for object tracking drone researchers.

Future work would be the addition of sensors (like one vertical camera) and performing sensor fusion on it using deep learning. Multi-terrain quadrotor would be a future work: The wheels would be used on roads (to save power) and the quadrotor would be flown when necessary (to save time). Swarm object tracking quadrotors for a single-object and multi-object tracking with swarm planning are possible research aspects. Another future work would be an ISO packaged with native GPU drivers of NVIDIA (the setup time would be reduced and the performance of the simulations would be much higher). To enable these developments, we have included the simulator’s setup and usage, placed the algorithms in the paper and open-sourced the Android apps’ code [16].

Acknowledgements We thank Vladimir Ermakov, leading developer of MAVROS for assistance regarding MAVROS. We thank Venkat, Edall Systems, for assistance regarding PWM RC values and also Harsha H N and Amrinder S R for their support in programming.

References

1. Yilmaz, A., Javed, O., & Shah, M. (2006). Object tracking: A survey. *ACM Computing Surveys*, 38(4), Article Number 13. <https://doi.org/10.1145/1177352.1177355>.
2. Smeulders, A. W. M., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A., & Shah, M. (2014). Visual tracking: an experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7). <https://doi.org/10.1109/tpami.2013.230>.
3. Yang, H., Shao, L., Zheng, F., Wang, L., & Song, Z. (2011). Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18). <https://doi.org/10.1016/j.neucom.2011.07.024>.
4. Li, R., Pang, M., Zhao, C., Zhou, G., & Fang, L. (2016). Monocular long-term target following on UAVs. CVPRW. <https://doi.org/10.1109/cvprw.2016.11>.
5. Staranowicz, A., & Mariottini, G. L. (2011). A survey and comparison of commercial and open-source robotic simulator software. PETRA 11. In *Proceedings of the 4th International Conference on Pervasive Technologies Related to Assistive Environments*, Article No. 56. <https://doi.org/10.1145/2141622.2141689>.
6. Blasco, P. I., Diaz, F., del Rio, M., Ternero, C. R., Muniz, D. C., & Diaz, S. V. (2012). Robotics software frameworks for multi-agent robotic systems development. *Robotics and Autonomous Systems*, 60(6), 803–821. <https://doi.org/10.1016/j.robot.2012.02.004>.
7. Mueller, M., Smith, N., & Ghanem, B. (2016). A benchmark and simulator for UAV tracking. In *Computer vision, ECCV*. https://doi.org/10.1007/978-3-319-46448-0_27.
8. Ivaldi, S., Padois, V., & Nori, F. (2014). Tools for dynamics simulation of robots: a survey based on user feedback, Humanoid Robots (Humanoids). In *2014 14th IEEE-RAS International Conference*. <https://doi.org/10.1109/humanoids.2014.7041462>.
9. APM 2.6 Documentation. <http://ardupilot.org/copter/docs/common-25-and-26-overview.html>.
10. Rother, C., Kolmogorov, V., & Blake, A. (2004). GrabCut interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics, SIGGRAPH*. <https://doi.org/10.1145/1015706.1015720>.
11. Greig, D. M., Porteus, B. T., & Seheult, A. H. (1989). Exact maximum a posteriori estimation for binary images. *Wiley for Royal Statistical Society*, 51(2), 271–279.
12. Color Models, Image Color Conversion, Volume 2, Image processing, Intel Integrated Performance Primitives for Intel Architecture Developer Reference. <https://software.intel.com/en-us/node/503873>.
13. Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT and SURF, Willow Garage. In *ICCV Proceedings of the 2011 International Conference on Computer Vision*, pp. 2564–2571. <https://doi.org/10.1109/iccv.2011.6126544>.
14. Configuring your Environment, Erle Robotics. http://docs.erlerobotics.com/simulation/configuring_your_environment.
15. Sseed Studio's Webcam Android Application. <https://github.com/xiongyihui/Webcam>.
16. Our core softwares for physical quadrotor. <https://github.com/traquad>.

IoT-Enabled Medicine Bottle



A. R. Shreyas, Soumya Sharma, H. Shivani and C. N. Sowmyarani

Abstract Internet of Things (IoT) has facilitated a reassuring opportunity to develop powerful and dynamic industrial systems and applications by making use of universality of sensors, actuators and network connectivity. IoT can immensely benefit people when used in the medical field. Non-adherence to a medical routine is a major problem faced by patients. According to the World Health Organization, increasing the effectiveness of adherence interventions may have a far greater impact on the health of the population than any improvement in specific medical treatments. In an attempt to address this problem, in this paper, a special system that monitors and tracks the consumption of medicine of any patient is presented. Adherence needs are met through the implementation of a mobile application. The result of this will be timely consumption of the medicine by the patients.

Keywords Internet of things (IoT) · NodeMCU · Infrared sensor · Ultrasonic sensor · Firebase cloud

1 Introduction

The Internet of Things (IoT) has facilitated the development of dynamic and powerful industrial systems. Network connectivity and integration of individual physical devices being the foundation of IoT have found use in multitudinous fields, with each field constructing its own platform on this very foundation. Currently, the agricultural industry, automobile industry, oil and gas industry and of course, the medical field heavily depend on IoT [1–4]. Harnessing the power of IoT leads to efficient solutions that cater to diverse needs, and present a complete package that serves a singular aim.

Technical advancements in modern medicine have been rampant, providing waves of solutions to diseases that were considered incurable a couple of decades back. However, at the most basic level, the efficacy of those advancements comes about

A. R. Shreyas (✉) · S. Sharma · H. Shivani · C. N. Sowmyarani
Department of CSE, R.V College of Engineering, Bangalore 560059, India
e-mail: ars.shreyas@gmail.com

only when the medicines are consumed at the right time, in the right quantity. Not adhering to a routine might lead to catastrophic results, like severe degradation in a patient's health, rendering the use of modern medicine futile.

Currently, the solution in use place to the aforementioned problem is the employing of helpers or caretakers to constantly monitor the patients or the use of alarm clocks. The former case maybe expensive and unreliable, while the latter has several restrictions, as it is not integrated medicinal bottles, to be able to gauge the changes in quantities consumed by the patient.

With the creation of IoT-enabled medicine bottle, and the support of an application, not only can timely reminders be sent to the patient on his mobile phone, but also the quantity consumed can be monitored as well. Through this implementation, several other features can be reinforced.

Such a system is efficient and systematic, as is explained through the course of the paper. The use of sensors coupled with a WiFi (Wireless Fidelity) module provides the hardware support. Data analysis and integration is done on the firebase platform, the result of which is elegantly presented in a simple to use application. Hence an IoT based system has been successfully implemented.

2 Motivation

The treatment of most diseases invariably includes the use of pharmacotherapy, which refers to therapy using pharmaceutical drugs. However useful these medicines maybe, adherence is the key to make them effective. According to a WHO (World Health Organization) report, adherence rates in developed countries is a mere 50% and worse in developing countries. Among patients with chronic illness, approximately 50% do not take medications as prescribed [5]. Thus, there is an urging need to oversee the medical consumption on a daily basis. A severe example is of hypertension (high blood pressure), where according to WHO, up to 80% patients suffer because of non-adherence. Specifically, they claimed that improving adherence to medical therapy for conditions of hypertension, hyperlipidemia, and diabetes would yield very substantial health and economic benefits. In another bold statement, they stated that a rise in adherence may have a greater effect on health than improvement in specific medical treatments. Effective solutions to combat this problem are not in place currently either. Thus, motivated by the multiple ramifications of non-adherence, and a lack of an existing efficient system, the IoT-enabled medicine bottle was developed.

3 Related Work

In this section, we will briefly highlight the related researches of this paper. There are multiple researches that have proposed various models with respect to IoT-enabled medicine applications. First, a medication adherence monitoring system was devel-

oped that is based on a wearable sensor [6]. Authors of this paper focus on two actions of twist-cap and hand-to-mouth and the act of pill intake is then identified by performing a moving window dynamic time warping in real time between signal templates and the signals acquired by the wearable inertial sensor. In another research, the authors have developed an alarm system for the community health care [7]. They track state changes of medicine bottles by using weight sensors. According to the tracking record, their proposed system will decide whether or not to alarm the patient. Rohokal et al. proposed a feasible IoT approach for the better health surveillance of poor and rural human being's health issues related to any part of the body [8]. In another system, an IoT framework was developed where medicine intake was ensured using activity recognition and an activity classification scheme [9]. Other work has been done to make intelligent medication boxes [10–12], and also to improve medicine adherence [13].

4 Design

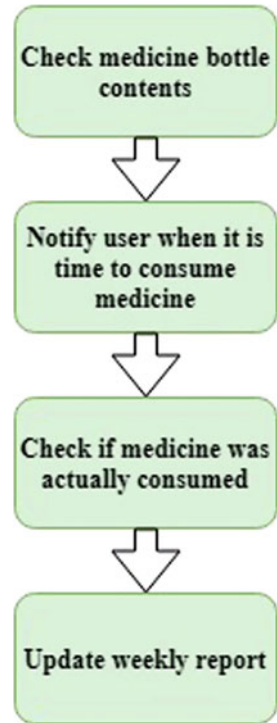
This section proposed the system architecture. The system is composed of three modules, where the first one is the alarm module, which notifies the user when it is time to consume medicine. The second module is the sensor module that continuously gathers sensor outputs and conveys them to the third module, which is the classifier module. This module determines whether medicine has been consumed or not.

The various steps are illustrated on a flow diagram in Fig. 1.

The IoT has facilitated the development of dynamic and powerful the system can be divided into various steps

1. The user sets the times at which medicines have to be consumed. The user does this on the mobile application that has been specially designed for the system.
2. When it is time to consume the medicine, the mobile application notifies this to the user.
3. The system waits for a certain time duration, after which the classifier module decides whether the user has actually consumed the medicine or not. The classifier module does this based on the results obtained from the sensor module.
4. If the medicine has not been consumed, the user is notified another time.
5. The system waits for another time duration, after which the system once again checks the status of the medicine consumption. The result obtained is added to the weekly report that can be viewed in the application.
6. In case the amount of medicine in the bottle is less than a certain threshold, the user is notified.

Fig. 1 Flow diagram for the system: after the medicine timing is set the above steps are followed whenever it is time to consume medicine



5 Implementation

In order to achieve the medication monitoring system, an IoT-enabled medicine bottle is developed. The user sets reminders on the mobile application, for the tablets he needs to consume as prescribed. Our surveillance approach includes the detection of whether the bottle is opened or not and also the difference in heights before and after the medicine consumption. The decision for the second action is taken only after the first action, i.e., only after the bottle cap is opened. The decisions taken are based on the sensor readings sent to the cloud database. Figure 2 shows the various components of the system.

FC-51 (Infrared Sensor), Infrared Obstacle Avoidance Proximity Sensors Module has a built-in Infrared transmitter and receiver that emits IR (InfraRed) energy and checks for a reflected IR energy in order to detect any obstacle in front of the sensor. This inexpensive sensor is used to detect if the bottle cap is opened or not.

Further, HC-SR04 (Ultrasonic Sensor) can be used to measure small distances [14, 15]. It is used here to measure the difference between the height of medicine bottle before and after medical consumption. This economical sensor provides 2–400 cm of noncontact measurement functionality with a ranging accuracy that can reach up to 3 mm.

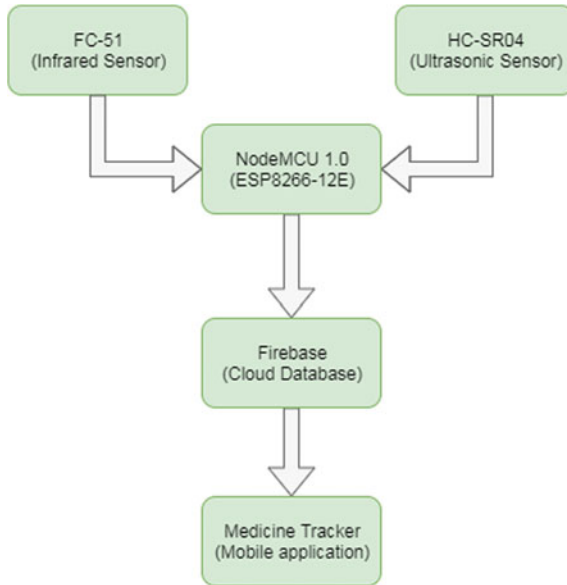


Fig. 2 Implementation of the proposed system: readings from the infrared sensor (FC-51) and ultrasonic sensor(HC-SR04) are monitored by the NodeMCU, and data is sent to the mobile application via firebase

The data from these two sensors is collected by a versatile open source IoT platform, NodeMCU 1.0 (ESP8266-12E). Figure 3 shows the hardware components wired together which act like the bottle cap.

This sensor is initially programmed once using Arduino IDE 1.8.4 to perform the function of a microprocessor for the sensors and as well as a WiFi module to transmit the data to Firebase which is a mobile and web application development platform. The NOSQL (non-Structured Query Language) feature of Firebase real-time database makes it extremely flexible to store the information from hardware [16]. It also permits the user to view the information obtained from the hardware components in a real time from any device with permissions that has access to the network. It stores data in key-value pairs. The necessary information that the database needs to store is whether the bottle is opened or not, which can be identified using Boolean values and the height of the medicine stack in the bottle, which can be stored using floating point values.

Figure 4 shows the IoT-enabled medicine bottle.

The mobile application is developed with the help of Android Studio. This application collects the data from firebase and takes required decisions accordingly.

Fig. 3 Medicine bottle cap-infrared and ultrasonic sensors attached to the cap along with the NodeMCU

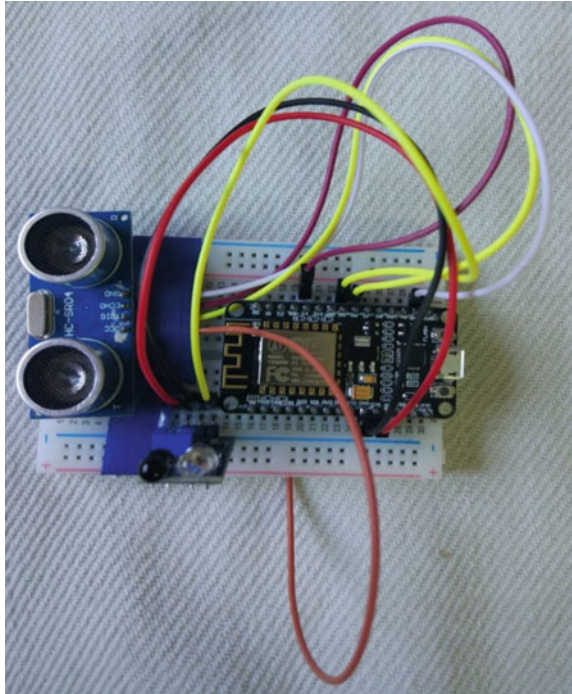
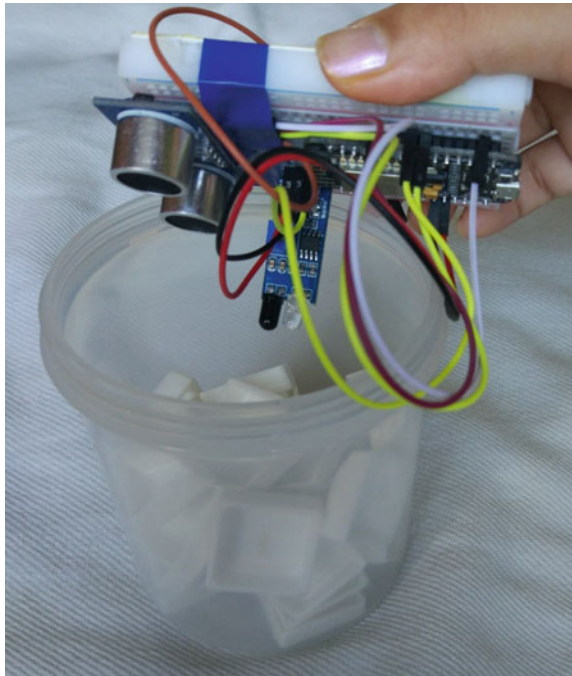


Fig. 4 Cap being placed on the medicine bottle

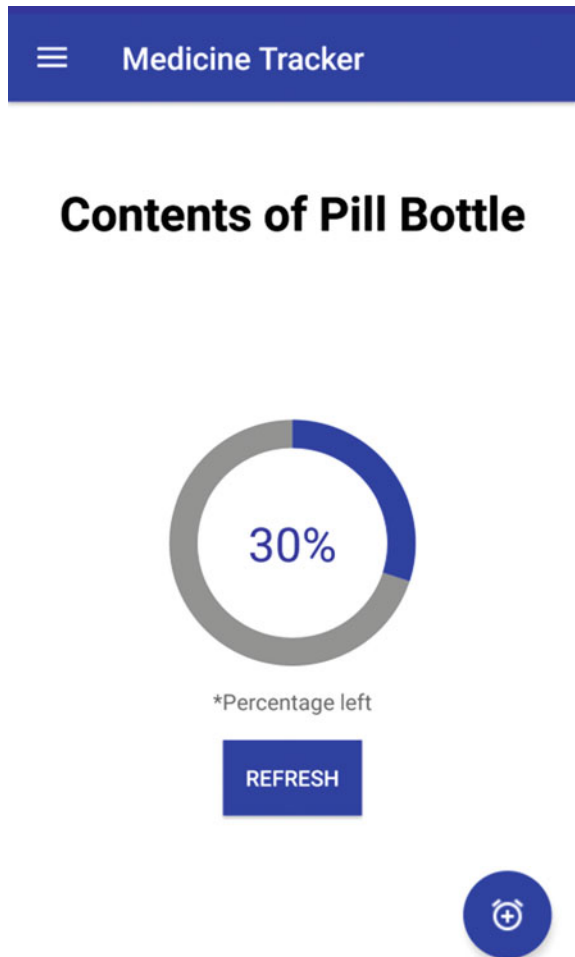


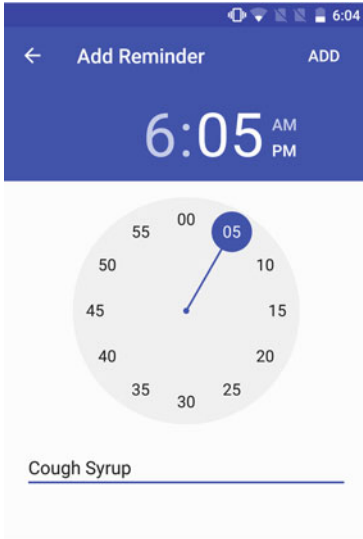
6 Results

The system is developed to detect changes in the contents of the medicine bottle in real time and respond to the changes appropriately by notifying the user. The system also notifies the user in case the contents of the bottle are going empty.

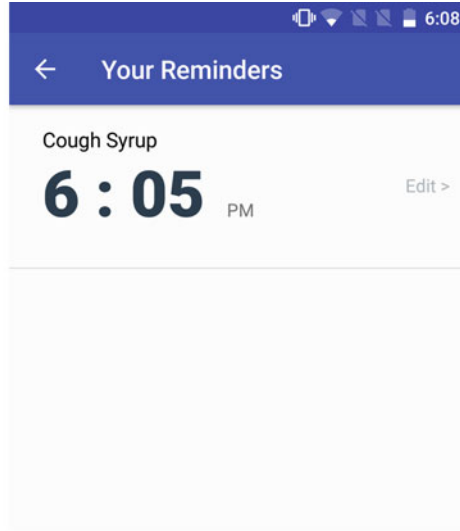
The results of the experiments are shown in the following section. Figure 5 shows the current contents of the medicine bottle. In Fig. 6a, the user sets a reminder on the mobile application. The user gets a notification (as shown in Fig. 6b at the time in which he sets the reminder, to consume the medicine. In case the user does not consume medicine at the right time, the android application reminds the user on another occasion as shown in Fig. 6c. A weekly report is also generated as shown in

Fig. 5 Page on the application showing contents of medicine bottle

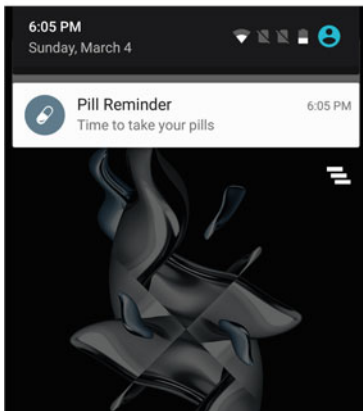




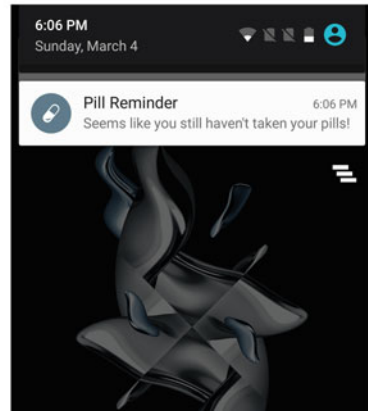
(a) Setting reminder



(b) Viewing the set reminders



(c) First notification



(d) Second notification when pills were not consumed

Fig. 6 Performing an experiment using the proposed implementation

Fig. 6d. In the same figure, the Yes or No signifies if the pills were consumed when the reminder went off, or/if they were not consumed.

The remainder of the section presents statistics obtained after performing trials.

In Table 1, the first column indicates if the bottle was actually opened or not. The next two columns show a comparison between the depths of the top of the medicine from the ultrasonic sensor. The fourth column indicates whether the medicine was

Table 1 Results after 6 trials

Bottle opened?	Initial depth (cm)	Final depth (cm)	Medicine actually consumed?	Report (shown in app)	Report correct?
No	4.56	4.56	No	No	Correct
No	3.86	3.86	No	No	Correct
Yes	6.79	7.02	Yes	Yes	Correct
Yes	5.23	5.51	Yes	Yes	Correct
Yes	4.56	4.54	No	No	Correct
Yes	3.86	4.10	No	Yes	Incorrect

actually consumed or not. The next column shows the report generated by the mobile application. Lastly, the correctness of each report is mentioned.

There are three outcomes to the experiment.

1. The bottle is not opened.
2. The bottle is opened and medicine is consumed.
3. The bottle is opened but medicine is not consumed.

The bar charts in Figs. 7 and 8 show the results obtained after performing 60 trials using syrup and pills as the medicine. The *x*-axis shows the various outcomes to the experiment. The *y*-axis shows the count obtained in each case as mentioned in the chart.

From these results, we can estimate the accuracy of the system.

$$\text{Accuracy} = \text{Number of Correct Results} / \text{Number of Trials.} \tag{1}$$

For syrup as medicine,

$$\text{Accuracy} = (20 + 18 + 19) / (20 + 20 + 20) = 0.95$$

So the system has an accuracy of around 95% with syrup as the medicine.

For pills as medicine,

$$\text{Accuracy} = (20 + 18 + 15) / (20 + 20 + 20) = 0.883$$

So the system has an accuracy of around 88.3% with pills as the medicine.

From these results, we can conclude that the system works more accurately when syrup is stored in the bottle rather than pills.

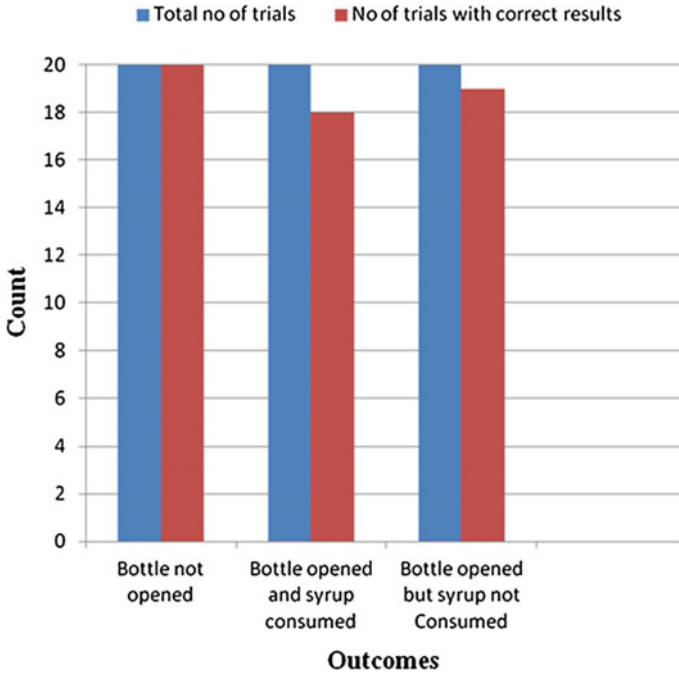


Fig. 7 Accuracy evaluation of the proposed system with syrup as the medicine

7 Limitations

The proposed model suffers from three main limitations. First, the system is unable to guarantee that the patient actually consumes the medicine, ensuring only the removal of the medicine from the bottle. Second, a majority of the patients that are unable to adhere to their medical routine include the elderly who may not possess sufficient knowledge on how to use the application and to utilize its benefits to the fullest extent. Helping them understand the system maybe the first hurdle, and the elderly maybe be unwilling and unresponsive to such a method of improving adherence. Thirdly, since the model monitors variations in height to predict when to send notifications, insignificant changes in the height may go undetected by the sensors, leading to erroneous results. This, however, can be eliminated by using more powerful hardware support. Also, the bottle must be kept away from sunlight at all times, as the sensors are sensitive to sun rays. However, most medicinal bottles are generally stored in dark and cool places so this limitation can be eliminated with ease too. Additionally, trivial reasons like the phone being out of battery or switched off may cause hindrances in the efficient results that the system aims to provide.

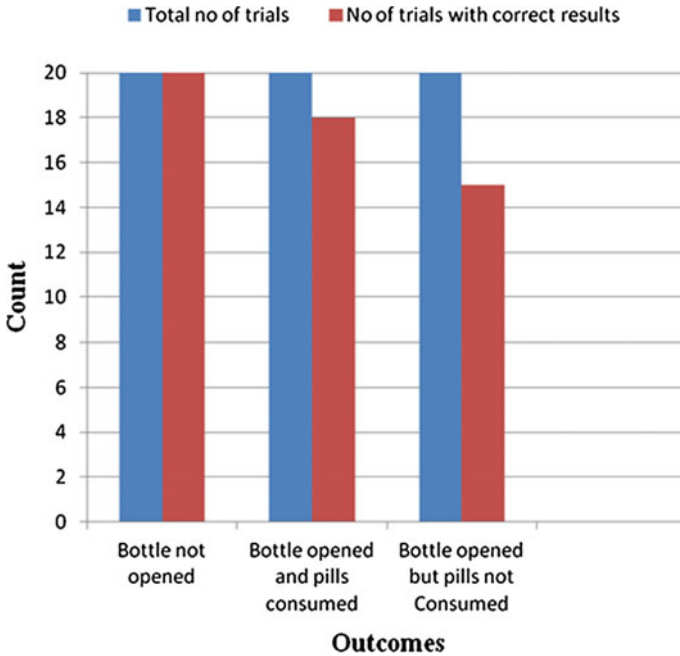


Fig. 8 Accuracy evaluation of the proposed system with pills as the medicine

8 Future Work

A potent method to make the system more streamlined and effectively monitored is by extending the application to provide doctors a view of their respective patients’ routines, and view their adherence levels regularly. This can help the patients and their doctors to work towards recovery in a more steady and harmonious manner.

Also, to make the medicinal bottles easy to carry and use, the model can be applied to a smaller medicinal bottle, which requires the assistance of more powerful hardware support. By improving the strength of the hardware used, the same model maybe applied to medicines of a relatively smaller dimension too, which as earlier mentioned is a limitation of the current model. The IoT-enabled bottle may also be presented in varying shapes, to suit the needs of different medicines. Encasing the hardware over the bottle, in order to provide it protection from any liquid medicine or any powdered medicine is another requirement, which might otherwise spoil the hardware.

Another technique to improve the current model, maybe to provide it in multiple languages. As already stated, the elderly forms a major consumer base for the model, and being provided of a solution in their native language might prove to be a relief to many, and increase the reach of the model.

9 Conclusion

The medicine bottle developed by incorporating the principles of the IoT aims at helping patients recover speedily and in a steady manner. On entering the quantity to be consumed, and setting reminders on when to consume the medicines, the patients can remain assured that they need not keep track of the number of times they have consumed in the day, the application will take care of that by sending notifications when it is time to consume. Weekly reports of consumption and adherence can be shared with the doctor or family members too.

References

1. Uddin, M. A., et al. (2017). Agriculture internet of things-Ag-IoT. In *27th International Telecommunication Networks and Applications Conference (ITNAC)* (pp. 1–6).
2. Zhang, X., et al. (2017). Monitoring citrus soil moisture and nutrients using an IoT based system. *IEEE J. Sens.*, 17(3), 430–447.
3. Kabir, M. H., et al. (2017). A low cost sensor based agriculture monitoring system using polymeric hydrogel. *J. Electrochem. Soc.*, 164(5), 107–112.
4. Mahendra, B. M., et al. (2017). IoT based sensor enabled smart car parking for advanced driver assistance system. In *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)* (pp. 2188–2193).
5. Brown, M. T., & Bussell, J. K. (2011). Medical adherence: WHO cares? *Mayo Clinic Proceedings*, 86(4), 304–314.
6. Chen, C., et al. (2014). A medication adherence monitoring system for pill bottles based on a wearable inertial sensor. In *2014 36th Annual International Conference on Medicine and Biology Society (EMBC)*. IEEE.
7. Sohn, S. Y., et al. (2014). Alarm system for elder patients medication with IoT-enabled pill bottle. In *2015 International Conference on Information and Communication Technology Convergence (ICTC)* (pp. 60–61). IEEE.
8. Rohokale, V. M., et al. (2011). A cooperative internet of things (iot) for rural healthcare monitoring and control. In *2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE)* (pp. 1–6). IEEE.
9. Serdaroglu, K., et al. (2015). Medication intake adherence with real time activity recognition on IoT. In *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)* (pp. 230–237).
10. Gomes, C. E. M., et al. (2015). Extending an intelligent medicine cabinet through the use of consumer electronic devices in order to increase the medication adherence. In *2016 6th International Conference on Information and Communication Technology for the Muslim World (ICT4M)* (pp. 98–102).
11. Xu, L., et al. (2014). A health-IoT platform based on the integration of intelligent packaging, unobtrusive bio-sensor, and intelligent medicine box. *IEEE Transactions on Industrial Informatics*, 10(4).
12. Magalhães, L., et al. (2017). A three-staged approach to medicine box recognition. 2017 24^o Encontro Português de Computação Gráfica e Interação (EPCGI), pp. 1–7.
13. Rodrigues, M. A. S., et al. (2015). An intelligent medication system designed to improve the medication adherence. In *2015 IEEE 5th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)* (pp. 46–49).

14. Zahir, E., et al. (2017). Implementing and testing an ultrasonic based mobility aid for a visually impaired person. In *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)* (pp. 453–456).
15. Mantoro, T., & Istiono, W. (2017). Saving water with water level detection in a smart home bathtub using ultrasonic sensor and Fuzzy logic. In *2017 Second International Conference on Informatics and Computing (ICIC)* (pp. 1–5).
16. Alsalemi, A., et al. (2017). Real-time communication using firebase cloud IoT platform for ECMO simulation. In *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)* (pp. 178–182).

Revamp Perception of Bitcoin Using Cognizant Merkle



J. Vijayalakshmi and A. Murugan

Abstract The bitcoin network adopts the regulation of peer-to-peer network for storing and retrieving data. Data synchronization and consistent verification is important in peer-to-peer networks for proving truth of data. Proof of existence of data is important for many applications and it can be verified by means of searching process. The goal of this paper is to minimize the time of verifying the transaction residence in the shared ledger of bitcoin network. In global ledger management like blockchain, the bitcoin transaction verification and validation is fundamental which is mainly used by miners for providing the proof-of-work for the transactions to achieve block reward and providing trust among the peers. In this paper we have provided a new way of verifying the transaction existence in blockchain by means of altering the data structure of Bitcoin Merkle into a new form of Cognizant Merkle which modifies the structure of existing bitcoin system into a new form which uses less memory and achieve more speed compared to Bitcoin.

Keywords Bitcoin · Mining · Bitcoin Merkle · Cognizant Merkle · Proof-of-Work

1 Introduction

The evolution of financial crisis in the year 2008 gives rise to the landing of cryptocurrency like bitcoin in the business world. This cryptocurrency provides secure digital transmission, reduces the transaction complexity, and prevents the seeking of intermediaries like security brokers, insurance agents, financial lawyers and credit card companies [1]. The need for electronic commerce payment system based on lexicography proof instead of centralized acception is that the transactions are irreversible, preventing from fraudulent transaction and protecting the buyers from deceptive person across the network [2]. Bitcoin is a first decentralized cryptocurrency which was

J. Vijayalakshmi (✉) · A. Murugan
Research Department of CS, Dr. Ambedkar Government Arts College, Chennai 600039, India
e-mail: jeyamaha2002@gmail.com

A. Murugan
e-mail: amurugan1972@gmail.com

© Springer Nature Singapore Pte Ltd. 2019
N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_12

introduced by *Satoshi Nakamoto* in the year 2009 for implementing virtual commodity. This cryptocurrency is treated as “Gold standard” which is widely accepted for providing services by various retailers like Amazon, Subway and Victoria Secret [3].

Bitcoin is the basis of digital money and it act as currency unit that is used to cache and broadcast the value among the members across the blockchain. Bitcoin network is available as free source software which can be run on multiple computing devices like laptops, smart phones and it can be communicated through internet and other transport networks. This can be acquired, sold and transferred at specialized currency exchanges in the perfect form of money for fast, secure and border less in internet. Bitcoin is flip-side of virtual money which is fully distributed across peer-to-peer network [4]. The following few features make the bitcoin transaction more appealing for vendors, they are bitcoin transactions are publicly available. They can have multiple inputs and multiple outputs and the consumer and merchant are identified through public keys and public–private key pairs [5]. The motivation behind the usage of bitcoin is anyone can use bitcoin anywhere around the world. Even novice user can use this bitcoin for buying and selling their products like any other fiat currencies [6].

2 Bitcoin Mining

Bitcoin system is connected with users along with wallet containing key functions which are propagated across the network. The miners are responsible for producing the consensus blockchain, which is the authentic ledger of all the transactions. Mining process discover new bitcoins through a solution to the associated difficult problem. Any colleague in the bitcoin network running full bitcoin node may act as a miner. For every ten minutes a new quick fix is found by someone who then confirm transactions for past ten minutes and compensated with fresh unused 25 bitcoins [2]. The bitcoin contract includes implicit algorithms that coordinate the mining task across the bitcoin network which prevents the central repository need in financial institutions. The key innovation of a miner [7] is the use of distributed computation algorithm (Proof-of-Work) which is used to conduct a global election for every ten minutes for arriving the consensus about the state of transactions [4].

Transferring of bitcoins from one or numerous source accounts to one or different destination accounts is called transaction [7]. A bitcoin transaction reveals the blockchain network that the bitcoin owner is authorized to transfer some bitcoins to another owner. Another transaction is created by the endorsed owner who can now spend these collected bitcoins and by licensing the ownership transfer of bitcoins to another in a chain of ownership. Transactions are analogous to double entry bookkeeping ledger. The summation of input and output may not be same. The transaction productivity is always less than total aid where the change is paid as transaction reward for miners for including the transaction in the shared ledger. Not all existing transactions in the network are included in the blockchain only those transactions that are verified by miners are included in the blockchain. Miners do the transaction

proven process by accumulating transactions into blocks which require huge amount of computation to prove but only a less amount of computation to justify.

The principle of mining is to create new bitcoins in each block and to create trust by ensuring the authorization of transaction. The transactions are organized as an ordered back linked list of blocks which is used as representation of blockchain data structure. Each block is linked back to the previous block in the chain. Block height is calculated from recently added block to the initial block. Each block is identified through hash on the block header and the sequence of hashes looking each block to its parents creates a chain to the genesis block. The structure of block includes blocksize, block header, transaction counter and transactions. The block header format includes software version number, hash of previous block, merkle root, blocktime stamp, target difficulty and Nonce value. Blocksize denotes the total bytes of blocksize. Transaction counter indicates the statistics of transactions where transactions indicates recorded agreement in this block.

The Merkle root points out the root hash that comprises of all subordinated transaction hashes within the current block. The timestamp denotes the approximate block creation time. The target value gives the block Proof-of-Work. The nonce value is a counter used for the proof-of-work. Each block contains a merkle tree which contains the summary of all the transactions in the block. The proposed work is focused on the revision of the merkle tree data structure which achieves less memory and more speed for verification of transaction survival. The transaction data existence in current block is identified using merkle tree and its proof is verified using previous blockhash. Merkle trees are exercised in distinct fields like Digital Currency (Bitcoin, Ethereum), Global Supply Chain (IBM and Laersk), Health Care Industry (Deep Mind Health), Capital Markets, Git and Mercurial platforms and Apache Cassandra [8].

Merkle [9], patented the concept of merkle tree in the year 1979. In peer-to-peer network, data verification is important because the cloned data may exist in multiple locations. Using merkle tree, data verification, synchronization and consistency can be easily achieved. Merkle trees ensure the authenticity of the data by verifying the data before downloading a file from peer-to-peer nodes and it can also be used to verify the information coming from the untrusted resource. The root hash is obtained from the trusted source and then lower nodes of the merkle tree which is a part of the same tree can be obtained from the untrusted peers. The nodes from the untrusted sources are checked against the trusted hash for matching process. If they match then the nodes are accepted and the process continues if not, they are discarded and searched again with the different source.

3 Literature Survey

Current bitcoin system uses merkle tree implementation based on Binary Hash Tree (BHT) which occupies more memory and more time for searching and data retrieval. For this, alternative data structures which support string data operations based on

tree is analyzed. One of the alternative data structure for BHT is a trie data structure. The other alternatives are hash table which is efficient and relatively fast but unsorted. A *trie* is a data structure which is an alternate to BHT, which is used for storing strings in sorted order which facilitate the dynamic support of data structure [10]. Searching in the trie is faster which require separate pointer traversal for each character in the query string.

Alternatively Heinz et al. [11] proposes a *Burst-trie* which is considered as efficient for small set of keys or values which is stored in container providing faster retrieval than a conventional tree. A Burst-trie is a data structure similar to trie but the leafs of the trie are replaced with container which is able to store a small number of string efficiently. Primary goal of Burst-trie is to lessen the moderate the number of string comparisons during a process of searching a string but it follows storing more frequent terms than less frequent terms [10]. Our target is to analyze the performance of different prototypes such as hash table, tries and others on Resource Description Framework (RDF) [12].

Of this a hybrid data structure of HAT-trie [11] which was proposed by Nikolas et al. is considered for implementation. The reason for this is, cache-attentive data structure which supports the combination of trie along with hash table and Burst-trie. HAT-trie is an extension scheme of Burst-trie which replaces the linked list containers with cache aware based hash tables [13]. Ruslan Mavlyutov et al. proposed that HAT-trie is best option for memory utilization, packing time and look up for string in the tree data structure.

4 Bitcoin Merkle

In Bitcoin, merkle trees are used to encode blockchain data in a secured manner. Merkle trees are used to produce a *Digital Finger-print* of the entire set of transactions. This can also be used to verify the existence of transactions in the block. Simplified Payment Verification (SPV) [2] nodes in the bitcoin network uses merkle trees. SPV nodes can be used to verify the payments without running a full network node. Using the merkle path, the SPV node connect the bitcoin transaction to the appropriate block and also verify its inclusion within that block. The transaction existence within the blockchain or not is identified by the combination of the links between transaction and block, and the link between block and the blockchain. In Bitcoin, a merkle tree is constructed by recurrent hashing the nodes pair until there is only one hash called merkle root which is the public key of merkle signature system which requires estimation of 2^h leaves and $2^h - 1$ hash computations where h denotes the merkle tree height [14].

A BHT is a self balancing tree whose frequently accessed nodes are moved close to the root and every terminal node is labeled with a data block and every non terminal node is labeled with the cryptographic hash of the labels of its child nodes. Double SHA256 cryptographic hash is used for increasing the efficiency and improving the security of bitcoin transaction. The merkle tree structure is based on BHT, the tree

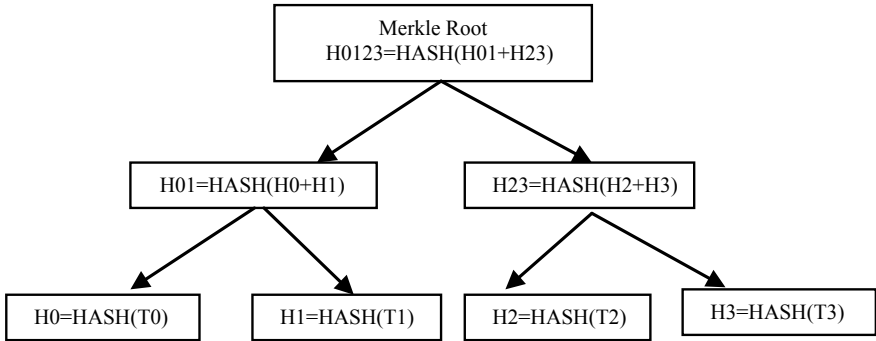


Fig. 1 Merkle tree construction based on even number of nodes

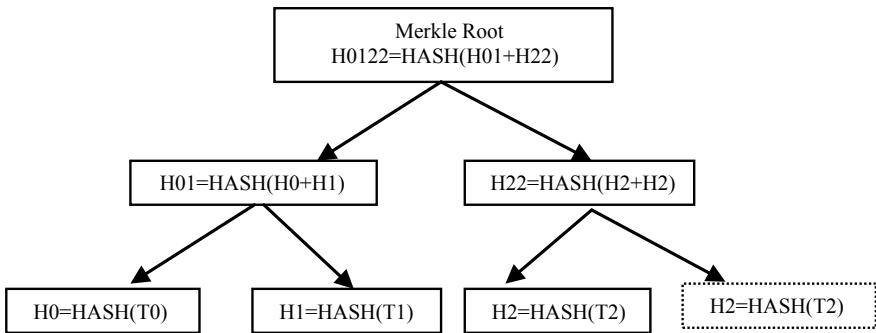


Fig. 2 Merkle tree construction based on odd number of nodes

must follow the representation of even number of transactions. If the leaf contains even number of transaction it is used as it is for merkle tree which is showed in Fig. 1. If number of transaction is odd sequence, the last transaction will be duplicated to achieve counterbalanced which is depicted in Fig. 2. Look at an example of merkle tree development in bitcoin transactions T_0, T_1, T_2, T_3 which is represented as leaves of the merkle tree. The hashes of the transactions are stored directly in merkle trees in each terminal node like H_0, H_1, H_2, H_3 . The hashes are determined by $H_0 = \text{SHA256}(\text{SHA256}(T_0))$ and $H_{01} = \text{SHA256}(\text{SHA256}(H_0 + H_1))$. In Fig. 1 merkle tree is produced for even number of nodes. Initially transaction hashes for T_0, T_1, T_2 and T_3 is calculated as H_0, H_1, H_2 and H_3 . Then intermediate hashes are computed at each level to achieve the final hash merkle root as H_{0123} . In Fig. 2 merkle tree is established based on odd number of nodes.

In a blockchain, every block contains few hundreds of transactions. If someone wants to verify the transaction existence in a specific block then he needs to download the entire blockchain. Instead, the proposed approach can download only a branch of the merkle tree in the blockchain. The blockchain contains those transactions obtained using *Light Client* or *Simplified Payment Verification (SPV)*. This can check

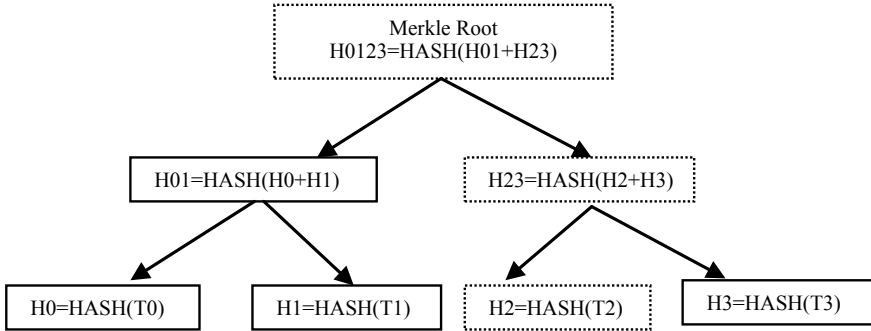


Fig. 3 Merkle tree path used to justify the incorporation of data

the hashes on traversing up the tree branch. If these hashes are equal then we know that this particular transaction exists in this block. When a SPV node inspect the particular transaction say T_2 in the merkle tree, then it can prove the inclusion of transaction T_2 in the block by producing a merkle path which consists of hashes H_2 , H_{23} and H_{0123} from the merkle tree and find out whether it is included in that block or not. With these hashes a SPV node can confirm that this transaction T_2 is included in the merkle tree. The above searching process is depicted in the following Fig. 3 which states whether the transaction T_2 is included or not.

5 Cognizant Merkle

The proposed system aims to achieve faster searching of transactions available in the merkle tree structure for its existence and its nonexistence. The recommended system combines the features of merkle tree along with HAT-trie to achieve more speed compared to Bitcoin Merkle. HAT-trie can be represented in two alternatives namely *hybrid* and *pure* HAT-trie which is exhibited in Fig. 4. HAT-trie variants are based on the technique of bucket management and splitting [15]. Pure HAT-trie employs the approach of burst-trie, which is faster but requires extra space for application. In pure HAT-trie, a single parent pointer is used as reference when the buckets contain strings that share only one common prefix. In a pure HAT-trie, a full bucket is bursted into small buckets of at most n buckets where n is the size of alphabet that are parented by a new trie. The removal of leading character in strings that are stored in the pure buckets are depicted in Fig. 5. Original strings can be consumed during the splitting procedure.

The *hybrid* HAT-trie employs B-trie splitting algorithm which reduces the buckets number with low cost and support several pointers to a bucket for supporting fast access. In case of hybrid HAT-trie, the last character part of prefix is stored along with string and the buckets also share a single prefix, but it is not removed. This has more than one parent pointer. This may also contains pure buckets. HAT tries are assembled

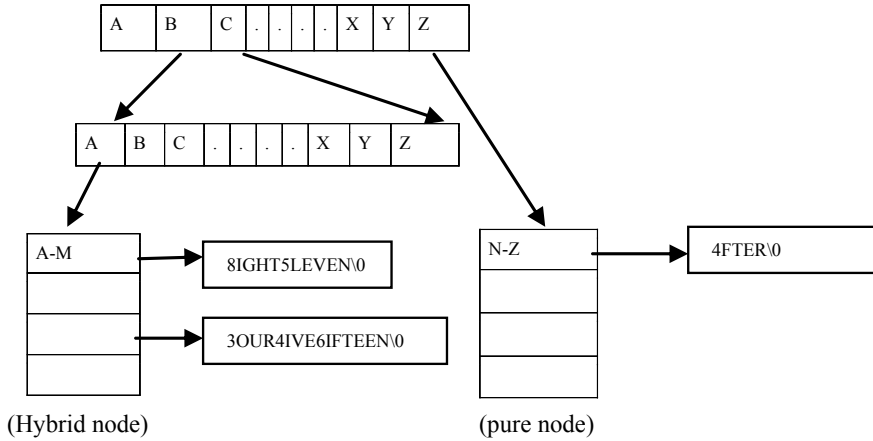


Fig. 4 HAT-trie with hybrid HAT-trie node representation for values 8, 11, 4, 15, 6, 7 and water

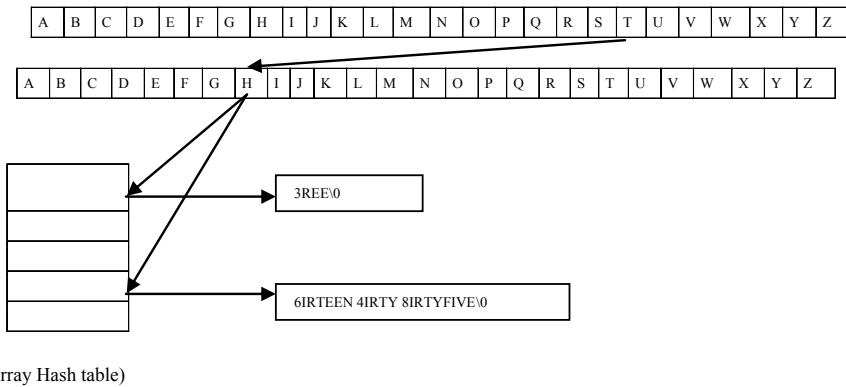


Fig. 5 HAT-trie with pure HAT-trie node representation for values of 3, 13, 30 and 35

and inspected in a top-down fashion. Initially it starts as a single unfilled hybrid bucket which is occupied until full. Buckets wont record duplicate values hence when a bucket is full, it is bursted based on the type of HAT-trie. After bursting, a new parent trie along with pointers are created where pure bucket is converted into a hybrid. The previous trie is moved as grandparent and the bursting procedure continues as a hybrid. To burst into a hybrid HAT-trie, a suitable split-point is chosen which achieves even distribution and is not applicable for all cases. String can be accessed by accumulating the occurrence of every leading character whose reoccurrence are then traversed in a linguist order to determine the number of movements of string. Once obtained, the split-point is elected as current occurrence counter alphabet. Based on split-point buckets are classified as pure or hybrid. Buckets of empty type are deleted and their corresponding parent pointers are set to empty.

6 Results and Discussion

In Bitcoin, merkle root is composed by applying SHA256 as binate for each and every transactions which is depicted in Fig. 6. Merkle root is constructed based on cryptographic hashes which is already discussed in Bitcoin Merkle. The planned system construct the merkle root based on HAT-trie which is both memory and speed improvement cache keen data structure. This uses base 64 encoding for preventing data loss across communication. This uses bit-wise hash to reduce the memory for storing and retrieval which is depicted in Fig. 7.

The proposed merkle tree serve buckets as an array of $n + 1$ word-length pointers, which are vacant or point to their respective slot entries. A dynamic array or bucket is used for storing base-64 encoded transactions. Here Base-64 encoding [16] is chosen for encoding the transaction because it can map all type of characters like ASCII, UTF8, UTF16 and other types and this can be transported to any systems in the network without data loss or modification of the content. The required slot is found by using the combination of base-64 encoding and fast bit-wise hash function [17] for performing insertion or searching of a transaction on a particular bucket. This is implemented based on shift-add-xor string hashing functions which are resistant of adversarial attack and the merkle root is determined based on the first encoded character. In some cases, space is wasted due to few string insertions for this; we implement the bucket as array hash table which makes changes from simple array when it is overflowing.

The projected system is compared with the speed of construction, searching, and memory requirements of Bitcoin Merkle and the conclusion is portrayed in Table 1. Our measurements of various factors indicate that the memory size is less and the searching speed is fast in Cognizant Merkle compared to Bitcoin Merkle. The searching speed of transaction in Bitcoin Merkle is very slow compared to Cognizant Merkle. Our dataset consists of a finite number (20 and 40 data sets) of null terminated strings that are unsorted. The above table shows for 20 inputs the memory size required by Bitcoin Merkle is approximately 6.33 MB and for Cognizant Merkle it requires only 3.33 MB and the searching time of Bitcoin Merkle is 0.0024 s whereas Cognizant Merkle requires only 0.0001 s. From this outcome, our research

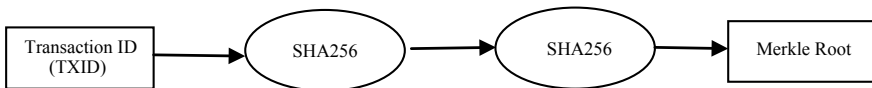


Fig. 6 Bitcoin Merkle Root

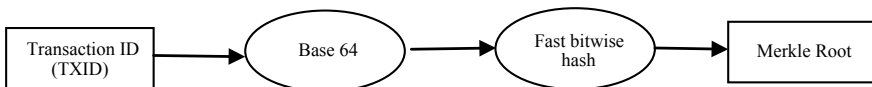


Fig. 7 Cognizant Merkle Root

Table 1 Comparison of Bitcoin Merkle with Cognizant Merkle

Factor/Tree type	Bitcoin Merkle		Cognizant Merkle	
Total inputs	20	40	20	40
Virtual memory size (MB)	6.33242	6.33242	3.33	3.33
Insertion time (s)	0.000206	0.000403	0.000645	0.001078
Searching time (s)	0.00248	0.000865	0.000179	0.000230

shows that the development of merkle tree like this structure can reduce the cost of searching time which yields rapid validation of transaction existence.

7 Conclusion

Currently, Bitcoin Merkle is a widely used data structure in blockchain for storing and verifying the transaction survival in the database. Still, memory requirement of Bitcoin Merkle is huge compared to the Cognizant Merkle and the depth of traversing the tree for searching of the transaction existence is high in Bitcoin Merkle compared to Cognizant Merkle model. The existing system of bitcoin is not affable for memory management and genial support for fast verification of transaction existence. Consequently the Cognizant Merkle, a variant of merkle tree that modifies the existence property of merkle tree data structure to yield, fast, adaptable, cache sensible and compressed data structure that maintains transaction id in memory which is stored in sorted order. The proposed new model for Cognizant Merkle tree is improved than 60% of existing Merkle tree searching. However, the proposed one provide appropriate speed and space efficiency of hash tables which can be further improved using cryptographic hashing without loss of transmission data across the peer-to-peer network which also prevent the mishandling of data in future occurrence of bitcoin transactions.

References

1. Vasek, M. (2015). The age of cryptocurrency. *Science*, 6241, 1308–1309.
2. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Journal of Cryptology*.
3. Vijayalakshmi, J., & Murugan, A. (2017). Crypto coin overview of basic transaction. *International Journal of Applied Research on Information Technology and Computing*, 2, 113–120.
4. Antonopoulos, A. M. (2014). *Mastering Bitcoin: Unlocking digital cryptocurrencies*. O’Reilly Media, Inc.
5. Reid, F., & Harrigan, M. (2013). An analysis of anonymity in the bitcoin system. In *Security and privacy in social networks* (pp. 197–223). Berlin: Springer.
6. Oliveira S., Soares, F., Flach, G., Johann, M., & Reis, R. (2012). Building a bitcoin miner on an FPGA. In *South Symposium on Microelectronics*.

7. Decker, C., & Wattenhofer, R. (2013). Information propagation in the bitcoin network. In *IEEE Thirteenth International Conference Peer-to-Peer Computing (P2P)* (pp. 1–10).
8. Investopedia Academy, <https://www.investopedia.com/terms/m/Merkle-tree.asp>.
9. Merkle, R. C. (1987). A digital signature based on a conventional encryption function. In *Conference on the theory and application of cryptographic techniques* (pp. 369–378). Berlin: Springer.
10. Hanandeh, F., Alsmadi, I., & Kwafha, M. M. (2014). Evaluating alternative structures for prefix trees. In *Proceedings of the World Congress on Engineering and Computer Science* (Vol. 1).
11. Heinz, S., Justin, Z., & Williams, H. E. (2002). Burst-tries: a fast, efficient data structure for string keys. *ACM Transactions on Information Systems (TOIS)*, 2, 192–223.
12. Bazoobandi, H. R., de Rooij, S., Urbani, J., & Bal, H. (2015). A compact in-memory dictionary for RDF data. In *European Semantic Web Conference* (pp. 205–220). Berlin: Springer.
13. Mavlyutov, R., Marcin, W., & Cudre-Mauroux, P. (2015). A comparison of data structures to manage URIs on the web of data. In *European Semantic Web Conference* (pp. 137–151). Berlin: Springer.
14. Knecht, M., Meier, W., & Nicola, C. U. (2014). A space-and time-efficient implementation of the Merkle Tree Traversal Algorithm. *arXiv, Vol. No. 1409.4081*.
15. Askitis, N., & Sinha, R. (2007). HAT-trie: a cache-conscious trie-based data structure for strings. In *Proceedings of the thirtieth Australasian conference on Computer science* (Vol. 62, pp. 97–105).
16. Stack Overflow Questions <https://stackoverflow.com/questions/4070693/what-is-the-purpose-of-base-64-encoding-and-why-it-used-in-http-basic-authentic>.
17. Ramakrishna, M. V., & Zobel, J. (1997). Performance in practice of string hashing functions. In *DASFAA* (pp. 215–224).

A Novel Algorithm for DNA Sequence Compression



K. Punitha and A. Murugan

Abstract Deoxyribonucleic Acid (DNA) sequences vary in terms of their size and fall within the range of billions of nucleotides. The value increases to twice or thrice its original value annually. Techniques of data compression and related methods that originate from the information theory are frequently understood as relevant for the field of data communication, exploration, and storage. In the present situation, it is vital to store data for biological sequences. The compressBest algorithm proposed in the paper for the compression of DNA sequences helps attain a better compression ratio and is much faster when compared to the existing compression techniques. compressBest algorithm is applicable to compression of DNA sequences with a reduction in storage space. The proposed algorithm is tested over the data from the UCI repository.

Keywords DNA sequences · Data compression · Deoxyribonucleic acid · Dynamic programming

1 Introduction

DNA databases are found and large in size [1, 2], where storage of them is a major issue. They contain considerable complexity and feature logical organization. Therefore, the data structure for storing, accessing, and processing this data in an efficient manner is a challenging and problematic task [3, 4]. So, an efficient algorithm for compression is required for storing huge data masses. The standard methods for compression [5, 6] are not relevant when it comes to biological sequences [7] owing to the subtle regularities in sequences of DNA [8, 9]. Biological sequences cannot

K. Punitha (✉)

Department of Computer Science, Agurchand Manmull Jain College, Chennai, Tamil Nadu, India
e-mail: punithsathish@gmail.com

A. Murugan

Department of Computer Science, Dr. Ambedkar Govt. Arts College (Autonomous), Chennai, Tamil Nadu, India
e-mail: amurugan1972@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,

https://doi.org/10.1007/978-981-13-5953-8_13

be compressed in a competent manner by standard algorithms used for compression. The compressBest algorithm for DNA is introduced which is based on the precise matching and gives the best results for comparison of the standard benchmarks in the context of DNA sequences.

The four nucleotide bases are Cytosine, Adenine, Thymine, and Guanine, depicted using their first character, which is *C*, *A*, *T*, and *G*. The meaning of compression is the reduction of data size by way of modifying it to assume a format which necessitates less number of bits than the ones required in the original format. Considerable effort has been expended for applying compression techniques on textual data for the accomplishment of a number of tasks in computational biology, from storing and indexing large sets of data to comparing sequences databases of DNA.

This paper proposed compressBest algorithm which can be employed for the compression of DNA sequence data, which in turn results in a reduction of storage space through the use of Dynamic programming. The mechanism which is proposed for the compression of the DNA sequence through the use of the 2-bits encoding method includes division into segments, exact matching, and the method of decompression matching. When bases are distributed in a random manner in a sequence, 2-bit encoding method serves as an efficient method. However, nonrandomness exists in an organism's life and the DNA sequences which are evident in the living organism are nonrandom in nature. They also possess certain constraints. Our algorithm resembles all other compression algorithms of the DNA sequence and is based on creating partitions for a sequence into diverse segments, which include the repeat segments which are also the copied segments, and the non-repeat segments.

2 Related Work

In [5], a two-stage algorithm suggested the combination of the features of the Huffman coding algorithm and Lempel–Ziv–Welch (LZW) algorithm. LZW is an algorithm which is based on a dictionary. It is a lossless algorithm and is unified into the standard pertaining to the consultative committee on telephony and international telegraphy [10]. In this case, the code for every character is accessible in the dictionary [11] which uses reduced bits when compared to the ASCII code (less than 5 bits).

A novel approach is proposed in [12], the compression of genomes which has the ability to modify successions in the genome to double grouping which uses the Genbit algorithm and the development of a grid takes place from the arrangement which has been encoded. In case the encoding group length is not a perfect square, a few bits are attached to the arrangement to create a flawless square with respect to its length. When both sides record the same number of times, a '0' is picked and the bit is connected again at the end to make the length a flawless square for the development of a framework.

Many algorithms have been recently introduced for the purpose which frequently uses the recognition of lengthy and approximate repeats. Another algorithm is known

as the DNAPack, which use dynamic programming as its basis. When compared to the earlier programs, the DNA compression is marginally improved while its cost can be ignored. A new lossless comparison algorithm is presented which enables the vertical and horizontal compression of the data. It is based on the statistical and substitution methods [13]. The representation of hiding data through the use of compression of DNA sequences is attempted. In the initial step, data is converted to a DNA sequence and the DNA sequence is compressed later on. The techniques of four compressions are used in this case, where better compression is produced by one of the options, which depends on the DNA sequence. The decompression algorithm which is developed also enables the procurement of the original data [14].

A lossless compression algorithm which is seed-based is developed in the paper to compress the DNA sequence which utilizes the method of substitution. This method has a similarity to the LempelZiv scheme of compression. The repetition of structures is exploited in the method proposed here, which are intrinsic to DNA sequences. The process is accomplished through the creation of an offline dictionary containing repeats accompanied by mismatch details. When it is ensured that only promising mismatches are allowed [15], a compression ratio is achieved by the method that is better than or in line with the existing compression algorithms pertaining to lossless DNA compression [16].

A DNA compression algorithm is introduced [17], which originated on the basis of exact reverse matching which allows best results for compression for the benchmarks of standard DNA sequences. When the DNA sequence is enormously long, it is easy to search for exact reverses. The approximate reverses which are optimal for the compression to take place can be found by the algorithm, but the task takes considerable amount of time to complete. The time taken is a quadratic time search or greater than that [18, 19]. Compression with high speed and the best ratio can be achieved, but it is challenging to do so. The compression for proposed DNA sequences achieves a better ratio for compression and it is running faster when compared to the existing algorithms [20] for compression.

3 Proposed Work

This is a form of motivator for the advancement of compression tools with high performance whose design is targeted at genomic data. The DNA compression methods used earlier were either statistical or dictionary-based in nature. The 2-bit encoding methods in use recently have become noticeable where the bases with 4-nucleotides in the DNA sequence $\{A, G, C, T\}$ are assigned the values 00, 10, 01, and 11 respectively. The technique proposed here is used for the compression of large DNA sequence bytes which have average ratio for compression. The decompression time initially needs an input which is available at the time of sequence compression. This value is used along with the compression input strings where the

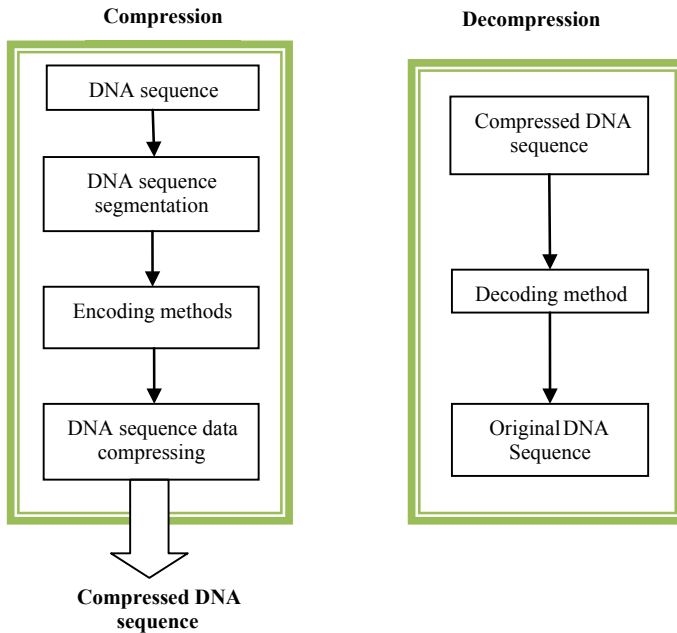


Fig. 1 Overall architecture of the proposed method

decompression takes place and the output is produced in the form of the original sequence. The decompressed input is obtained from the original sequence as per the segmented values. The DNA sequence algorithm for compression attains a compression ratio which is moderate and runs considerably faster when compared to present compression programs. The overall process of the proposed method is given in Fig. 1.

3.1 Segmented in Dynamic Programming Techniques

Different symbols are in use with respect to the occurrences of different DNA sequences which are chosen for the compression of the sequence. This means that the DNA sequence data is divided into a number of parts throughout the process. A technique used for solving the optimization of a sequence of decisions is the dynamic programming technique. The underlying rationale behind this is the representation of the problem by the way of a process which is an evolution from one state to another in reaction to specific decisions.

3.2 *Encoding of Numbers*

The proposed algorithm needs the encoding of diverse integers. Let, $A = 00$, $C = 01$, $G = 10$, $T = 11$ is the segmentation which is accompanied by a segmented number. The process initiates with the alignment of each sequence in the dataset with respect to the reference sequence through the use of the local sequence alignment. The objective behind sequence alignment is the placement of homogenous segments in the same column through the insertion of blank space. Further, aligning similar sequences can assist in the discovery of patterns and sequence relationships, which can improve the ratio for compression.

As an example, the segments are not fixed in length, and the encoding of their length is important. For a repeated segment, the values which are segmented with respect to the reference substring of the input are taken into consideration, and the segment is copied from here, after which encoding takes place. In the case of approximate copies, rather than exact ones, it is important to encode the value which has been segmented with respect to the modifications. Any of these numbers do not feature any bounds, and the integers must be encoded in a way which is self-delimited as opposed to encoding with respect to a fixed number of bits. The reference segmented values are encoded by encoding the relative difference with respect to the reference segmentation and it is preferred to make use of its copy.

The encoding method which involves 2-bits can make use of 2 bits for the encoding of every character which means that 00 is for A, 10 is for G, 11 is for T, and 01 is for C. Therefore, "gaccgtca" can be encoded by "10 00 01 01 10 11 01 00". This requires 16 bits in all. The exact matching method can make use of repeat length and repeat position for the representation of an exact repeat. In this way, three bits are used to encode an integer and two bits are used to encode a character, and a single bit is used to indicate whether the next part contains a pair, which also signifies an exact repeat or plain character.

3.3 *Compression*

In experimental research work which involves the text file with a dot txt file extension, a series of four successive base pairs are present, which are a , g , t , and c . This ends in a blank space as a terminal character. The basic element, in this case, is the text file which is used to consider compression as well as decompression. The output file is also a text file containing information about four base pairs which are unmatched and an ASCII character which has a coded value.

3.4 *CompressBest Algorithm*

Compression Algorithm

Input: DNA Sequence

Output: Compressed data

Input: Compressed text file

Output: original DNA Sequence

- 1: Repeat the following steps 2 to 3 until the codes are converted into bases.
 - 2: According to the segmentation number, replace all the codes by the corresponding bases.
 - 3: do
 - {
 - Read the character one by one from the DNA sequence;
 - If char = \$ followed by the 2-bit based sequence base then
 - Assign 3 times the base
 - If char = # followed by the 2-bit based sequence base then
 - Assign 4 times the base
 - Else
 - Read two characters and
 - Assign A if char = 00;
 - Assign C if char = 01;
 - Assign G if char = 10;
 - Assign T if char = 11;
 - } until char = NULL;
 - 4: The original DNA sequence is obtained.
 - 5: End
-

3.5 *Decompression*

The input string used for compression of a particular value is decompressed to produce the original file in the form of the output. The lookup table is created from the structure of the compressed file by finding repeat patterns present in the source. The size of the lookup table is extracted from bit blocks which represent the pattern. The blocks of the compressed pattern are extracted and the pattern type is recollected in the form of pattern ID followed by patterns. Decompression takes place from the beginning of the file to the end of the file for obtaining the original sequence of DNA.

Decompression Algorithm

Input: DNA Sequence
 Output: Compressed data

- 1: Divide the sequence into segments with four nucleotides each in DNA sequences.
 - 2: Find the matched segments in the DNA sequence and assign the numbers with symbol 'S' followed by corresponding numbers.
 - 3: Repeat the following steps for each in the segment.
 - (i). If there is three repeat bases, then assign 'S' as a suffix and assign two bits based on DNA sequence values (A=00, C=01, G=10, T=11).
 - (ii). If there is four repeat bases then assign '#' as a suffix and assign two bites based on DNA sequence values (A=00,C=01,G=10,T=11).
 - 4: Repeat the steps from 2 to 3 until there are no repeat bases.
 - 5: Assign two bits code for remaining bases.
 - 6: End
-

The compression ratio can be calculated as follows:

$$\text{Compression ratio: } \frac{\text{compressed file size}}{\text{original file size}}$$

4 Results and Discussions

The program code of the compressBest algorithm is executed Java 1.7.0 JDK and tested on a dataset of DNA sequences which is from Genbank through UCI repository. Genbank is a popular public database where the human genomic data is available. The datasets include six DNA sequences such as two are HUMAN GENES [HUMDYS-TROP, HUMHDABCD], two from chloroplast genomes (CHMPXX AND CHN-TXX), and two are Viruses (HEHCMVCG AND VACCG).

The storage of the DNA sequence and its accuracy must be considered since even a single mutation in the base, deletion, or insertion would reflect in terms of a major change in its phenotype. For the purpose of accuracy, file compression is developed one after the other through the use of the compressBest algorithm.

The efficient result level in terms of the performance analysis of the compression ratio percentage of DNA sequence data is represented in Fig. 2 along with other techniques such as Lossless compression algorithm and BDNAS compression algorithm. The compressBest algorithm proposed here produces efficient output when compared to the other techniques in existence.

The compression of the original sequence size before compression is depicted in Table 1, and compression ratio after compression is also depicted in Table 1. The advantages of the compressBest algorithm are, it is a high-speed algorithm with minimized compression ratio, minimized execution time and it gives lossless compression data.

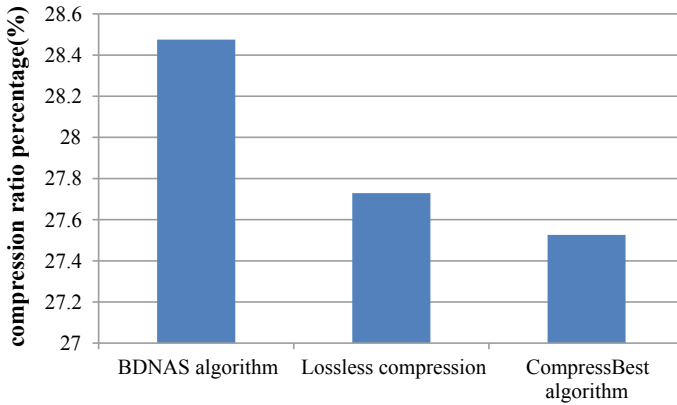


Fig. 2 Comparison of compression techniques

Table 1 Experimental analysis

S. No.	Type of DNA sequences	Original size of sequence before compression (Bytes)	Compression ratio percentage (%) after compression		
			BDNAS algorithm	Lossless compression	CompressBest algorithm
1	HUMHDABCD (Human Gene)	58,864	33.33	32.61	32.53
2	HUMDYSTROP (Human Gene)	105,265	33.33	32.62	32.59
3	CHMPXX (Chloroplast Genome)	121,024	4.18	4.02	3.66
4	CHNTXX (Chloroplast Genome)	155,844	33.33	32.42	32.35
5	HEHCMVCG (Virus Genome)	229,354	33.33	32.58	32.14
6	VACCG (Virus Genome)	47,912	33.33	32.08	31.86

5 Conclusion

The experimental result shows that CompressBest algorithm gives better compression ratio for mostly repeated sequences. It provides a reduction in file size without losses of clear data. The results from the simulator help to achieve a minimization in the time required for compression, high speed, efficiency, reduction in file size, and accuracy with respect to the original file. The UCI repository was accessed to collect the datasets and the Java environment is used for the development.

References

1. International nucleotide sequence database collaboration, <http://www.insdc.org>. (2013).
2. Karsch-Mizrachi, I., Nakamura, Y., & Cochran, G. (2012). The international nucleotide sequence database collaboration. *Nucleic Acids Research*, 40(1), 33–37.
3. Deorowicz, S., & Grabowski, S. (2011). Robust relative compression of genomes with random access. *Bioinformatics*, 27(21), 2979–2986.
4. Brooksbank, C., Cameron, G., & Thornton, J. (2010). The European Bioinformatics Institute's data resources. *Nucleic Acids Research*, 38, 17–25.
5. Shumway, M., Cochran, G., & Sugawara, H. (2010). Archiving next generation sequencing data. *Nucleic Acids Research*, 38, 870–871.
6. Kapushesky, M., Emam, I., & Holloway, E. (2010). Gene expression atlas at the European bioinformatics institute. *Nucleic Acids Research*, 38(1), 690–698.
7. Ahmed, A., Hisham, G., & Moustafa, G. (2010). EGEPT: Monitoring Middle East genomic data. In: *Proceedings of 5th Cairo International Biomedical Engineering Conference* (pp. 133–137). Egypt.
8. Korodi, G., Tabus, I., & Rissanen, J. (2007). DNA sequence compression based on the normalized maximum likelihood model. *Signal Processing Magazine, IEEE*, 24(1), 47–53.
9. Deepak, H. (2013). State of the art: DNA compression algorithms. *International Journal of Advanced Research in Computer Science and Software Engineering*, 397–400.
10. Kaur, S., & Verma, V. S. (2012). Design and Implementation of Lzw data compression algorithm. *International Journal of Information Sciences and Techniques (IJIST)*, 2(4). <https://doi.org/10.5121/ijist.2012.240771>.
11. Shrivani Kulkarni, S., & Kini, Y. (2017). Pre equal architecture for lossless data compression and decompression using hybrid algorithm. *International Journal of Advance Electrical and Electronics Engineering (IJAEEL)*, 6(1_2). ISSN (Print): 2278-8948
12. Bhukya, R., Viswanath, B. V., Mahendra Kumar, D., Swathi Kiran, D. S., & Bagdia, P. (2017). DNA sequence decompression using bitmap matrix & wavelet transformation in image processing. Received: 20th April 2017, Accepted: 28th April 2017, Published: 1st May 2017, Copyright © 2016 Helix ISSN 2319– 5592 (Online)- Helix Vol. 8: 1491–1497.
13. Mishra, K. N., & Aagarwal, A. (2010). An efficient horizontal and vertical method for online DNA sequence compression. *International Journal of Computer Applications*, 3(1), 0975–8887.
14. Bandyopadhyaya, S. K., & Chakraborty, S. (2011). Data hiding using DNA sequence compression. *Journal of Global Research in Computer Science*, 2 (1), Vol. 27–33. © JGRCS 2010, All Rights Reserved.
15. Cover, T. M., & Thomas, J. A. (2012). *Elements of information theory*. New York: Wiley.
16. Eric, P. V., Gopalakrishnan, G., & Karunakaran, M. (2015). An optimal seed based compression algorithm for DNA sequences. Correspondence should be addressed to Pamela Vinitha Eric; pamela.vinitha@gmail.com Received 28 November 2015; Revised 9 May 2016; Accepted 19 June 2016.
17. Mukherjee, R., Mandal, S., & Mandal, B. (2016). Reverse sequencing based genome sequence using lossless compression algorithm. *International Research Journal of Engineering and Technology (IRJET)*, 3(5). e-ISSN: 2395-0056. www.irjet.net p-ISSN: 2395-0072.
18. Jacob, G., & Murugan, A. (2013). DNA based cryptography an overview & analysis. *International Journal of emerging science*, 3(1), 36–42.
19. Chouhan, D. S., & Mahajan, R. P. An architectural framework for encryption and generation of digital signature using DNA cryptography. In *International Conference on Computing for Sustainable Global Development (INDIACom)*.
20. Jahaan, A., Ravi, T. N., & Panneer Arokiaraj, S. (2017). Bit DNA squeezer (BDNAS): A unique techniques for data compression. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* ©, 2 (4). ISSN: 2456-3307.

DigiPen: An Intelligent Pen Using Accelerometer for Character Recognition



Ankur Agarwal and Swaroop Sudhanva Belur

Abstract The idea is to identify the character being written by a user on a piece of paper by observing the acceleration of the pen while writing these characters and passing this acceleration data to an Artificial Intelligence (AI) algorithm. Using an off-the-shelf accelerometer paired with a microcontroller using Inter-Integrated Circuit (i2c) protocol, we get sequential acceleration values in three directions. Upon observing this with this data, we noticed the same patterns when writing a specific character and decided to use efficient Long Short-Term Memory (LSTM) cells to recognize the patterns. After collecting some data for training the neural network and doing some preprocessing of the data, we used Basic LSTM cells in the models to recognize the patterns from the sequences of these values. LSTM cells were preferred over regular Recurrent Neural Network cells (RNN) due to LSTMs ability to remember longer sequences. Multiple LSTM models with a different number of layers and sizes with different activation functions and dropout values were trained and tested for performance and we were able to achieve a test accuracy of 47% on a fairly small dataset which far exceeds the 10% accuracy benchmark which would have been simple guesswork. With some more optimization of the hyperparameters of the neural network and training with a larger dataset, we believe better performance can be achieved. For the purpose of this paper, we have used numerical digits (0–9) as the characters to be classified.

Keywords Smart pen · Artificial intelligence · LSTM · IoT · Cloud · Supervised learning

A. Agarwal (✉)

Department of CSE, SRM Institute of Science and Technology, Kattankulathur
603203, Tamil Nadu, India
e-mail: ankuragarwal_ma@srmuniv.edu.in

S. S. Belur

Department of ECE, SRM Institute of Science and Technology, Kattankulathur
603203, Tamil Nadu, India

© Springer Nature Singapore Pte Ltd. 2019

N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_14

1 Introduction

Despite the technological advances in the twenty-first century, the act of taking notes is still relevant. This explains the plethora of note-taking applications on phones and computers. Current techniques of digital note-taking are sub-optimal for reasons which include but are not limited to—small touch keyboard interfaces, heavy and uncomfortable devices, etc. We want to bring back the usage of notebooks and thus conceived of a device that uses twenty-first century technology but has the feel of a regular notebook [1]. This is how we arrived at the DigiPen. When trained with a proper amount of data, this pen was able to accurately detect the characters being written without the need for any other special type of book or pen. Using the analytical power of AI [2], it can even understand characters of different alphabets and symbols. The days are not far off when professionals like doctors writing prescriptions and students writing notes can refer to and search within the digital copy instead of using their regular books [3]. This brings out the real power of the digital revolution to something as banal as note-taking.

2 Similar Technologies

Although devices that have a similar functionality exist, none of them use the same technology as DigiPen and usually require extra peripherals like a special book or the use of a mobile or a computer unlike DigiPen. Multiple products have come up with a pen that uses a dot pattern on an accompanying book which is used to recognize the characters being written using the location of the pen's tip to trace out what has been written. Moreover, these products require a mobile phone to work. This is innovative technology but the problem is that it requires the purchasing of a new book every time it runs out of pages. However, the advantage these products have over the DigiPen is that they are much easier to implement due to the availability of datasets like MNIST. In case of the DigiPen, it requires a completely different type of dataset which uses accelerometer values. Due to the same reason, DigiPen has a relatively lower accuracy than the others. However, with the addition of more data into the training dataset, comparable or even better accuracy can be achieved by DigiPen.

3 The Analogy and Inside the Accelerometer

The accelerometer [4] contains a surface machined polysilicon structure which is positioned on top of a silicon wafer. The polysilicon structure is suspended using springs and provides resistance against acceleration forces. Differential capacitors, consisting of independent fixed plates, are used to measure deflection. The deflection

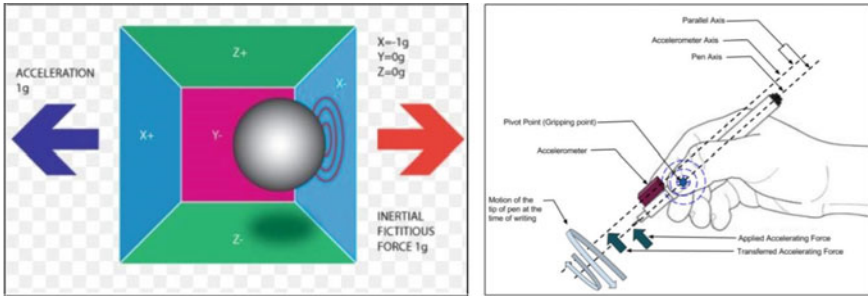


Fig. 1 (a) The accelerometer, (b) The pen [5]

of the beam causes an unbalanced capacitor output which, in turn, results in a sensor output whose amplitude is proportional to the acceleration. Phase-sensitive demodulation is then applied to determine the polarity and magnitude of the acceleration.

Simply put, there is a polysilicon ball placed inside a hollow cube whose surfaces correspond to the differential capacitors. The movement of the ball, due to accelerative forces, causes its movement inside the cube, which results in an unbalanced capacitor output.

We use these acceleration values as our parameter for recognizing the written character. Since the accelerometer is placed on the flat surface of the pen, both the pen and accelerometer become parallel to each other. Now as they are parallel and connected, the accelerating forces applied on the pen while writing are also applied on the accelerometer. The lines of accelerating forces on both the objects at the point of connections have the same magnitude and direction. Due to this reason, the accelerometer is able to capture the true forces of accelerations acting on the pen. As the gripping point is the pivot point during writing, the accelerometer is attached at a location from where maximum motion can be transferred on to the sensor effectively (Fig. 1a).

4 Data Collection

The contraption that was built is an accelerometer attached near the writing tip of a regular pen and the accelerometer data is collected using an microcontroller using the inter-integrated circuit (i2c) communication protocol. The arduino microcontroller is used in conjunction with an open source software called Processing [6] to take the serially printed data into a comma separated variable (CSV) file format. We obtain x , y , and z coordinate accelerometer data separately. These constitute the three comma separated variables which are fed into a neural network. The Microcontroller used is an ATmega 328P which is programmed and used by means of the Arduino Uno development board which has general purpose input output (GPIO) pins, an oscillator crystal, a USB connection, support for i2c and SPI (serial peripheral interface)

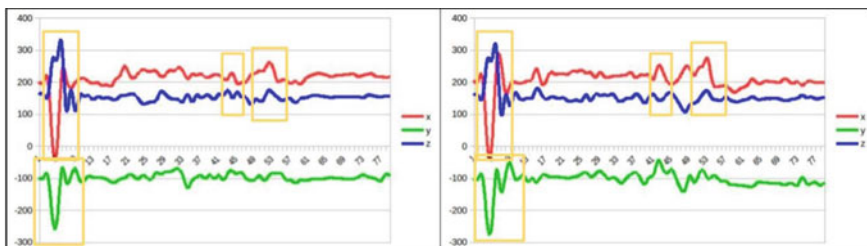


Fig. 2 Similar acceleration patterns for a given character written

communication protocols and has an accompanying integrated development environment for programming and flashing. The Arduino is programmed to collect data for a short amount of time which we believe is sufficiently long enough to write a character. After some experimentation, the time we arrived at was 1200 ms. When the user wants to write a character, he/she has to just double tap the pen which will trigger the accelerometer and it will start collecting data for a period of 1200 ms (Fig. 2).

Due to resource constraints, we were only able to collect a limited amount of data to train the LSTM. We have 200 data points for each numerical digit from 0 through 9. This adds up to 2000 data points in total. Although this is not ideal for the proposed LSTM model, we observed that the model does fairly well as discussed above and far exceeds the 10% accuracy benchmark which would have been simple guesswork by the LSTM. While collecting data, we tried to ensure that the size of the character written is small and about the same size as that written in a regular notebook. It is important because first, that is how we believe people will write with the digipen, but more importantly, the acceleration forces when writing small are more pronounced and are easier to patternize. Moreover, as mentioned, we used a constant amount of time of 1200 ms to collect data for each datapoint.

Another aspect we considered while collecting the data is the orientation in which the data was collected. We tried to take the readings in various different orientations to ensure that DigiPen can predict whatever has been written even if it was held in a different position. The dataset consists of comma separated variables (CSV) files for each datapoint. Each data point consists of x , y , z followed by the data collected by the accelerometer for 1200 ms corresponding to each of the three axes. Due to small, insuppressible errors in the sensor, the number of values corresponding to x , y , and z axes are slightly variable despite taking to same amount of time to record. We found that they range from 75 to 85. To standardize the data going into the LSTM, we employ zero-padding. This ensures that the LSTM receives a uniform number of values for each datapoint. We used an 80:20 split for training and test data for the LSTM.

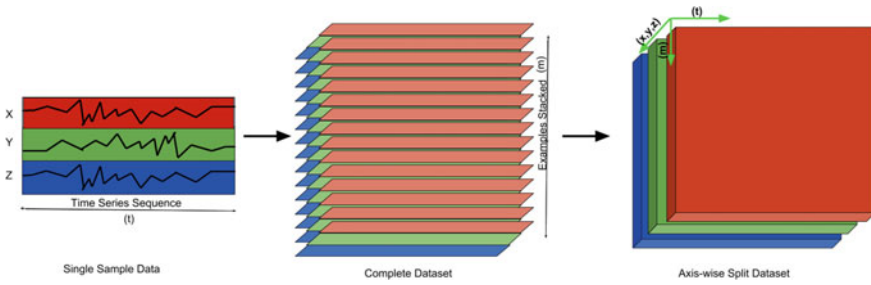


Fig. 3 Preparing the dataset

5 Preparing the Dataset

The input data is a sequence of acceleration values in the x , y , and z axes. Each sampling is completed within t secs and each sample recording gives three column time series sequences of acceleration in each axis direction. Hence, each sample recording is of shape $(t, 3)$. Each sample recording is saved as comma separated value sequence. Initially while creating the model input dataset, each file is read as a single matrix having 3 columns (x, y, z) & ' t ' rows and the complete dataset becomes a 3D matrix in which many such 2D matrices are stacked. The label y is taken from the filename, i.e., the digit it stores. The label is further encoded into one hot encoding of 10 classes, i.e., for digits between 0 and 9. The dataset is normalized and shuffled randomly. The dataset is split into training and test dataset in the ratio 80:20. The dataset is split into three 2D matrices each representing the sequential values of the samples in different axis directions. This data is given as input to the neural network for further classification model (Fig. 3).

6 The Different Models

The input values to the neural network model are time series based. When writing a digit there is a particular pattern which cannot be skipped no matter which font is used as the digit is predefined. Example, the digit 1 will always have a straight line, the different fonts will always have this straight line as the major part. The accelerometer catches this pattern in terms of accelerations in different directions and the pattern of this series is important to the model. The output is to be given as a label from given classes of 10 digits. This clearly becomes a sequential and classification problem. The different models used have RNN layer with LSTM cells [7] common in all to find out the important sequential patterns for a longer time. The sequential patterns from each direction (i.e., x, y , and z) are important, and so LSTM cells are used. The following three different models have been designed and implemented (Fig. 4).

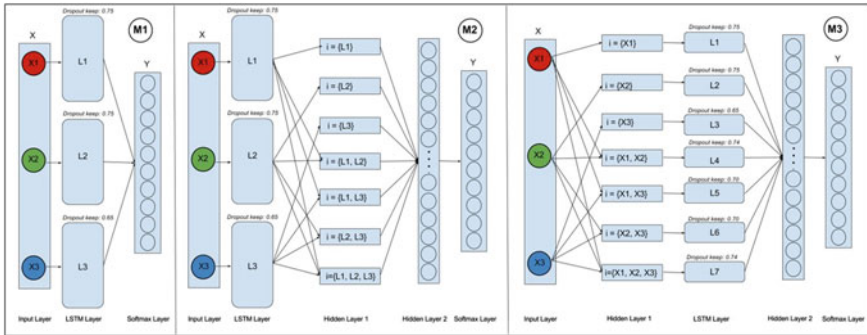


Fig. 4 The different model designs

6.1 Model M1

A basic simple model is created. The input of sequence of values in x , y , and z directions are directly given as input to the channelized LSTM cells. The outputs from the LSTM cells are sent to a softmax layer of 10 classes which gives the output of the model in the form of one hot encoding.

6.2 Model M2

Using the similar design as that of model M1, model M2 also has three LSTM cells channelized with their inputs as the input to the model. The next layer has seven neurons, all together having every possible input combination from the output of the three LSTM cells. This layer is added with the intuition that all possible combination of probabilities from the LSTM outputs will be generated, which might become valuable to help find out the classification. Each of the seven neuron units has the same size as that of the single output of a LSTM cell. The outputs from these neuron units are sent to a layer having 100 neuron units which is then connected to a reduced 10 neuron unit softmax layer. This softmax layer gives the output of the model in the form of one hot encodings.

6.3 Model M3

This model is similar to the previous model but with a more intuitive design. In this model, the sequential inputs x , y , and z are sent to a layer of seven neurons, all together having all possible input combinations of x , y , and z . Here, all the possible combination of the sequences in different directions are generated. The size of each

of these seven neuron unit is same as that of a single direction sequence input to the model. The output of these seven neuron units is sent to a layer having seven channels of LSTM cells. Passing all the possible combinations of sequences through LSTM cells might help find more hidden sequences. The output from these seven LSTM cells is sent to a layer having 100 neuron units which is then connected to a reduced 10 neuron unit softmax layer. This softmax layer gives the output of the model in one hot encoding form.

6.4 Hyperparameters, Regularization, Optimizer and Activation Function

In the above three models, each LSTM cell has 32 units as the dataset is small. Also in above three models, a dropout layer is attached at the end of each LSTM cell. The keep probability of LSTM dropout layer having z direction as an element is kept lower than others as the data samples where created with characters written on a sheet of paper (viz. two-dimensional) with very less variation in the z direction. In M1 and M2, dropout layers for LSTM cells L1 and L2 have a keep probability of 0.75 and for L3 it is 0.65. In model M3, LSTM cells L1 & L2 have keep probability of 0.75, L3 has 0.65, L4 and L7 have 0.74 and L5 and L6 have 0.70 as keep probability. Apart from the dropout layers, exponentially decaying learning rate is used to help the models converge gradually with a lower learning rate near the convergence point. ELU activation function was used to take care of the vanishing gradient problem. For optimizing the backpropagation cost of training, the Adam optimizer was used (Fig. 5).

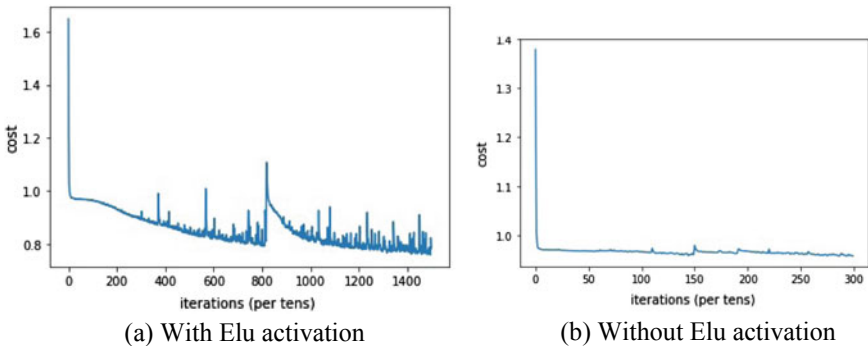


Fig. 5 Purpose of using Elu as activation

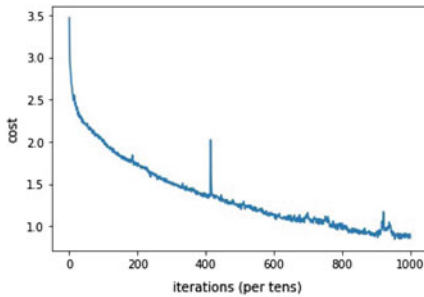
7 Performance

The models were trained after tuning the hyperparameters. A few hyperparameters were set with common values so as to judge the three models. The values were set keeping in mind the small size of the dataset available. The few common parameters and hyperparameters which were set with same values are

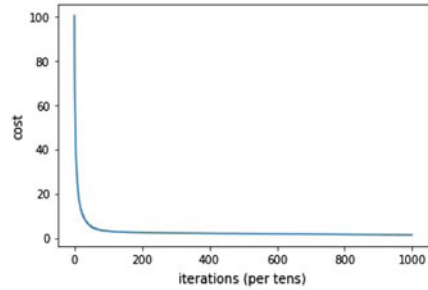
- learning rate = 0.001
- decay rate = 0.05
- decay step = 5
- mini-batch size = 96
- iterations = 1000
- number of units in each LSTM cells = 32
- common random seed (Table 1).

Table 1 The performance of the three models

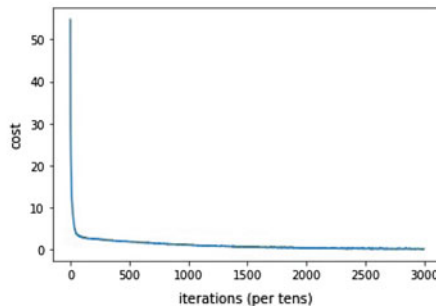
Model	Training accuracy	Test accuracy
M1	64.93	34.25
M2	56.35	32.75
M3	74.76	42.25



(a) Model 1



(b) Model 2



(c) Model 3

Fig. 6 Performance of the three models

The model M3 seems to work best compared to the others. The model was run for higher number of iterations to get better test accuracy results (Figs. 6 and 7; Table 2).

It is clearly visible from the confusion matrix that the resulting model has a good precision and recall as the frequency of the actual digits being predicted is far greater than the frequency of any other wrong digit being predicted.

8 Future Development

The future development from this concept is the addition of various features by integrating with various office suites using the internet of things (IoT). This will be

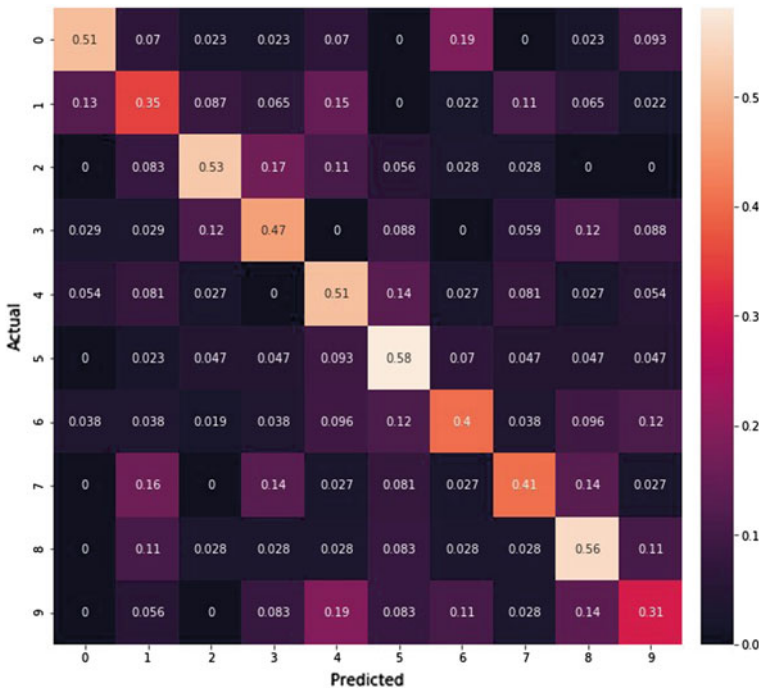


Fig. 7 Normalized confusion matrix (Test set)

Table 2 Performance of Model 3 with different iterations

Model	Iterations	Training accuracy	Test accuracy
M3	1000	74.76	42.25
M3	2500	96.43	43.00
M3	3000	96.64	47.00

especially handy when the user wants to use the device completely independent of a mobile phone or a computer. We could send raw sensor data, preferably encrypted, to the cloud where a trained LSTM network will be deployed. The output generated can be then be saved as a text document, which can then be shared. As mentioned, this results in DigiPen being able to be used as a standalone device.

9 Conclusion

We have found an innovative way to recognize characters from the fundamental basis of the method of writing, which is the movement of the pen and data collection is very simple. We have found how to process the data in order to classify them into characters using an intuitive architecture model and achieve a test accuracy of 47%.

Acknowledgements The authors would like to thank Next Tech Lab and SRM Institute of Science and Technology for their continued guidance and support. We would also like to thank Nvidia for their generous grant of an Nvidia Titan X which was vital for us to be able to test the various models.

References

1. Reynaerts, D., & Van Brussel, H. (1995). Design of an advanced computer writing tool. In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science* (pp. 229–234). MHS '95. Nagoya.
2. Wang, J. S., & Chuang, F. C. (2012). An accelerometer-based digital pen with a trajectory recognition algorithm for handwritten digit and gesture recognition. *IEEE Transactions on Industrial Electronics*, 59(7), 2998–3007.
3. Toyozumi, N., Takahashi, J., & Lopez, G. (2015). Development of pen based interface system for digital operation. In *2015 IEEE/SICE International Symposium on System Integration (SII)*, Nagoya (pp. 593–598).
4. Accelerometer working <https://www.sparkfun.com/datasheet/Sensors/accelerometer/ADXL345.pdf>.
5. Accelerometer Image (Fig. 1a) http://www.starlino.com/imu_guide.html.
6. Processing <https://processing.org/>.
7. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computing* 9(8), 1735–1780.

Design of FPGA-Based Radar and Beam Controller



Adesh Panwar and Neha Goyal

Abstract Radar controller is vital component in a Radar system. It enables transmission of RF energy, generation of complex waveform schemes and precision timing to gate the receiver. The paper presents architecture for FPGA-based Radar controller. It includes beam controller to schedule dwell requests among all tasks. Proposed design is robust, flexible, and uses VHDL-based modules to promote reuse and cost optimization. The paper depicts detailed hardware and software methodology adopted during the development.

Keywords FPGA · Controller · Phase gradient

1 Introduction

Radar system provides situation awareness to the operator. It assists in understanding the nature, action and intent of the “targets of interest”. It provides such information in a complex environment, in all weather conditions, day and night. In such systems, electromagnetic energy is transmitted in the form of a train of pulses and the reflected energy in its direction enables the detection and measurement of the target position. Such a system has an antenna which concentrates the transmitted energy in a preferred direction and also intercepts and captures the received echoes [1].

Traditionally, antennas are classified into two major categories, mechanically steered and electronically steered antenna. The electronic scanning uses a group of radiating elements wherein the relative phase of the respective signals is controlled in such a way that the effective radiation pattern of the array is reinforced in the desired direction [2]. Phased array Radar provides significant advantage such as

- Solid state technology
- Distributed architecture
- Modular design

A. Panwar (✉) · N. Goyal
Bharat Electronics Limited, Bangalore 560013, India
e-mail: adeshpanwar@bel.co.in

- Beam agility
- Improved reliability.

This paper presents design of FPGA-based Radar and beam controller used in the phased array. Section 2 introduces a typical architecture of the Radar system. The proposed system is presented in Sect. 3. Section 4 gives details on features incorporated in the design. Sections 5 and 6 gives test results and conclusion respectively.

2 Background

The system block diagram is shown in Fig. 1. Operator selects the desired volume to scan and communicates it to the Radar and Beam Controller (RBC) via Operator Console (OC). RBC sets the timing parameters and provides the required waveforms. It plans ahead the future beam position and communicates the phase values to Phase Gradient Controller (PGC) unit.

PGC in synchronization with RBC sets Transmit Receive (TR) modules to transmit in a specific direction upon which target-acquiring microwave beams are transmitted from the antenna, which also collects the reflection. The echo radio frequency (RF) signal is multiplied with the local oscillator (LO) signal and translated to an intermediate frequency (IF).

The received signal is digitized and discrete values are provided to the FPGA chip for digital pulse compression processing. It de-chirps and digital orthogonal transformation is performed to form in-phase and quadrature signals. Each such pair of samples is associated with a “numerical index” that corresponds to a specific time offset from the beginning of the Radar pulse, so each of them represents the energy reflected from a specific range.

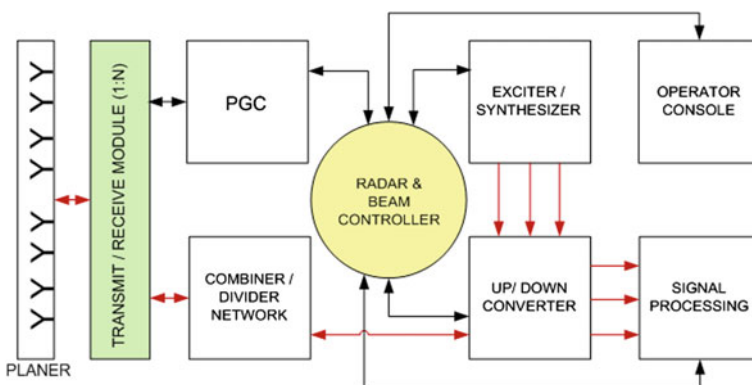


Fig. 1 System block diagram

The processed information is presented to the operator in the form of a bright spot on the display along with the target characteristics.

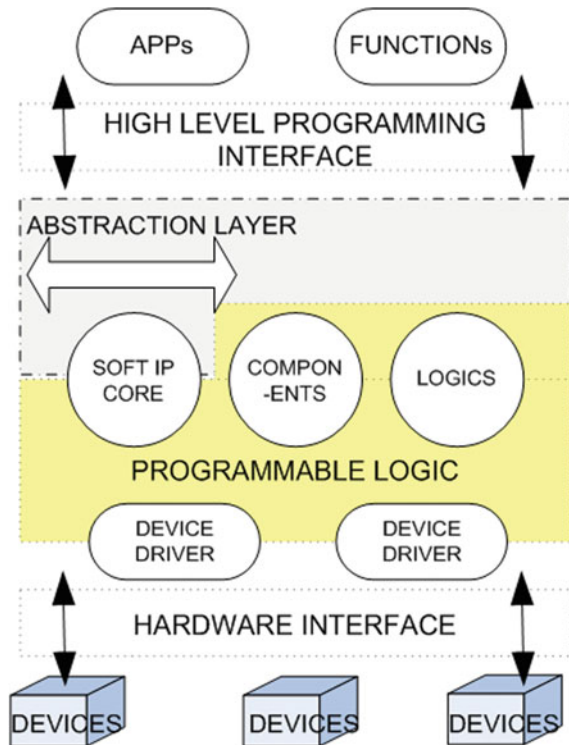
3 Proposed System

RBC acts as a focal point for all the communications in the system. It initiates commands to RF modules, SP module, and PGC. It provides synchronization among subunits. Design of such a system is complex and requires deep understanding of hardware, algorithms, data processing, and the operating environment.

It requires complexity of a CPU, memory controller, Serial Peripheral Interface (SPI) bus interface controllers, Serial interface and Local Area Network (LAN) module. It requires computations that can be pipelined.

To accomplish the requirements of RBC, FPGA based hardware with multi-layer software architecture is proposed and is shown in Fig. 2. Starting at the basic level, hardware layer provides the physical interface between the FPGA and the external world. Programmable logics are built using VHDL on FPGA. Functionalities are divided into number of small, well defined modules to improve code maintenance.

Fig. 2 FPGA-based multi-layer software architecture



Soft Processor IP core is interfaced with VHDL as a component. An abstraction layer provides high level programming interface between the processor and application specific functions.

Distinct advantages of proposed system are

- Higher flexibility and re-programmability
- Capability for parallel processing
- Small design cycle
- Lower design cost.

4 Design Methodology

4.1 Hardware Design

The proposed design is built around Stratix [3] series FPGA. It has power supply and conditioning circuitry. As shown in Fig. 3, it has peripheral memory interface for SRAM, SDRAM, Flash storage, and EEPROM. It contains multiple trans-receivers and a PHY device. It uses SPI bus [4] to interface RBC over Low Voltage Differential Signaling (LVDS) voltage levels [5].

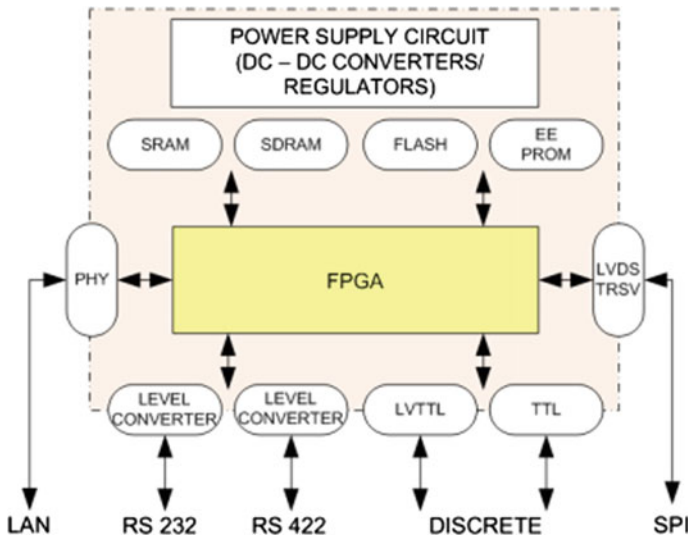


Fig. 3 Hardware design

4.2 Software Design

The proposed design has multiple software modules, they are as follows.

4.2.1 Phase Computation Module

Considering an array of (m, n) elements, the phase of the individual element is represented as follows. Here (m, n) are the indices of the individual antenna channel, P_A is the phase gradient in the azimuth and P_B is the phase gradient in the elevation.

$$\text{Phase}(m, n) = m * P_A + n * P_B \quad (1)$$

The phase gradients are represented as follows. Here “az” is steering angle azimuth, “el” is steering angle elevation, “ λ ” is wavelength of the EM wave being used, “dh” is the element spacing in azimuth and “dv” is element spacing in elevation.

$$P_A = (2\pi/\lambda) * dh * \text{Sin}(az) * \text{Cos}(el) \quad (2)$$

$$P_B = (2\pi/\lambda) * dv * \text{Sin}(az) * \text{Sin}(el) \quad (3)$$

Once the operator selects the volume to be scanned, RBC communicates the common phase gradient data for all the radiating elements over SPI bus. PGC receives this data, computes the position-wise phase values for each element, caters for the calibration error, and applies it to the individual elements on the panel.

SPI master module is located in RBC. It initiates and controls the entire communication. It derives PA values for a given azimuth from the beam control module.

4.2.2 Beam Control Module

System using electronic beam steering control can carry out the concurrent functions. A proper scheduling algorithm is to be devised to multiplex the requests of each function for transmission of energy. Such scheduler must be able to execute priority functions on the expense of the less critical ones. One such scheduler is provided in RBC and is interfaced with Radar Data Processor (RDP) over LAN interface.

As shown in Fig. 4, once the dwell requests are submitted to the scheduler, they are maintained in one of the dictated queue. Each dwell request consists of an identification number, priority, duration, and the desired time for execution.

Each queue contains requests corresponding to the similar function. Within a queue, dwell requests are ordered according to the desired time of transmission. Dwell request is played within the specified time period only if any higher priority request is not pending for the same specified period. Expired requests are unlinked from the queue.

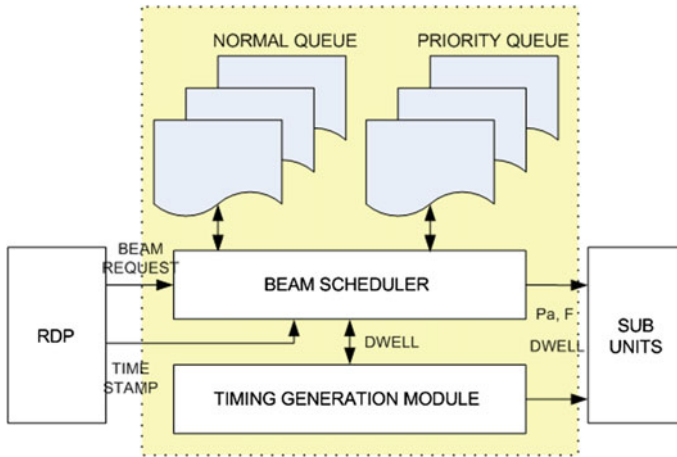


Fig. 4 Beam control module

Different type of dwell corresponds to specific waveform parameters which include PRF, carrier frequency, etc., and are synchronously executed in all the sub-systems via waveform and timing generation module.

4.2.3 Waveform and Timing Generation Module

Waveform and timing generation module (WTGM) provides all the timing and synchronization signals in a Radar system. Timing signals like dwell, PRT (Pulse Repetition Time) and cover pulse are generated and interfaced to subsystem over system clock. During initiated Built in Test (BIT), it generates “Test Pattern” control signals to RF subsystem and exciter unit for fault diagnostics. It facilitates the generation of simulated targets at the input of the receiver. Such “Test Pattern” provides a complete check of the total receiver chain including the exciter, down converter, SP, plot extractor, RBC and display unit.

4.2.4 Sensitivity Time Control

The power level at the input of a Radar receiver is inversely proportional to the fourth power of the target range. In some cases, echo signals from near range clutter saturates the receiver and small changes in signal strength go undetected. Here, increasing the threshold levels substantially, or reducing the receiver gain, could permit detection of such near range targets, but the distant range targets would go undetected. Therefore it is required to attenuate the large target returns occurring close to the trailing edge of the transmitted pulse and gradually reduced according to the system requirement. Such attenuation function is called as Sensitivity Time Control (STC).

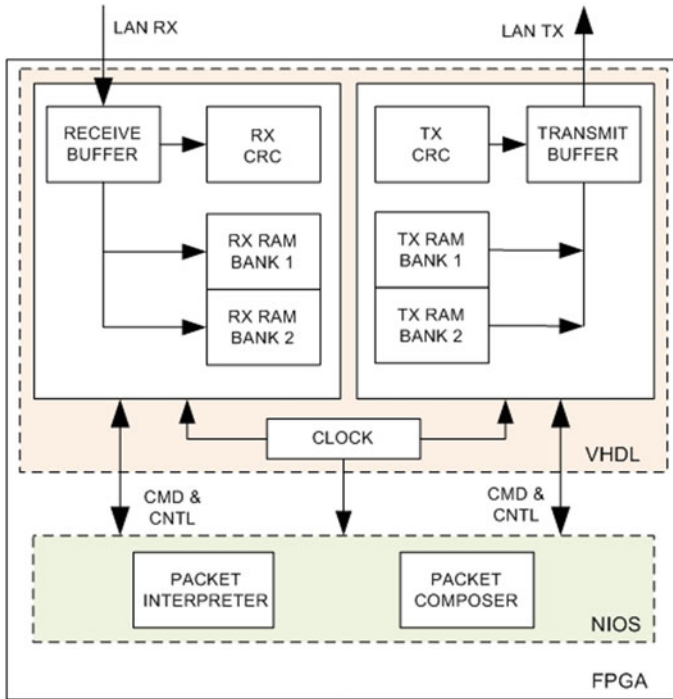


Fig. 5 LAN interface module

The control signals are synchronized with the Pulse Repetition Frequency (PRF). Each time a pulse is transmitted, the controls set the attenuation to maximum. It gradually reduces until a new pulse is transmitted.

4.2.5 LAN Interface Module

In the present design, an application tailored subset of UDP/IP stack is implemented for a cost effective connection to a network. As shown in Fig. 5, it consists of four main modules such as: receiver, packet interpreter, transmitter, and packet composer. Both the RAMs are implemented as dual port FPGA RAM.

The receiver module manages incoming packets. Once a new packet is detected, it is saved in one of the RX RAM bank. Each byte is sent to CRC (Cyclic Redundancy Check) checker, which progressively calculates the checksum. The packets are checked for its validity, headers, MAC address, IP address, CRC, message type, etc., in the packet interpreter module. Receive state machine is shown in Fig. 6.

The packet sender will act after initiation from one of the packet type's viz. User Datagram Protocol (UDP), Address Resolution Protocol (ARP), Reversed ARP (RARP) or Internet Control Message Protocol (ICMP). The required data is written

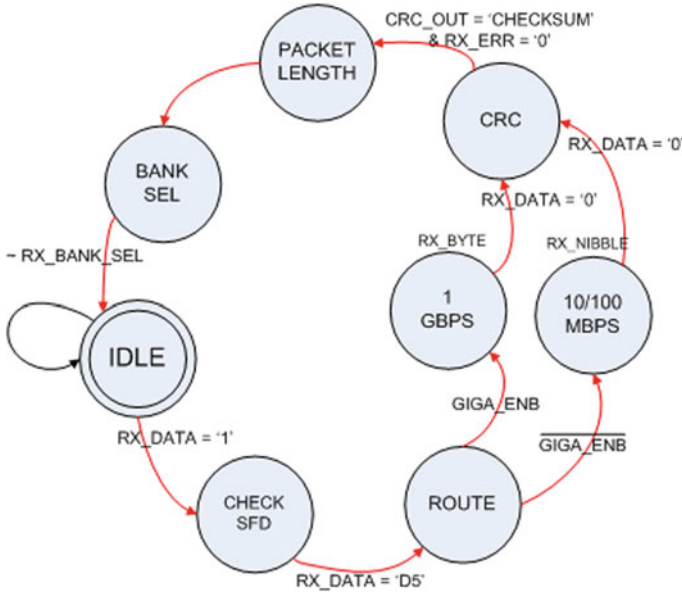


Fig. 6 Receive state machine

on one of the TX RAM bank. The transmitter module reads this data and puts out the packet to the PHY data bus and sets control signals. CRC generator progressively calculates check bits and is sent as the last byte. Transmit state machine is shown in Fig. 7.

5 Test Results

To ensure the timing requirements in the proposed design, verification and validation is performed using timing analysis tool and signal tap logic analyzer respectively.

The data exchange between Phase Computation Module in RBC and PGC over SPI is shown in Fig. 8. Signal “SCK_RC_IN” “SDI_RC_OUT” and “SDO_RC_IN” represents master clock, data from and data to RBC respectively. The communication is without glitches and errors.

The trace outputs are enabled in RBC for the scheduled reports. Different priority requests are scheduled and there timings are shown in Table 1. “Desired Time” represents the expected time for execution for the respective beam. Similarly, “Stamp” is current RBC time, “Scheduled” is the time at which RBC executed the request and “Delta” represents the difference between “Desired” and “Scheduled” time.

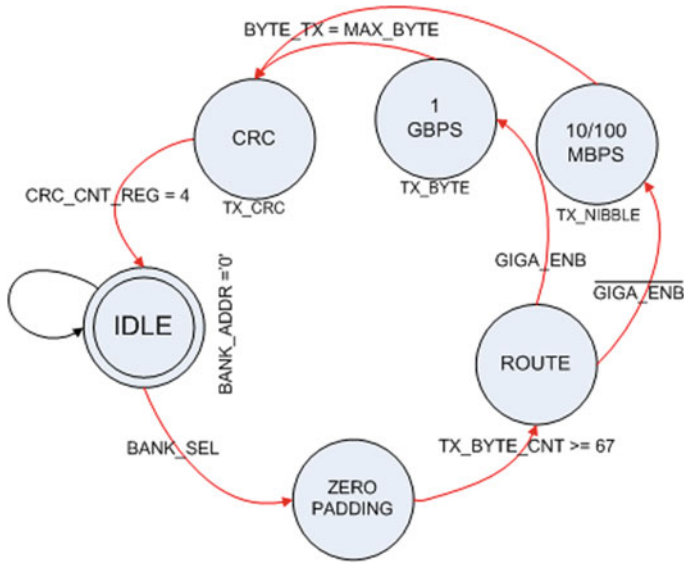


Fig. 7 Transmit state machine

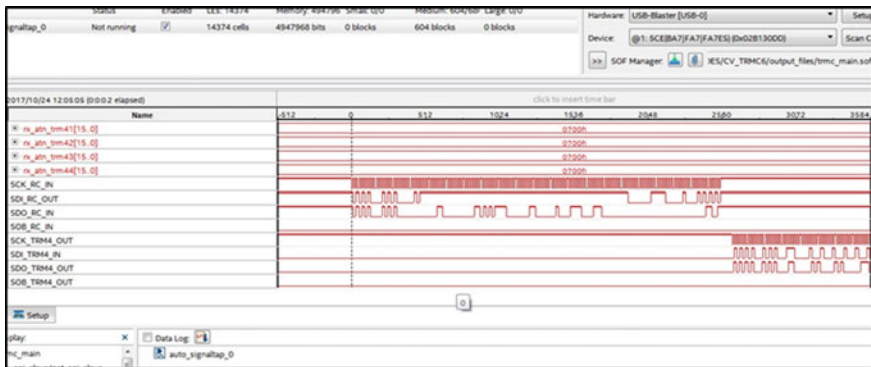


Fig. 8 Data exchange between RBC and PGC

Table 1 Timing results

ID	Priority\l	Time (in s)				
		Length	Absolute			
		Duration	Stamp	Desired	Scheduled	Delta
1	1	67	18,437	18,504	18,449	55
1	2	218	18,473	18,691	18,609	82
1	2	219	18,633	18,852	18,769	83
1	2	220	18,793	19,013	18,929	84
1	5	6022	18,953	24,975	24,887	88

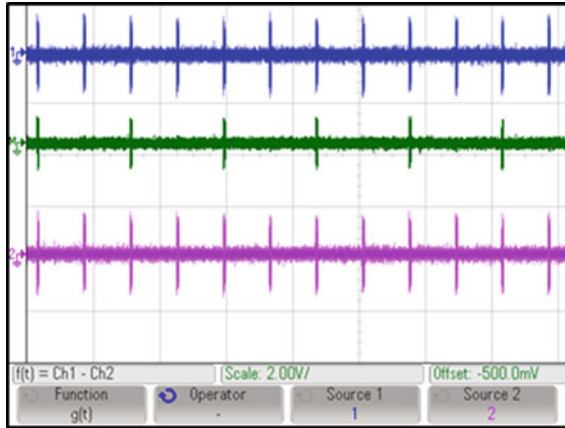


Fig. 9 Doppler in BIT mode



Fig. 10 Data exchange with LAN transmit module

As shown in Fig. 9, Doppler is simulated in BIT mode using WTGM. It represents the IF signal on channels and there phase difference. The target profile represents Doppler of PRF/2.

As shown in Fig. 10, signal “tx_clk” and “stage_reg” represents the input clock and state of LAN transmit module. During transmit initiation, signal “tx_mem_bank_sel” is changed and “tx_mem_bank_reg” is set to zero. Number of bytes are checked and based on “max_byte_cnt1_reg” value zero bytes are padded. Signal “gig_en” is enabled for byte-wise transmission. Signal “byte_reg” holds transmit byte count and once it reaches to its maximum value CRC byte transmission is enabled.

Received data is shown in Fig. 11. Signal “rx_dv” goes high and initiates data reception. Signal “rx_data_reg” indicates the input data and is checked for preamble. If “gig_en” is set then data is received byte wise. Each byte is stored in Rx memory

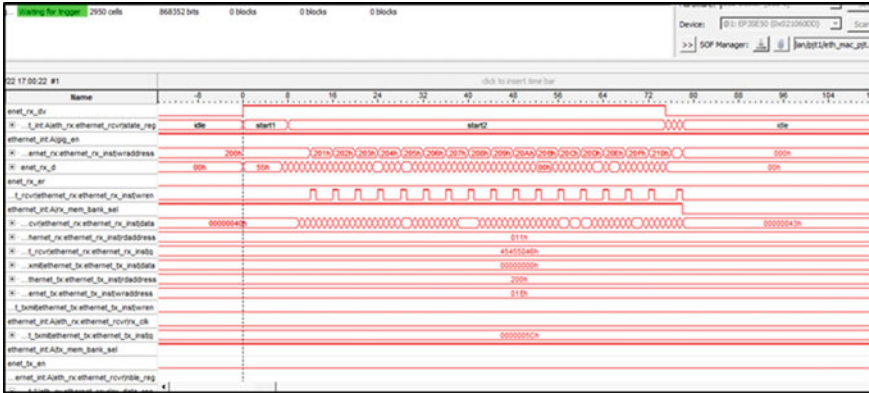


Fig. 11 Data exchange with LAN receive module

bank and “wr_address” is incremented. Once “rx_dv” is low checksum is matched, memory address is locked and next memory bank is selected.

6 Conclusion

The work presents FPGA-based Radar and beam controller for surveillance system. The key modules of the design are discussed. The test results suggest that the design meets the required functionality and the timing parameters are well within specifications. Use of programmable device offers flexibility with high performance.

Acknowledgements Authors are very much thankful to General Manager of Military Radar SBU in Bharat Electronics Ltd., Bangalore for giving an opportunity to publish this technical paper. Thanks to dear colleagues whose kind support made this study and design possible.

References

1. Kraus, J. D. (1997). *Antennas*. New York: Tata McGraw-Hill.
2. Stimson, G. W. *Introduction to airborne radar*. Scitech Publishers.
3. Stratix Device Handbook, www.altera.com.
4. Chen, D. (2016). *Introduction to SPI interface*.
5. An Overview of LVDS Technology, www.ti.com.

Effect of Lattice Topologies and Distance Measurements in Self-Organizing Map for Better Classification



Sathiapriya Ramiah

Abstract Self-Organizing Map (SOM) is a widely used algorithm in artificial neural network for classification. Despite the general success of this algorithm, there are several limitations which some of them are poor classification accuracy and slow rates of convergence when the standard lattice topology and distance measurement are implemented. This paper investigates the performance of SOM using different topologies and different distance measurements. The results obtained showed that SOM with hexagonal topology and Euclidean distance measurement outperforms other topologies and distance measurement using at any scale datasets.

Keywords Self-organizing map · Best matching unit · Topology · Distance measurement · Accuracy

1 Introduction

Classification can be defined as assigning objects to groups on the basic criteria made on the objects into several classes. It is mainly used to understand the existing data pattern and predict how new instances will behave. Classification plays an important role in many fields including medical, education, business, marketing, shipping, transportation, and others.

Self-Organizing Map (SOM) also known as Kohonen SOM is a neural network algorithm invented by the founder of the Neural Networks Research Centre, Professor Kohonen [1]. Similar to most artificial neural networks, SOM operates in two modes namely training and testing. Training is a competitive process called vector quantization to build the map using input data samples. Testing process automatically classifies a new input sample into respective classes [2].

S. Ramiah (✉)

Asia Pacific University of Technology & Innovation, Kuala Lumpur, Malaysia
e-mail: sathiapriya@yahoo.com

© Springer Nature Singapore Pte Ltd. 2019

N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_16

183

SOM is called a topology preserving map because of the mapping of the nodes from high dimensional space into low dimensional, usually two-dimensional lattice [3]. Lattice topology specifies the arrangement of nodes on the map forming the layer. Therefore, in this paper, three types of topologies, mainly triangular, rectangular, and hexagonal are evaluated.

There are several ways to work out the distance between two points in multidimensional space for classification. Changing the distance measure can have a major effect on the overall performance of the classification system. The distances are measured using Euclidean, Manhattan, Chebychev, and Canberra in this experiment.

The rest of the paper is organized as follows. In Sect. 2, experimental dataset used for evaluation are discussed. SOM is described in detailed in Sect. 3. Sections 4 and 5 illustrate lattice topology and distance measurement. Whereby experimental results discussed in Sect. 6. Finally Sect. 7 state the concluding remarks.

2 Experimental Data

Dataset used in this experiment is obtained from UCI machine learning repository to evaluate the performance of SOM in classification [4]. Iris data classifies it species, Breast Cancer Wisconsin (Diagnostic) data predict whether the cancer is benign or malignant and Car Evaluation data evaluates acceptability of a car. Iris data represents small-scale dataset. It contains data of 105 instances for training and 45 instances for testing. Meanwhile, Breast Cancer Wisconsin (Diagnostic) data used to represents medium scale dataset. 400 instances used for training and 169 instances were used for testing purpose. Finally, Car Evaluation representing large-scale dataset. It contains total of 1200 instances used in training and 528 instances for testing.

3 Self-organizing Map (SOM)

SOM is two layer network comprises an input layer and an output layer of neurons in a two-dimensional lattice as is shown in Fig. 1. Each node in the network has a specific coordinate and contains weight vectors of the same dimension as the input vectors [5]. Each neuron in input layer is connected to all the neurons of the output via the weight. There are main two steps: training and testing. The network needs to be trained using input and correct output so that it can learn the pattern of output to be produced. Once the training is done, network needs to be tested using real sample data.

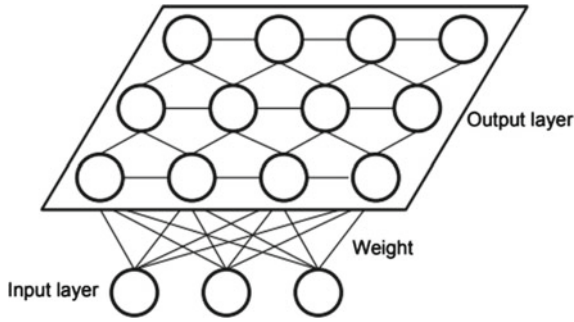


Fig. 1 Typical architecture of SOM. *Source* Lohninger [8]

A few parameters need to be specified in order to develop SOM are defined below

i. *Number of input nodes*

The number of input determines the number of data to be fed into the network for training and testing.

ii. *Map size*

Map size determines the size of the output layer.

iii. *Number of neurons*

The number of neurons on the output layer is determined by the map size. For this study, map size of 10×10 will be used.

iv. *Number of learning iteration*

The number of learning iteration is used to train the network.

v. *Learning rate*

Learning rate decides the speed of algorithm and the value is between 0 and 1. The larger the value, the faster the network grows.

vi. *Lattice topology*

Lattice topology triangular, rectangular or hexagonal to be identified as this is one of the comparison criteria.

vii. *Distance measurement*

Distance measurement method to be identified as this is one of the comparison criteria too.

Once the parameters are defined, training process shall begin. The training processes in steps as described below

a. *Initialization of weights for output units*

Weight vector (W) for each node initialized to small standardized random values between 0 and 1.

b. *Present input*

Input vector (V) is fed into the network to train. The inputs are the training dataset.

c. *Calculate the Best Matching Unit (BMU)*

BMU is determined by iterating all the nodes and calculate the distance between each node's weight vector and input vector. The node which has weight vector closest to the input vector is labeled as the BMU. Five different methods will be used to calculate the distance. They are Euclidean, Manhattan, Chebychev and Canberra. This is to identify which measurement gives better accuracy.

d. *Update neighbouring nodes*

Weight vector of every node within the BMU's neighbourhood including the BMU adjusted so that their weight gets more similar to the weight of input vector based on following equation:

$$W(t+1) = W(t) + \Theta(t)L(t)(V(t) - W(t)), \quad (1)$$

where,

t time step (iteration of the loop)

L learning rate which decreases over time

Decay of learning rate calculated based on following:

$$L(t) = L_0 \exp\left(-\frac{t}{\lambda}\right) \quad (2)$$

As training goes on, the area of the neighbourhood shrinks gradually over time to the size of just one node which is the BMU by formula

$$\Theta(t) = \exp\left(-\frac{\text{dist}^2}{2\sigma^2(t)}\right) \quad (3)$$

$\Theta(t)$ influence rate

$\sigma(t)$ width of the lattice at time t

e. *Determine convergence*

Steps b, c and d are repeated for all input data. Convergence will happen when the training stops once an accurate topological map is obtained. Upon completion of training process, the output vector holds the cluster for classification. Meanwhile, weight vector defines the attributes of an element that falls into the segment. Both this information will be used for evaluating test data.

In the testing phase, the dataset fed into the algorithm together with output vector and weight vector obtained from training process. The output achieved from the testing is compared with the expected target output to determine the ability of SOM to classify data into correct predefined classes.

4 Lattice Topology

A topological map is a mapping that preserves the relative distance between the points. A lattice is connected if neighbouring points in the input space are mapped to nearby map units [6]. Neighbourhood function is applied on top of the topology. In this study, triangular, rectangular and hexagonal topologies are experimented.

In triangular topology, neurons are arranged in a triangular lattice with each neuron having three immediate neighbours. However, in rectangular and hexagonal topologies, neurons are arranged in a rectangular and hexagonal lattices with each neuron having four and six immediate neighbours respectively.

5 Distance Measurement

There are several ways to work out the distance between two points in multidimensional space. Changing the distance measurement can have a major impact on the overall performance of the classification system [7]. Below are the four distance measurements used for this experiment

(i) Euclidean Distance

Calculates the square root of the difference between coordinates of pair of nodes.

$$d_{ij} = \sqrt{\sum_{k=1}^K (x_{ik} - x_{jk})^2} \quad (4)$$

(ii) Manhattan Distance

Calculates absolute difference between coordinates of pair of nodes.

$$d_{ij} = \sum_{k=1}^K |x_{ik} - x_{jk}| \quad (5)$$

(iii) Chebychev Distance

Calculate the absolute magnitude of the differences between coordinate of a pair of nodes. Chebychev also known as maximum value distance.

Table 1 Analysis of iris dataset

	Euclidean (%)	Manhattan (%)	Chebychev (%)	Canberra (%)
Triangular	95.56	77.78	97.78	77.78
Rectangle	97.78	77.78	97.78	77.78
Hexagonal	100.00	84.44	97.78	77.78

$$d_{ij} = \sqrt[\lambda]{\sum_{k=1}^K |x_{ik} - x_{jk}|^\lambda} \quad \lambda = \infty \quad (6)$$

(iv) Canberra Distance

Calculate the sum of series of fraction differences between coordinates of a pair of nodes.

$$d_{ij} = \sum_{k=1}^K \frac{|x_{ik} - x_{jk}|}{|x_{ik}| + |x_{jk}|}, \quad (7)$$

where,

x_i input to the node i at time k

x_j output node j at time k with weight from input node i

k no of iterations

for all of the above measurements.

6 Experimental Results

In this study, SOM is tested in different topology and distance measurement using datasets as specified in Sect. 2.

Iris dataset represents small-scale data for this experiment. The result obtained using triangular, rectangular and hexagonal topologies for this dataset is summarized in Table 1.

For iris dataset, classification accuracy shows good result. All three topologies work well with Euclidean distance whereby it gives 100% accuracy for hexagonal and Euclidean combination. Rectangular and triangular topologies with Euclidean also shows high accuracy percentage. Followed by Chebychev distance measurement where all topologies give the same accuracy. Manhattan and Canberra show similar result in term of its classification accuracy, but hexagonal topology using Manhattan shows a slight increase in its percentage. Overall performance comparing all three topologies confirms that hexagonal topology works better compare to triangular and rectangular topologies using Euclidean distance measurement.

Table 2 Analysis of Breast Cancer Wisconsin (diagnostic) dataset

	Euclidean (%)	Manhattan (%)	Chebychev (%)	Canberra (%)
Triangle	88.76	88.76	88.76	71.00
Rectangle	88.76	88.76	88.76	71.00
Hexagonal	88.76	88.76	88.76	79.89

Table 3 Analysis of car evaluation dataset

	Euclidean (%)	Manhattan (%)	Chebychev (%)	Canberra (%)
Triangular	90.91	90.91	90.91	85.23
Rectangle	91.86	94.70	91.86	71.97
Hexagonal	96.60	90.91	85.23	75.76

For the second experiment, Breast Cancer Wisconsin (Diagnostic) dataset was used as medium scale dataset. The result obtained is summarized in Table 2.

For breast cancer dataset, almost all three topologies show similar classification accuracy for Euclidean, Manhattan and Chebychev distance measurement. Accuracy percentage does not increase even if multiple testing is done. However, Canberra distance measurement showed a slightly higher accuracy percentage for hexagonal topology despite the other two remain lowest among the overall testing output. Overall performance comparing all three topologies confirms that hexagonal topology works better compare to triangular and rectangular topologies using any distance measurements for this dataset.

For the last experiment, Car Evaluation data used to test as large-scale data. Result for this experiment is represented in Table 3.

Based on the above result, can be seen that hexagonal topology using Euclidean distance gives the highest accuracy. Rectangle topology with collaboration of Manhattan can be said as good classification too. Results of other testing show similar classification accuracy percentage and among all Canberra proves the poorest. Overall performance comparing all three topologies confirms that hexagonal topology works better compare to triangular and rectangular topologies using Euclidean distance measurement.

Based on the above experiments, can be seen that classification accuracy differs in each type of dataset. As described above, this study is done to address two issues. First to investigate which lattice topology is better and second is to identify which distance measurement proves better classification. Comparison between lattice topologies and distance for each dataset is done as described in Table 4.

Euclidean and Chebychev proved good classification accuracy for small-scale dataset. Euclidean, Manhattan and Chebychev work well for medium-scale dataset. For larger scale dataset, Euclidean and Manhattan performed better.

Table 4 Summary of result analysis based on topology and distance measurement

Topology	Iris	Breast Cancer Wisconsin (Diagnostic)	Car evaluation
Triangular	Chebyshev	All except Canberra	All except Canberra
Rectangular	Euclidean, Chebyshev	All except Canberra	Manhattan
Hexagonal	Euclidean	All except Canberra	Euclidean

Hexagonal topology showed higher accuracy percentage for all scale dataset followed by rectangular. Triangular topology does not show much performance to be stressed out. Euclidean distance measurement moves well with all topologies for all scale dataset.

7 Conclusion

SOM is a famous algorithm for classification yet there are areas of research to be carried out and one of it is classification accuracy. Therefore, in this paper lattice topologies and distance measurements are investigated to evaluate classification accuracy.

Based on experimental results, can be concluded that SOM using hexagonal topology enhances the performance of classification accuracy mainly due to connection between neurons in SOM map in hexagonal topology is more compared to rectangular and triangular.

Euclidean distance has proven to calculate distance between nodes perfectly in every topologies analyzed. The efficiency of classification relies on computation complexity. Euclidean distance which requires less computation is seen to reduce the computation complexity of classification system compared to other distance measurements which are used in this experiment. Since the computation complexity is less, it proves better performance in term of accuracy.

The size of dataset used does not reflect much difference in this study probably due to the data used from clean dataset. Thus, network corresponds accordingly to form good classification.

References

1. Kohonen, T. (1997). Self-organizing maps. In *Springer series in information sciences*. Berlin, Heidelberg, New York: Springer.
2. Pastukhov, A., & Prokofiev, A. (2016). Kohonen self-organizing map application to representative sample formation in the training of the multilayer perceptron. *St. Petersburg Polytechnical University Journal: Physics and Mathematics* 2, 134–143.
3. ChandraShekar, B. H., & Shoba, G. (2009). Classification of documents using Kohonen's self-organizing map. *International Journal of Computer Theory and Engineering*, 1(5), 610–613.

4. Dua, D., & Karra Taniskidou, E. (2017). *UCI machine learning repository* <http://archive.ics.uci.edu/ml>. Irvine, CA: University of California, School of Information and Computer Science.
5. Jaramillo-Vacio, R., Ochoa-Zezzatti, A., & Figueroa-Godoy, F. (2015). Self-organizing map improved for classification of partial discharge using desirability function. *International Journal of Combinatorial Optimization Problems and Informatics*, 6(3), 49–65.
6. Chaudhary, V., Ahlawat, A. K., & Bhatia, R. S. (2013). An efficient self-organizing map learning algorithm with winning frequency of neurons for clustering application. In *Proceedings of International Advanced Computing Conference* (pp. 664–668). IEEE.
7. Singh, A., Yadav, A., Rana, A. (2013). K-means with three different distance metrics. *International Journal of Computer Applications*, 67(10).
8. Lohninger, H. (2018). Kohonen Network—Background Information, http://www.lohninger.com/helpsuite/kohonen_network_-_background_information.htm.

Evaluation and Classification of Road Accidents Using Machine Learning Techniques



Jaspreet Singh, Gurvinder Singh, Prithvipal Singh and Mandeep Kaur

Abstract The root cause of traffic accidents is hard to determine these days due to complex combination of characteristics like mental state of driver, road conditions, weather conditions, traffic, and violations of traffic rules to name a few. The deployment of machine learning classifiers has replaced traditional data mining techniques like association rule mining. Application of machine learning techniques in the field of road accidents is gaining popularity these days. This paper utilized four machine learning techniques viz. Naïve Bayes, k-Nearest Neighbours, Decision trees, and Support Vector Machines for evaluation of Punjab road accidents. This work had a challenge of performing parametric evaluation to extract highly relevant parameters especially for Punjab. The outcome of this study yields 12 most suitable parameters and maximum performance of 86.25% for Decision Tree classifier.

Keywords Road accidents · Machine learning · Parametric evaluation · Punjab road safety

1 Introduction

Road accident is one of the most prominent issues in the modern times of equipped and fast moving traffic on roads. World Health Organization has reported top ten disastrous reasons for taking human's life, and unfortunately road accidents come at ninth place, where cardiac arrest sits on the top. The impact on society seems significant when cost of casualties and injuries from road accidents is evaluated. The young researchers these days have witnessed the increasing trend of evaluating the causes and implementing safety measures in order to preserve human life from dangerous road mishaps. Also, the trending machine learning techniques have facilitated the overall process of understanding the features like demographic factors, physical

J. Singh (✉) · P. Singh · M. Kaur
Department of CS, GNDU, Amritsar 143005, Punjab, India
e-mail: profjaspreetbath@gmail.com

G. Singh
Department of CS, Faculty of Engineering and Tech, GNDU, Amritsar 143005, Punjab, India

© Springer Nature Singapore Pte Ltd. 2019
N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_17

factors, victim related factors, driving environment, mental factors, and other culprit features for accident severity. This paper employs four machine learning techniques viz. Naïve Bayes (NB), K-Nearest Neighbours (k-NN), Decision Trees (DT), and Support Vector Machines (SVM). The dataset for the evaluation is taken from Punjab government's authentic organization named Punjab Road Safety Organization. However, it was difficult to extract data from reports in diverse Punjabi fonts but, the use of Google Punjabi to English translator has facilitated the data preprocessing task marvelously. Section 2 of the manuscript provides literature work of some state-of-the-art machine learning algorithms used for evaluation and classification of road accidents. The parametric evaluation (Sect. 3 of the manuscript) is exhaustively performed thereby examining around 71 parameters taken from 12 benchmarks published in the same research direction. Further, the close investigation of dataset for Punjab state of India motivated this work to consider 12 highly relevant parameters associated with Punjab road mishaps. Section 4 gives experimental results, where the highlights of best performed classifier (Decision Tree) are being presented in terms of confusion matrix followed by concluding remarks in Sect. 5.

2 Related Work

Road accident is a global issue with skyrocketing trend where the immediate need is to analyze data through some data classification algorithms which help in anticipating the relevant factors influencing the problem of accident severity. Table 1 shows summary of related research work, where the well-known machine learning classifiers are exploited for the evaluation of traffic accident analysis.

3 Parametric Evaluation

This study considers twelve benchmarks related to evaluation of road accidents. The close investigation of 71 parameters threw light on enormous range of possibilities where the accurate understanding of causes behind accidents is gained. However, most of these parameters are not suitable for evaluation of road accidents in the Punjab state of India. Table 2 gives insights to consider the 12 contributing parameters viz. Road type, Vehicle type, Vehicle speed, Light conditions, Traffic volume, Accident time, Accident severity, Stray animal, Alcohol consumption while driving, Age, Gender, and mental conditions of driver.

4 Experimental Results and Discussion

This work has utilized Scikit-Learn package from SciPy library of Python 3.6 for experimental evaluation. The classification models viz. Naïve Bayes (NB), K-nearest

Table 1 Summary of related work

Author(s), publication, (Year)	Description	Dataset	Attributes	Machine learning techniques	Performance
Ramani, R. G., et al. WCECS (2012) [1]	Analyzed and classified road traffic data through data mining techniques	Obtained data from Fatality Analysis Reporting System (FARS)	State, Country, month, date, collision, seating position, age, gender, range, injury, severity, ejection, alcohol	C4.5, CR-T, ID3, CS-CRT, CS-MC4, Naïve Bayes, Random trees, Arc-X4, Feature ranking algorithm	Arc-X4 meta classifier has shown highest accuracy of 99.73%
Elfadil A. Mohamed, DAVID JCC, (2014) [2]	Proposed a framework for predicting cause of road traffic accidents	Taken from police department in Dubai and United Arab Emirates	Location, vehicle type, country a driver belongs to, age, year of vehicle made, gender	Multiclass support vector machine	Precision = 0.767, Recall = 0.754, F1 measure = 0.752, Accuracy = 75.395%
ARIMURA et al. Infrastructure Planning Review (2007) [3]	Performed road accident analysis by extracting dangerous locations with high fatality and injury	Taken dataset from Institute for Traffic Accident Research and Data Analysis (ITARDA)	Road structure, driving environment, designated maximum speed	SVM	Accuracy of classification observed was 70%
Amira. A. El Tayeb., et al., IJSCE, (2015) [4]	Evaluation of traffic accidents using association rule mining algorithm	Dataset was taken from Dubai traffic department, UAE from 2008 to 2010	Day, month, year, number of injured persons, cause of accident, type of accident, gender, age, vehicle type, alcohol, seat belt, lightening, weather, road conditions	Apriori algorithm and predictive apriori algorithm	Apriori algorithm was found more accurate than predictive apriori algorithm

(continued)

Table 1 (continued)

Author(s), publication, (Year)	Description	Dataset	Attributes	Machine learning techniques	Performance
Martin et al., Elsevier (2014) [5]	Proposed a decision tree based algorithm for detection and improvement of susceptible elements in Spanish roads	Dataset was taken from Spanish national government	Layout, signaling and marking of road obstacles, intersection, junction, roundabout, crossings, tunnels, road access, etc.	Data mining techniques (Decision Tree)	Performance of Decision Tree was satisfactory
Fu. H., et al., IEEE, (2011) [6]	Presented a prediction technique for traffic accidents using neural networks	The national accident statistics from 1985 to 1997, and 1995 to 1999	Number of accidents, time, death, direct loss	Backpropagation neural networks	Improved LMBP performed better than traditional BP network
Miao, Chong., et al. Informatica, (2005) [7]	Developed a machine learning based model for accurate classification of severity of injuries in road accidents	Obtained from National Automotive Sampling System (NASS) and General Estimates System (GES)	Age, gender, eject, alcohol, restraining system, body type, vehicle type, rollover, road surface condition, light condition	Artificial Neural Networks (ANN) using hybrid learning, Decision Trees (DT), Support Vector Machines (SVM)	DT_ANN method outperforms ANN and DT with maximum average accuracies of 91.53% and 90.00% for training and testing respectively
K. Geetha., et al., IJARCSSE, (2015) [8]	Proposed a hybrid machine learning model for analysis of traffic accidents	Traffic accident dataset taken from Tamil Nadu Government	Age, gender, alcohol usage, eject vehicle, body type, vehicle role, vehicle age, rollover, road surface conditions, light conditions, weather conditions	J48, PART algorithm, Hybrid DTANN	DTANN outperforms

(continued)

Table 1 (continued)

Author(s), publication, (Year)	Description	Dataset	Attributes	Machine learning techniques	Performance
Sohn, S. Y., et al., Elsevier, (2003) [9]	Evaluation of severity of road accidents in Korea	International database on road traffic and accident (IDRTA)	Road width, shape of car body, accident category, speed before accident, violent drive, and protective drive	DTNN, Dempster Shafer algorithm, Bayesian procedure, logistic model, ensemble, bagging, clustering, and K-means algorithm	Clustering method using Decision Tree produced best accuracy of 76.10%
Vasavi, S., Springer Nature Singapore, (2018) [10]	Proposed model analyze hidden patterns to identify the root causes for road accidents	Dataset was taken from police authorities of Andhra Pradesh (Nunna Dataset)	Traffic, time of accident, age, weather, speed limit, type of accident, deceased emotions, hospital reported etc.	K-medoids, and EM clustering algorithm	Precision = 0.8, recall = 0.6, F-measure = 0.69
Razaq, S., et al. ICCEEE, (2016) [11]	Coined a system based on multiple factors for evaluation of road accidents	Hypothetical dataset obtained from virtual sensors in the simulator	Environmental, physical and mental factors	Used fuzzy logic's modeling in Multi Factor Based Road Accident Prevention System (MFBRAPS)	Proposed system outperformed over traditional methods
Kumar, S., ICCCS (IEEE), (2015) [12]	Performed clustering and association rule mining of road traffic accidents	Taken data from Emergency Management Research Institute (EMRI)	Victim injured, age, gender, economic status, time of accident, day, month, region, lightning, road features	K-mode clustering and association rule mining algorithms	Clustering followed by association rule mining yields satisfactory results

Table 2 Parametric comparison among twelve benchmarks

Parameters		Benchmarks																								
		B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12													
Demographic factors	Weather	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X				
		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X			
	Road conditions	X	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Physical factors	Light conditions	X	✓	X	X	✓	X	X	X	✓	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
	Traffic	X	X	✓	X	✓	X	X	X	✓	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
	Vehicle conditions	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Accident severity	✓	X	X	X	✓	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
Possible Injury	✓	X	X	X	✓	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		

(continued)

Table 2 (continued)

Parameters	Benchmarks											
	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12
Accident type	Non-incapacitating	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗
	Incapacitating	✓	✗	✗	✓	✗	✓	✓	✗	✗	✗	✓
License	Collision Manner	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	Seating Position	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	Harmful Event	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
	Punctured Tire	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
	Reckless driving	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
	Wrong direction	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
	Incorrect pass	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
	U-turn	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
	Stray animal	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
	Sleep	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Ejection	Door open	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
	With	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
Destructive activity	Without	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
	Path	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
Overspeed	Extrication	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	Eating	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
	Texting	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
	Overspeed	✗	✓	✓	✗	✗	✗	✗	✓	✓	✓	✗

(continued)

Table 2 (continued)

Parameters		Benchmarks												
		B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	
Victim related	Driver's profile	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	State	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Country	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Age	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Experience	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Gender	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Nationality	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Economic Status	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Driver	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Passenger	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Year	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
On the spot death	Month	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Day	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Alcohol	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Drug test	Other Drug	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Suburb	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Accident location	Country side	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Month	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Date	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Accident time	Day	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Time	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

(continued)

Table 2 (continued)

Parameters	Benchmarks											
	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12
Mental factor	Year	X	X	X	X	X	X	✓	X	X	X	X
	Mental condition of driver	X	X	X	X	X	X	X	X	X	✓	X
	Fear	X	X	X	X	X	X	X	X	X	✓	X
	Temper	X	X	X	X	X	X	X	X	X	✓	X
	Behavior	X	X	X	X	X	X	X	X	X	✓	X
	Negligence	X	✓	X	X	X	X	X	X	X	✓	X
Others	No. of persons injured	X	X	X	✓	X	X	X	✓	X	X	X
	Accident cause	X	X	X	✓	X	X	X	X	X	✓	X
	Signaling	X	X	X	X	✓	X	X	X	X	X	X
	Ambulance used	X	X	X	X	X	X	X	X	✓	X	X

B1 Shanthi et al. [1], B2 Mohamed [2], B3 Hironobu et al. [3], B4 Amira et al. [4], B5 Luis et al. [5], B6 Fu Huilin et al. [6], B7 Chong et al. [7], B8 Geetha et al. [8], B9 So et al. [9], B10 Vasavi [10], B11 Sheeba et al. [11], and B12 Kumar et al. [12]

Table 3 Classification accuracy

S. No.	Classification model	Accuracy (%)
1	Naïve Bayes	79.99
2	K-Nearest neighbor	85.73
3	Decision tree	86.25
4	Support vector machine	84.33

Table 4 Top three contributing parameters for accidents

Contributing factor	Percentage of accidents (%)
Mental condition of driver	24.57
Alcohol consumption	26.49
Speed of vehicle	16.92

neighbours (k-NN), Decision Trees (DT), and Support Vector Machine (SVM) were built to analyse causes for road accidents in Punjab. The dataset consists of 38,604 accident cases reported for Punjab state on www.punjabroadsafety.org [13]. The initial 500 cases were manually preprocessed, thereby providing average values at missing places for collecting 12 parameters. The rest of the cases are managed by Python code to fill missing values either with null in case of nominal factors or with average values for ratio scale parameters. The dataset is sliced in the proportions of 80, 10, and 10% for training, testing and cross-validation purposes respectively. Table 3 gives classification accuracies of four machine learning models. The performance of Decision Tree learning model is higher among the four classifiers.

The close investigation of dataset shows that the top three contributing parameters among 12 are mental conditions of driver, alcohol consumption, and speed of vehicle. Table 4 shows the topmost three contributing factors in terms of percentage of accidents [13]. The confusion matrix of 12 parametric classes classified with Decision Tree method depicts an error rate of 13.75% thereby yielding the accuracy of 86.25% as shown in Table 5.

5 Conclusion

This paper evaluated road accidents using four machine learning classifiers on Punjab road safety organization's dataset. The twelve benchmarks illuminated 71 parameters extensively; however, these parameters depend upon regional factors. The keen examination of all collected parameters (in Table 2) assisted the focus on 12 parameters which are most suitable for evaluation of Punjab road accidents. The results obtained after 10-fold cross-validation show that Decision Tree classifier carries maximum accuracy of 86.25% among four classifiers. The main causes behind the road accidents in Punjab come from three most contributing factors viz. mental state of driver, alcohol consumption, and speed of vehicle. The future scope of this work

Table 5 Confusion matrix of decision tree classifier (Error rate: 13.75%)

Error rate: 0.1375		RT	VT	AT	TV	LC	MC	AC	AS	Age	Gen	Sp	SA	Sum
RT	1140	0	2	2	0	0	3	2	0	0	0	1	0	1150
VT	0	1040	2	1	2	2	1	2	100	0	0	0	0	1148
AT	2	0	1029	0	10	1	1	0	0	0	0	0	0	1042
TV	0	0	0	1230	0	0	0	22	208	0	0	72	1	1533
LC	0	0	0	0	2310	0	0	0	0	0	0	0	0	2310
MC	12	0	1	2	0	0	6964	2326	0	10	0	103	0	9418
AC	0	0	0	0	0	0	101	7937	2114	2	0	0	0	10154
AS	2	0	0	0	1	0	0	0	2277	2	2	12	2	2298
Age	0	12	10	0	0	0	0	0	0	1894	0	0	0	1916
Gen	0	0	0	13	0	0	0	1	0	0	752	0	0	766
Sp	0	0	0	0	32	105	0	0	0	0	0	6349	0	6486
SA	2	0	2	0	3	0	0	0	2	0	0	0	374	383
Sum	1158	1052	1046	1248	2358	7175	10,290	4701	1908	754	6537	377	38,604	

Here *RT* Road Type, *VT* Vehicle Type, *AT* Accident Type, *TV* Traffic Volume, *LC* Lightening Condition, *MC* Mental Condition, *AC* Alcohol Consumption, *AS* Accident Severity, *Age* Age of driver, *Gen* Gender, *Sp* Speed of Vehicle, *SA* Stray Animal

is to consider sentiment analysis of road accident's cases using ensemble classifiers and deep neural network.

References

1. Ramani, R. G., & Shanthy, S. (2012). Classifier prediction evaluation in modeling road traffic accident data. In *2012 IEEE International Conference on Computational Intelligence and Computing Research*. <https://doi.org/10.1109/iccic.2012.6510289>.
2. Mohamed, E. A. (2014). Predicting causes of traffic road accidents using multi-class support vector machines. *Journal of Communication and Computer*, *11*(5), 441–447. <https://doi.org/10.17265/1548-7709/2014.050004>.
3. Arimura, M., Hasegawa, H., Fujii, M., & Tamura, T. (2007). A consideration on application of support vector machine for non-linear optimization. *Infrastructure Planning Review*, *24*, 421–426. <https://doi.org/10.2208/journalip.24.421>.
4. El Tayeb, A. A., Pareek, V., & Araar, A. (2015). Applying association rule mining algorithms for traffic accidents in Dubai. *International Journal of Soft Computing and Engineering*, *5*(4), 1–12.
5. Martín, L., Baena, L., Garach, L., López, G., & De Oña, J. (2014). Using data mining techniques to road safety improvement in Spanish Roads. *Procedia—Social and Behavioral Sciences*, *160*, 607–614. <https://doi.org/10.1016/j.sbspro.2014.12.174>.
6. Fu, H., & Zhou, Y. (2011). The traffic accident prediction based on neural network. In *2011 Second International Conference on Digital Manufacturing & Automation*. <https://doi.org/10.1109/icdma.2011.331>.
7. Chong, M., Abraham, A., & Paprzycki, M. (2005). Traffic accident analysis using machine learning paradigms. *Informatica*, *29*, 89–98.
8. Geetha, K., & Vaishnavi, C. (2015). Analysis on traffic accident injury level using classification. *International Journal of Advanced Research in Computer Science and Software Engineering*, *5*(2), 953–956.
9. Sohn, S. Y., & Lee, S. H. (2003). Data fusion, ensemble and clustering to improve the classification accuracy for the severity of road traffic accidents in Korea. *Safety Science*, *41*(1), 1–14. [https://doi.org/10.1016/s0925-7535\(01\)00032-7](https://doi.org/10.1016/s0925-7535(01)00032-7).
10. Vasavi, S. (2016). A survey on extracting hidden patterns within road accident data using machine learning techniques. *Communications on Applied Electronics*, *6*(4), 1–6. <https://doi.org/10.5120/cae2016652455>.
11. Razzaq, S., Riaz, F., Mehmood, T., & Ratyal, N. I. (2016). Multi-factors based road accident prevention system. In *2016 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*. <https://doi.org/10.1109/icecube.2016.7495221>.
12. Kumar, S., & Toshniwal, D. (2015). Analysing road accident data using association rule mining. In *2015 International Conference on Computing, Communication and Security (ICCCS)*. <https://doi.org/10.1109/iccscs.2015.7374211>.
13. www.punjabroadsafety.org. Last Accessed January 11, 2018.

Multi-language Handwritten Recognition in DWT Accuracy Analysis



T. P. Umadevi and A. Murugan

Abstract Discovery of gender orientation from penmanship of an individual shows an intriguing exploration issue with applications in measurable record test, essayist recognizable proof and mental investigations. This paper displays a compelling strategy to anticipate the gender orientation of a person from disconnected pictures of penmanship. The proposed strategy depends on a worldwide approach that thinks about composing pictures as surfaces. Each written by hand picture is changed over into a finished picture which is disintegrated into a progression of wavelet sub-groups at various levels. The wavelet sub-bands are then stretched out into information successions. Every datum succession is quantized to create a Discrete Wavelets Transform (DWT) that produces extraction. These highlights are utilized to prepare two classifiers, numerous occurrence learning, and multiclass bolster vector machine to separate among male and female handwriting recognition compositions. The execution of the proposed framework was assessed on two databases, HWSC and TST1, inside various testing exploratory situations and acknowledged arrangement rates of up to 94.07%.

Keywords Disconnected penmanship examination · Gender location · Surface investigation · Wavelet sub-band · Emblematic progression

1 Introduction

Examination of penmanship and hand-drawn shapes is a delightful area of research for clinicians, report analysts, paleographers, graphologists, scientific investigators, and pc science analysts. While the guide assessment of penmanship has been an activity for a long time, the mechanized assessment appreciates a restored look

T. P. Umadevi (✉)

Department of Computer Science, JBAS College for Women, Teynampet, India
e-mail: umashiva06@gmail.com

A. Murugan

Department of Computer Science, Dr. Ambedkar Government Arts College (Autonomous),
Vysarpadi, Chennai 600039, Tamil Nadu, India

© Springer Nature Singapore Pte Ltd. 2019

N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,

https://doi.org/10.1007/978-981-13-5953-8_18

into pastime because of the very most recent innovative and algorithmic headways in extraordinary zones of workstation sciences. In spite of the fact that penmanship acknowledgment remains the greatest programming of automated penmanship examination, a wide range of fascinating applications have likewise been explored. These incorporate order of composing styles, watchword perceiving in written by hand documents and, recognizable proof and confirmation of essayists from their transcribed specimens. A nearly related inconvenience to make recognizable proof is the order of shopper socioeconomics from penmanship. Distinguishing proof of statistic properties which incorporates gender orientation, handedness, age, race and soon have been researched in the writing [1, 2]. Additionally, a couple of mental research propose the presence of connection between's penmanship and particular traits two of the character of the creator [3–8]. The cases of these graph logical considers, nonetheless, stay subjective and are however to be approved by method for trial ponders. As unfavorable to private traits which tend to be subjective, statistic properties are objective and the order of these qualities from penmanship can be tentatively approved through measured outcomes. Among an assortment of statistic characteristics, a fairly solid relationship has been tried among gender and parts of penmanship in a wide assortment of studies [9–14]. Mental research advocate that the male compositions, by and large, tend to be more prominent 'spiky', 'rushed' and 'chaotic' while the lady compositions are likely to be additional 'embellishing', 'homogenous' and 'reliable' [13–15]. Programmed discovery of gender construct absolutely in light of these highlights makes the issue of our examination.

Programmed gender orientation forecast from transcribed examples can be inferred in a wide assortment of intriguing applications not exclusively for clinicians however furthermore for criminological pros furthermore, record inspectors. It can fill in as a sifting step allowing experts to diminish the inquiry space and concentrate on a subset of compositions originating from a particular gender orientation gathering. In addition, gender orientation characterization can moreover prompt expanded impacts in many creators distinguishing proof and check purposes [16]. Notwithstanding these used parts of gender identification, the find out about of connection among gender orientation and particular written work signs offers a fascinating inconvenience of crucial research. Gender orientation arrangement is deliberately connected with the creator recognizable proof issue. A fundamental assessment of the gender orientation grouping procedures principally in light of penmanship styles uncovers that in many examples features utilized for creator recognizable proof have been adjusted for gender order [1, 2, 16]. This paper tends to be computerized gender orientation forecast from disconnected manually written pictures. We advocate a novel brilliant approach that thinks about works as surfaces what's more, utilizes wavelets to symbolize male and lady works. An outline of the means stressed in the proposed strategy is displayed in Fig. 1. The proposed work include extraction, relies upon wavelet change the utilization of representative dynamic separating (SDF). In a current report, SDF-based capacity extraction from the time succession of robot development conduct has been completed by means of Hassaine et al. [17]. Moreover, SDF-based capacity extraction from time arrangement data has been proposed through Al-Maadeed et al. [18] for target recognition and order in outskirts locales. To

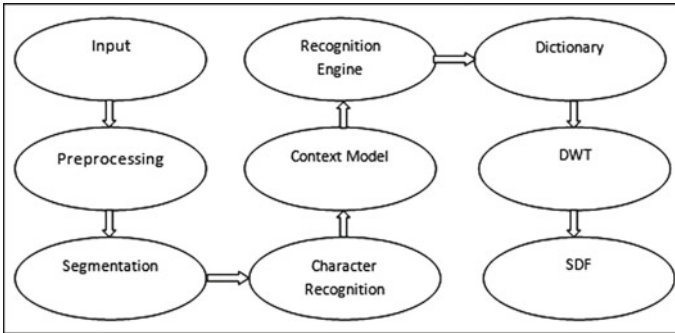


Fig. 1 Hand written proposed framework DWT

the best of our insight, this is the principal endeavor of abusing emblematic dynamic separating to imply gender from penmanship and speaks to the essential commitment of this work. In our component extraction technique, the wavelet sub-band is stretched out into a data grouping that is symbolized to develop Discrete Wavelets Change (DWT) which produces the component vectors.

These component vectors are marked by methods for two k-closest neighbors and Multi-bolster vector figuring gadget (SVM). The proposed technique is assessed on the normal HWSC also, TST1 databases with composing tests in English, Cursive, and Arabic.

2 Related Work

In this area, we initially talk about the connection among penmanship and gender as perceived in the manual assessment of penmanship. Gender orientation and Penmanship Various investigations [12–14] have inferred that gender can be anticipated from penmanship through the precision of expectation may shift. It has likewise been watched that people who every now and again manage manually written pictures end up plainly gifted at separating the compositions of the two gender orientation classes [1]. These distinctions are for the most part credited to the distinctions in engine coordination among the two sexes. The designs explanatory examination of Pratikakis [19] considered the impact of gender on penmanship. Looking at male and female penmanship styles is a goal reconsideration of Lesters graphology discoveries that there are contrasts between them works originating from various gender orientations [9]. As talked about before, when all is said in done, female works have a tendency to be slick, even, efficient, adjusted, little and symmetrical.

Male compositions, then again, are for the most part portrayed by properties like rough, uneven, messier, inclined, and slanting [13]. An arrangement of point by point separating factors among male and female compositions can be found in [20].

Eames did an examination utilizing tests from 12 college understudies to decide if penmanship has any relationship with the evaluations of the understudies and whether the gender of an understudy could be perceived through his or her penmanship. The consequences of the examination demonstrated that evaluations of understudies were most certainly not impacted by the nature of penmanship. In any case, as a rule, it was conceivable to decide the gender orientation of the understudy. In another investigation, Agius et al. [14] performed a few analyses with male and female understudies and revealed a 75% right distinguishing proof of gender orientation even with exceptionally constrained written work tests. The creator closed that by and large, one letter or image is frequently enough for distinguishing proof of gender orientation. In a different report on manual examination of penmanship by Chahi et al. [12], the writers investigated English and Urdu penmanship tests gathered from 30 distinct people. Every benefactor duplicated a 50-word entry (both in Urdu and English) and the literary substance of the section was same for every one of the people. These examples were then dissected by 25 inspectors for recognizable proof of gender orientation. It was inferred that gender orientation distinguishing proof is similarly dependable for content in both the dialects with a normal exactness of 68%.

The fascinating discoveries of analysts on the connection among penmanship and gender orientation enlivened the PC researchers to apply picture investigation and example arrangement methods to computerize this investigation. The accompanying subsection shows a diagram of the automated grouping of gender orientation from penmanship.

SVM proposed a gender and handedness order system from internet penmanship that, notwithstanding the disconnected portrayal of character shapes permit misusing the fleeting data of composing too. The creators extricate an arrangement of 29 highlights from the online-specimens and its disconnected portrayal what's more, utilize bolster vector machine and Gaussian blend models for arrangement. Exactnesses of up to 67.06 and 84.66% are accounted for gender orientation and handedness arrangement separately. A different examination of disconnected and online works uncovered that the grouping rates on online penmanship styles are superior to those on disconnected pictures of composing. In [21], the writers misuse highlights like Fourier descriptors and shape data to distinguish gender orientation from penmanship. The creators be that as it may, do not present any measured outcomes and examine the estimations of these parameters for male and female compositions.

3 Problem Analysis

While DWT technology can be effective in converting handwritten or typed characters, it does not give as high accuracy as of Document Form for reading data, where users are actually marking forms.

Additional workload to data collectors SDF has severe limitations when it comes to human handwriting. Characters must be handprinted with separate characters in boxes.

Flow of work.

3.1 Dataset

The trial investigation of the proposed framework is done on two surely understood penmanship databases, the Qatar College Essayist Recognizable proof (HWSC) database furthermore, Multi-content Manually written Database (TST1).

3.2 HWSC Database

The HWSC database contains composing tests contributed by 1017 journalists. Every author gave four tests, two in English and two in Arabic. Page 1 and Page 3 of all scholars contain a content from essayist's own creative energy in Arabic and English individually. In like manner, page 2 (Arabic) and page 4 (English) of every author contains the same literary substance. To permit significant examination of the execution of our framework with different investigations, we completed the analyses on composing tests of 475 journalists. The preparation set incorporates composing tests of 282 authors while the test set contains 193 authors.

3.3 TST1 Database

The TST1 database contains written by hand tests gathered from an aggregate of 84 journalists. Every essayist was required to compose 12 pages, six each in Cursive and Arabic and each page had the same printed content for all essayists. In the greater part of the assessments, the preparation and test sets involve composing tests of 42 authors each (Fig. 2).

4 Letter Style

4.1 Offline and Online

1. Online—captured by pen-like devices, the input format is a two-dimensional signal of pixel locations as a function of time ($x(t)$, $y(t)$).
2. Offline—captured by scanning devices, the input format is a two-dimensional image of grayscale colors as a function of location $I(m*n)$. Strokes have significant width (Figs. 3 and 4).

Fig. 2 DWT using data base analysis

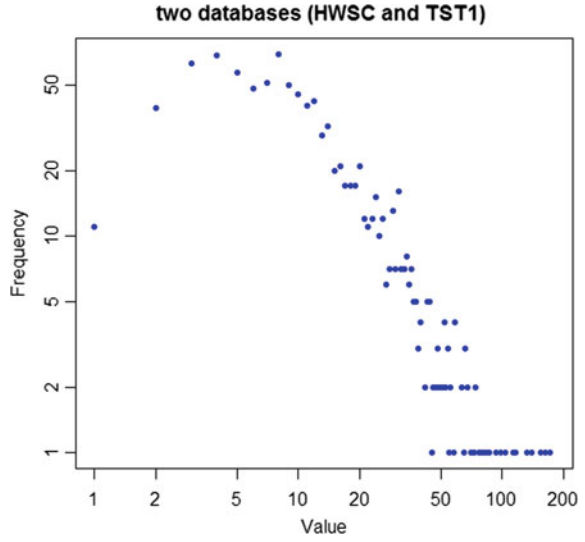


Fig. 3 Handwritten offline and online pre processing

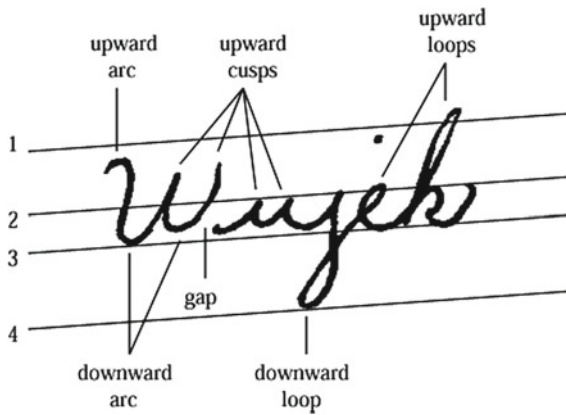
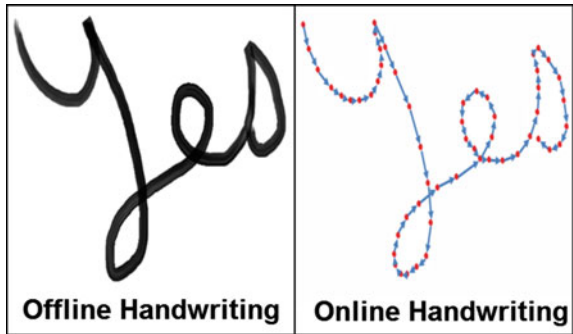


Fig. 4 Pattern recognition and machine learning conflicts

Table 1 Database in documents framework analysis

Database	HWSC	TST1
Text	Form English	Form English
Number of writers (training test)	475 (282 + 193)	84 (42 + 42)
Distribution of writers	221 male & 254 female	37 male & 47 female
Samples per writer	4 (2 Arabic, 2 English)	12 (6 Arabic, 6 Cursive)
Total samples	1900 (4 × 475)	1008 (12 × 84)
Size of texture block	256 × 256	256 × 256
Texture blocks from n pages	$n = 4, 52 (11 + 15 + 11 + 15) (p_1 + p_2 + \dots + p_n)$	$n = 12, 180 (15 + 15 + \dots + 15)$
Total number of texture blocks	24,700 (52 × 475)	15,120 (180 × 84)

Table 2 DWT background on cursive handwriting

Letter and number	Pos	Orientation	Angle
Down cusp	3.0	-90°	Left
Up loop	2.0	90°	Right
Down arc	2.75	180°	Two hand level (Left, right)

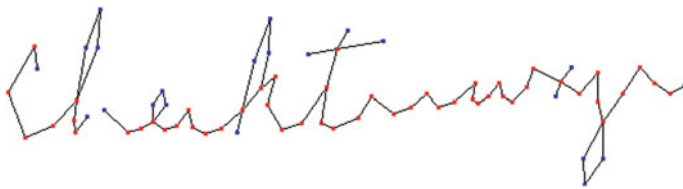


Fig. 5 Documents form offline cursive word indication representation

It ought to be noticed that in every single test setting, the preparation and test sets do not contain any written work test of a similar author with the goal that the issue compares to gender orientation grouping and not essayist recognizable proof. The division of scholars in preparing and test sets alongside other data is exhibited in Table 1 (Table 2).

This area displays the subtle elements of the proposed approach including pre-preparing, highlight extraction, and arrangement. The proposed portrayal of gender orientation from pictures of penmanship principally depends on Wavelet change and system Dynamic Separating (SDF). We display a concise outline of these ideas in the accompanying before talking about their application to our specific issue (Fig. 5).

5 Handprinted Implementation DWT

Wavelets are by and large named as a numerical magnifying lens in flag and picture handling. Portraying time-subordinate information utilizing wavelet premise brings about an intense portrayal of data that is at the same time confined in time and recurrence spaces. This is as opposed to the Fourier portrayal where it is not conceivable to relate particular frequencies to particular interims of time. Wavelets are particularly effective with regards to examination of signs and pictures with discontinuities and sharp spikes and discover applications in a wide assortment of issues in Material science, Arithmetic, and Electrical Designing, supplanting the traditional Fourier changes much of the time. Run of the mill uses of wavelets in flag preparing incorporate picture pressure, picture denoising, discourse acknowledgment, EEG, EMG and ECG examinations and so on. An inferable capacity $\Psi \subseteq L2(R)$ is

- Wavelet: finite interval function with zero mean (suited to analysis transient signals)
- Utilize the combination of wavelets (basis function) to analyze arbitrary function
- Mother wavelet $\Psi(t)$: by scaling and translating the mother wavelet, we can obtain the rest of the function for the transformation (child wavelet, $\Psi_a, b(t)$)

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)$$

Performing the inner product of the child wavelet and $f(t)$, we can attain the wavelet coefficient

$$w_{a,b} = \langle \psi_{a,b}, f(t) \rangle = \int_{-\infty}^{\infty} \psi_{a,b} f(t) dt$$

We can reconstruct $f(t)$ with the wavelet coefficient by

$$f(t) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} w_{a,b} \psi_{a,b}(t) \frac{da db}{a^2}$$

Huge estimations of the widening parameter $|s|$ compare to little frequencies while the little esteems relate to high-frequencies. Changing the interpretation parameter μ moves the time limitation focus. Each $\Psi_{\mu,s}(t)$ is restricted around $t = \mu$. Thus, an impeccable time recurrence depiction of capacity f can be gotten utilizing the wavelet change. For our situation, the written by hand pictures are disintegrated into a progression of wavelet sub-groups utilizing the Mallat calculation [40]. This deterioration is like the one displayed in SDF for author distinguishing proof.

Symbolic Dynamic Sifting (SDF) This area quickly shows the basic ideas of Representative Dynamic Sifting (SDF) presented by Beam [22] for extraction of highlights. In our execution, we have connected the SDF calculation introduced to remove gender particular highlights from the written work pictures. The key advances engaged with SDF-based component extraction are examined in the accompanying.

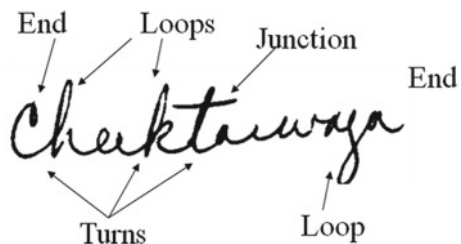
5.1 Symbolization/Quantization of Information

The initial phase in SDF is the quantization of information (wavelet changed pictures for our situation). For symbolization, the given information arrangement is isolated into a limited number of cells. Every cell is then allocated a novel name or image and the aggregate number of remarkable images is the same as the aggregate number of cells. Accepting the image letter set to be $A = \{a_1, a_2, a_3, a_4\}$ with $\text{card}(A) = 4$, the information succession has been partitioned into four parcels on the y-hub. These districts are totally unrelated and additionally comprehensive (the total information profile is secured). Each of the four districts is allotted a mark from the letters in order A . For an information grouping under thought, the flag an incentive at a given point in time is doled out the image comparing to the phone in which it is found. This permits speaking to the information grouping by a limited series of images regularly named as the image piece. Additionally points of interest on the symbolization of information can be found in DWT (Fig. 6).

For dividing of information, systems like uniform parceling (UP) most extreme entropy apportioning (MEP) have been examined. Uniform dividing, as the name recommends, creates cells of equivalent size. Most extreme entropy apportioning, despite what might be expected, endeavors to boost the entropy of the created images by guaranteeing that every cell contains roughly a similar number of information focuses. Thusly, the cell measure is little in data rich locales while it is extensive in scanty areas. In the two cases, the measure of the letter set is picked as a component of the information under examination and the objective framework execution.

$$\Phi_W = \sum_{i=1}^C \sum_{j=1}^{N_i} (\mathbf{v}_j^i - \boldsymbol{\mu}_i)(\mathbf{v}_j^i - \boldsymbol{\mu}_i)^T$$

Fig. 6 Loop identification reduce noisy in SDF



v_{ji} : j -th data vector of class i .

Given a projection matrix W (of size n by m) and its linear transformation, the between-class scatter in the projection space is

$$\mathbf{p} = \mathbf{W}^T \mathbf{v}$$

$$\begin{aligned} \Psi_B &= \sum_{i=1}^C N_i (\boldsymbol{\mu}'_i - \boldsymbol{\mu}') (\boldsymbol{\mu}'_i - \boldsymbol{\mu}')^T \\ &= \sum_{i=1}^C N_i (\mathbf{W}^T \boldsymbol{\mu}_i - \mathbf{W}^T \boldsymbol{\mu}) (\mathbf{W}^T \boldsymbol{\mu}_i - \mathbf{W}^T \boldsymbol{\mu})^T \\ &= \sum_{i=1}^C N_i (\mathbf{W}^T \boldsymbol{\mu}_i - \mathbf{W}^T \boldsymbol{\mu}) (\boldsymbol{\mu}_i^T \mathbf{W} - \boldsymbol{\mu}^T \mathbf{W}) \\ &= \sum_{i=1}^C \mathbf{W}^T N_i (\boldsymbol{\mu}_i - \boldsymbol{\mu}) (\boldsymbol{\mu}_i^T - \boldsymbol{\mu}^T) \mathbf{W} \\ &= \mathbf{W}^T \left(\sum_{i=1}^C N_i (\boldsymbol{\mu}_i - \boldsymbol{\mu}) (\boldsymbol{\mu}_i - \boldsymbol{\mu})^T \right) \mathbf{W} \\ &= \mathbf{W}^T \Phi_B \mathbf{W} \end{aligned}$$

5.2 Discrete Wavelets Transform (DWT)

The development of Discrete Wavelets Transform (DWT) depends on the supposition that the image age process can be demonstrated as a Markov chain, named as the D-Markov machine [19]. Advertisement arrange Markov chain is a stochastic procedure where the likelihood of event of an image is a capacity The states in DWT speak to mixes of words in the grouping of images while the edges compare to progress between various states. In our work, we take $D = 1$, subsequently, the quantity of states are the same as the quantity of images in the letters in order. The arrangement of conceivable states is given by $S = \{s_1, s_2, \dots, s_N\}$ with N being the aggregate the quantity of states. The likelihood of change from state s_i to state s_j would then be able to be characterized as takes after










$$a = a_0^{-m}, b = nb_0 a_0^{-m} \quad m, n \in \mathbb{Z},$$

where $N(s_i, s_j)$ is the aggregate number of occasions when s_j seems adjoining s_i .

If $a_0 = 2, b_0 = 1$, the set of the wavelet

$$\psi_{m,n}(t) = a_0^{m/2} \psi(a_0^m t - nb_0) \quad m, n \in \mathbb{Z}$$

Table 3 DWT recognition parameters SDF

Meaningful straight lines		Meaningful Arcs	
Horizontal line		C like	
Vertical line		D like	
Positive slanted		U like	
Negative slanted		A like	
		O like	

$$\psi_{m,n}(t) = 2^{m/2} \psi(2^m t - n) S_i \times S_j$$

When all the change probabilities $P(s_j|s_i), \forall s_j, s_i \in S$ have been assessed, they have gathered the $N \times N$ state progress framework. Get the maximum coefficient = 26 (Table 3).

Pseudo code

```

Initialize HWSC = 15, TST1 = 10
int[] list = new int[HWSC]
for (int index = 0; index < HWSC; index++)
Round Toward zero (NIST)
if sign=positive
Else if any bits to the right of the result DWT=1
Else if sign=negative
Compute Round to nearest English
Else if sign=neutral positive
FLD X=SDF (st(0) = X)
FLD Y (st(0) = Y, st(1) = X)
FLD U (st(0) = U, st(1) = X*Y)
FLD V (st(0) = V, st(1) = U, st(2) = X*Y)
FDIV= TST1 ;st(0) = U/V, st(1) = X*Y
FSUB(SVM)= ;st(0) = X*Y - U / V
FSTP Z= DWT ;Z= result, st(0) = empty
while(!kbhit())
for(int j=0;j<50;j++)
*(p+160*j+i+1)=2
*(p+160*j+i)=a[random(15)]
for(j=0;j<25;j++)
c=*(p+160*j+i)
d=*(p+160*(j+1)+i)
end while *(p+160*(j+1)+i)=c
end if *(p+i)=
    
```

Regardless of the way transcribed content is masterminded on the page, the associated parts in the composition are modified into another space keeping the first inclination yet decreasing the spaces between lines of content and segments. This creates surface pictures which save the overall look and feel of the written work and permits utilizing a worldwide approach staying away from the intricacy of division. The significant strides in this procedure are recorded in the accompanying.

1. The picture is binarized utilizing worldwide thresholding and the associated parts are separated utilizing 8-network.
2. Small parts which are probably going to relate to accentuation stamps and com-motion are expelled utilizing region-based separating.
3. The bouncing boxes of the rest of the segments are then used to extricate the relating parts from the grayscale picture.
4. Components in the primary line of the first picture are adjusted in another picture utilizing the focal point of mass of the bouncing box. The inclination of the content lines, in this way, is standardized.

After filling the primary line, the normal stature of the parts is figured and this esteem is utilized to characterize the y-organize of the following line, which is given as takes after

$$\Phi_B = \sum_{i=1}^C N_i (\mu_i - \mu)(\mu_i - \mu)^T$$

C number of classes

N_i number of data vectors in class i

μ_i mean vector of class i and μ : mean vector

6 Results and Implementation

We now show the consequences of the examinations to approve the adequacy of the proposed method in portraying gender from penmanship. To begin with, we talk about the grouping rates on the two databases (HWSC and TST1) and later present the consequences of various fascinating examinations. Notwithstanding the proposed highlights, we likewise assess best in class essayist distinguishing proof highlights for gender orientation order on the two databases. These incorporate introduction and bend highlights, fractal measurement, neighborhood paired examples (LBP) and auto-backward (AR) coefficients. Moreover, to think about the effect of line pressing, we figure the grouping rates without age of finished pictures (Figs. 7, 8 and 9).

Comparison study

We have compared our proposed method of implementation with various other methods, the below table depicts that our proposed method has more accuracy than other compared methods.

Author	Methods	Accuracy (%)
G. Huang, D. Zhang, X. Zheng and X. Zhu	DTW	90
H. Ding	RFIPad	91
N. Sharma, B. Kumar and V. Singh,	SVM	92
Gaganjot Kaur, Monika Aggarwal	Levenberg–Marquardt	93
Proposed method	DWT	94.07

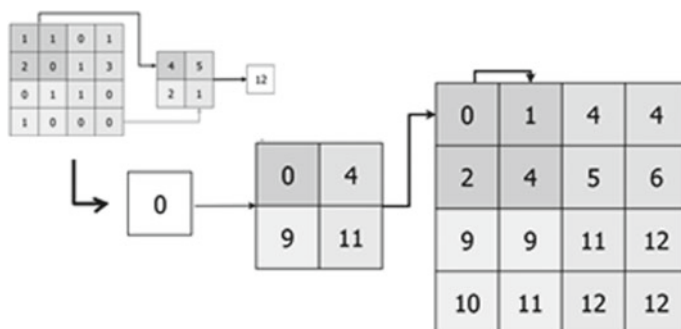


Fig. 7 Origin pixel matrix base the recognition technique

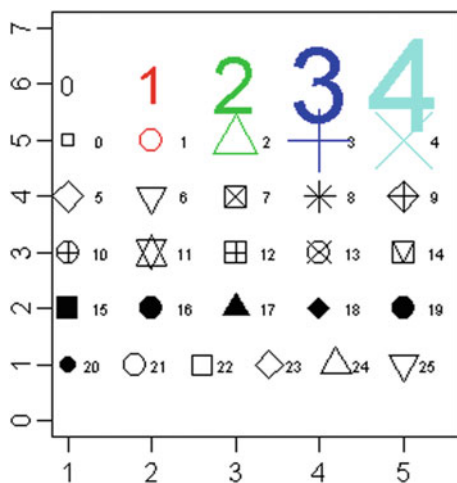


Fig. 8 KNN designed for DWT of input and maximum symbol separation edification

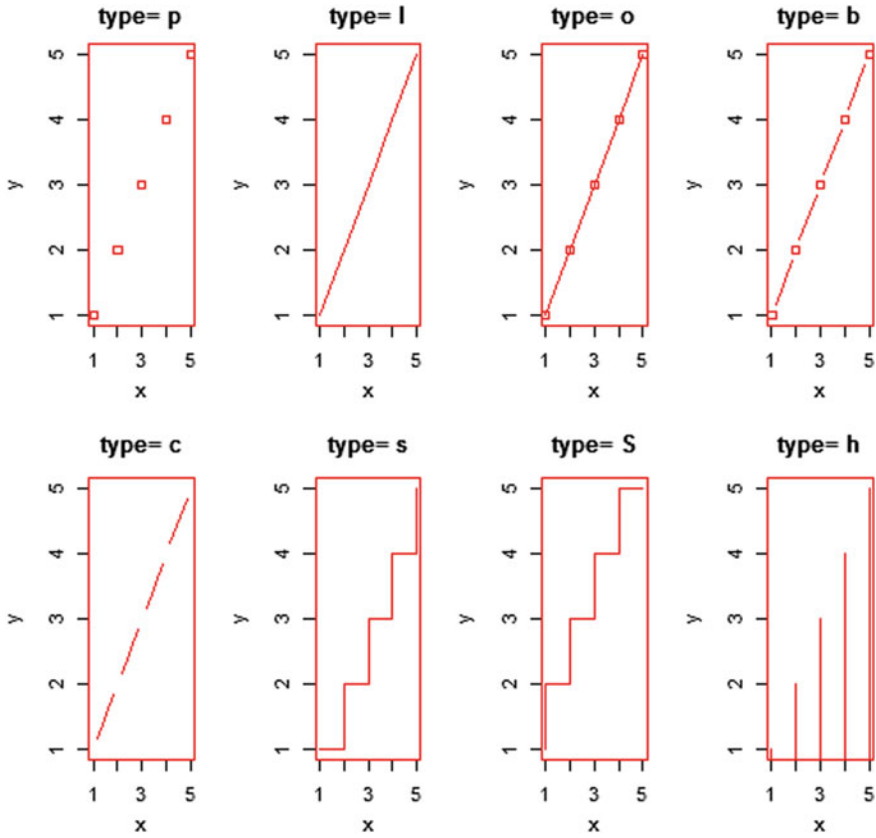


Fig. 9 Order rates as an element of SDF letter set size utilizing KNN classifier

7 Conclusion and Future Works

We displayed a novel approach for gender orientation location of essayists from pictures of penmanship styles. The method depends on a worldwide surface based approach by thinking about each composing as a surface. The transcribed pictures are utilized to create surface pieces which are disintegrated into a progression of wavelet sub-bands. Each of the sub-groups is at that point stretched out into an information arrangement that is symbolized to build a probability state automata (DWT). The DWT is then used to create the element vector describing gender of the essayist of a given penmanship test. For grouping, counterfeit neural systems and bolster vector machines are utilized. The procedure was assessed on two databases containing transcribed specimens in English, Cursive, and Arabic. A progression of examinations in various testing situations demonstrated and acknowledged arrangement rates of up to 94.07%.

For our further examinations on this issue, we intend to upgrade the list of capabilities and apply a component choice method to locate the ideal arrangement of highlights for this issue. We additionally plan to incorporate the location of other statistic properties of scholars from manually written pictures. These may incorporate handedness (left or right), age gathering and race and so on. Connection among penmanship and individual and scholarly properties of essayists can likewise be contemplated.

References

1. Almusaly, I., & Metoyer, R. (2015). A syntax-directed keyboard extension for writing source code on touchscreen devices. In *2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (pp. 195–202). IEEE.
2. Alonso, M. A. P. (2015). Metacognition and sensorimotor components underlying the process of handwriting and keyboarding and their impact on learning. An analysis from the perspective of embodied psychology. *Procedia-Social and Behavioral Sciences*, 176, 263–269.
3. Doetsch, P., & Ney, H, et al. (2013). Improvements in rwth's system for off-line handwriting recognition. In *2013 12th International Conference on Document Analysis and Recognition* (pp. 935–939). IEEE.
4. Keyzers, D., Deselaers, T., Rowley, H. A, Wang, L.L., & Carbone, V. (2016). Multi-language online handwriting recognition.
5. Poznanski, A., & Wolf, L. (2016). Cnn-n-gram for handwriting word recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2305–2314).
6. Capobianco, S., & Simone, M. (2017). Deep neural networks for record counting in historical handwritten documents, In *Pattern recognition letters*. ISSN 0167-8655.
7. Kumar, G., & Govindaraju, V. (2017). Bayesian background models for keyword spotting in handwritten documents. In *Pattern recognition* (Vol. 64, pp. 84–91).
8. Yagoubi, M. R., Serir, A., & Beghdadi, A. (2016). Joint enhancement-compression of handwritten document images through DjVu encoder. *Journal of Visual Communication and Image Representation*, 41, 324–338. ISSN 1047-3203.
9. Akbari, Y., Jalili, M. J., Sadri, J., Nouri, K., Siddiqi, I., & Djeddi, C. (2018). A novel database for automatic processing of Persian handwritten bank checks. *Pattern Recognition*, 74, 253–265. ISSN 0031-3203.
10. Vo, Q. N., Kim, S. H., Yang, H. J., & Lee, G. (2018) Binarization of degraded document images based on hierarchical deep supervised network. *Pattern Recognition*, 74, 568–586. ISSN 0031-3203.
11. Jia, F., Shi, C., He, K., Wang, C., & Xiao, B. (2018). Degraded document image binarization using structural symmetry of strokes. *Pattern Recognition*, 74, 225–240. ISSN 0031-3203.
12. Chahi, A., El khadiri, I., El merabet, Y., Ruichek, Y., & Touahni, R. (2018). Block wise local binary count for off-Line text-independent writer identification. *Expert Systems with Applications*, 93, 1–14. ISSN 0957-4174.
13. Mondal, T., Ragot, N., Ramel, J.-y., & Pal, U. (2018). Comparative study of conventional time series matching techniques for word spotting. *Pattern Recognition*, 73, 47–64. ISSN0031-3203.
14. Agius, A., Morelato, M., Moret, S., Chadwick, S., Jones, K., Epple, R., et al. (2018). Using handwriting to infer a writer's country of origin for forensic intelligence purposes. *Forensic Science International*, 282, 144–156. ISSN0379-0738.
15. Siddiqi, I., Djeddi, C., Raza, A., & Souici-Meslati, A. (2014). Automatic analysis of handwriting for gender classification. *Pattern Analysis and Application*.
16. Liwicki, M., Schlapbach, A., & Bunke, H. (2011). Automatic gender detection using online and off-line information. *Pattern Analysis and Application*, 14, 87–92.

17. Hassaine, A., Al-Maadeed, S., Aljaam, J., & Jaoua, A. (2013). ICDAR 2013 competition on gender prediction from handwriting. In *Proceedings of IEEE 12th International Conference on Document Analysis and Recognition* (pp. 1417–1421).
18. Al-Maadeed, S., Ayouby, W., Hassaine, A., Aljaam, & Quwi, J.M. (2012). An Arabic and English handwriting dataset for offline writer identification. In *Proceedings of 13th International Conference on Frontiers in Handwriting Recognition* (pp. 746–751).
19. Pratikakis, I., Gatos, B., & Ntirogiannis, K.: ICFHR 2012 competition on handwritten document image binarization (H-DIBCO 2012). In *2012 International Conference on Frontiers in Handwriting Recognition (ICFHR)* (pp. 817–822). IEEE.
20. Pratikakis, I., Zagoris, K., Barlas, G., & Gatos, B. (2016). ICFHR2016 handwritten document image binarization contest (H-DIBCO 2016). In *15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, (pp. 619–623).
21. Huang, G., Zhang, D., & Zheng, X. (2010). An EMG-based handwriting recognition through dynamic time warping 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology. *Buenos Aires, 2010*, 4902–4905.
22. Sokic, E., Salihbegovic, A., & Ahic-Djokic, M. (2012). Analysis of offline handwritten text samples of different gender using shape descriptors. In *Proceedings of IX International Symposium on Telecommunications (BIHTEL)* (pp. 1–6).

A Novel H-∞ Filter Based Indicator for Health Monitoring of Components in a Smart Grid



E. Ranjini Warriar, P. V. Sunil Nag and C. Santhosh Kumar

Abstract Health monitoring of a smart grid is very important to ensure reliability of the grid. This can be achieved by developing various indicators for the components of the grid. These indicators are very powerful if they are model based as they can be used in real time without need for extra hardware provided the model of the systems and the model parameter values are available. This work presents a novel H∞ filter based fault indicator. The fault chosen here is the stator interturn fault of a wound rotor synchronous generator. It will be proved that (1) the indicator sensitive to fault and is insensitive to other kinds of spurious effects like load imbalance. (2) The indicator can be used to find the magnitude of fault. (3) The indicator can function irrespective of the type of uncertainties assumed in modelling the system. As far as the knowledge of the author goes this is the first time H-∞ filter based indicators are used for stator interturn fault of a wound rotor synchronous generator.

Keywords H-∞ filter · Fault indicators · Model based · Wound rotor synchronous generators · Smart grid

1 Introduction

Research in Smart grids is being actively pursued in various countries to cater to the escalating and variable demand for energy. One of the functions of a smart grid is to monitor the health of its components by means of health/fault indicators [1]. These indicators should flag incipient faults. They should provide rich information like the type of malfunction/fault magnitude and location of fault. In an advanced system these indicators can be used to obtain the remaining useful life of the component. Various machine intelligence algorithms have been proposed to derive such indicators [2]. These can be broadly classified as model-based- and data-based indicators. Data-based indicators [3, 4] use information-rich historical data from the systems under

E. Ranjini Warriar · P. V. Sunil Nag (✉) · C. Santhosh Kumar
Department of Electronics and Communication Engineering, Amrita School of Engineering,
Amrita Vishwa Vidyapeetham, Coimbatore 641112, India
e-mail: pv_sunil@cb.amrita.edu

consideration to generate the indicators. Model-based indicators use the model of the system for the indicators. This approach uses the physical principles governing the system to develop a model and expressed in the form of mathematical equations. State estimation is one of the popular methods for generating indicators in the model-based category.

The agent which performs this estimation can be categorized as state estimator. Several approaches are existing for state estimation of systems [5]. Stochastic estimators like Kalman filters and its variants assume that the external noises, disturbances that affect the system and the “model-real system mismatches” follow Gaussian or normal distribution. H_∞ filters on the other hand solve the state estimation problem without assuming any statistics for the uncertainties. Hence are very powerful for generating fault indicators. These filters are also known as minimax filters.

There are various fields where H_∞ filters are utilized. In [6] this filter is used for parameter estimation in SLAM problem and in Li-Fe batteries [7], signal reconstruction [8] and target tracking [9]. Another such major area where this filter can be applied is to generate health/fault indicators. In [10, 11] these filters are used constructing fault indicators for diesel engines and induction motors respectively. Henry [12] explains how H_∞ filters are used generating fault indicators for micro strip satellite thrusters. The present work focuses on using H_∞ filters for developing fault indicators of wound rotor synchronous generator here after referred to as synchronous generators. This work can be extended to health monitoring of any component in a smart grid. Here synchronous generators are chosen as their functioning is critical in a typical power system. Further research in applying H_∞ filters for synchronous generator fault indicators is very rare.

Synchronous generators (SG) are the most common sources of Electrical Energy. It is the primary component in almost every power system/smart grid. Since the complexity and expense of power systems has gradually increased, there is less tolerance for performance degradation and safety issues. This leads to the necessity of detecting the faults in systems at the early stage and rectify it [11]. This paper deals with the most common and critical electrical fault known as stator interturn fault that can occur in Synchronous Generators. The residual signal, which the difference between the actual outputs from the system and the predicted outputs from the filter/state estimator acts as an indicator of fault [13].

The next section explains the methodology proposed for machine fault indicators using H_∞ filter. Modelling of synchronous generator is explained in brief in third section. Fourth section explains about the H_∞ filter used in this work. Results, analysis and comparisons with the EKF filter are presented in section five. Summary, conclusion and the future work constitutes the final section.

2 Methodology Proposed for Machine Fault Indicators

The schematic representation of the methodology used in this work is shown in Fig. 1. A synchronous generator contains a three phase stator and a DC excited rotor. The synchronous generator when connected to the grid is has a fixed terminal voltage at its stator terminals and is mechanically powered with a prime mover which is running at a constant speed. The inputs to the generator are the terminal voltages and the DC field excitation. The outputs are the three stator currents. The state estimator that is the H∞ filter takes the same input as the generator and provides the estimate of the stator currents along with a residual signal. The residual signal is obtained by subtracting the stator currents of the synchronous generator from the estimated stator currents of the H∞ filter. The residual has to be further processed to obtain the fault indicator which provides full information regarding the fault like time, magnitude and location of the fault.

Here is instead of an actual generator a fault injection capable MATLAB implementation of a generator [14] is used to get the data. The stator interturn short fault can be introduced in the simulated generator by changing a parameter μ which is the fraction of the total number of turns that are shorted in a given phase. In this particular implementation the fault can be simulated only in phase A. As mentioned earlier the H∞ filter being a model-based technique needs a model of the generator for its operation. The model used in this work is explained in the next section.

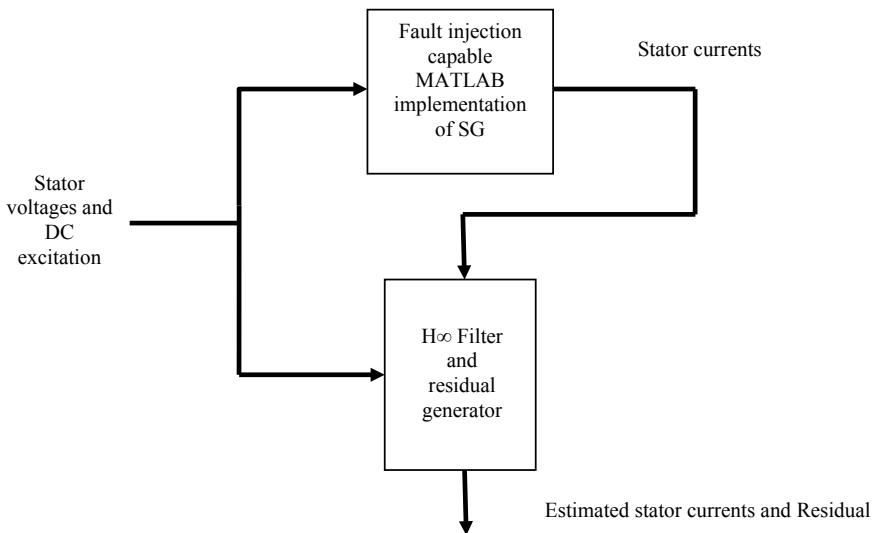


Fig. 1 Schematic of the proposed methodology

3 Modelling of Synchronous Generator

The SG as explained in [15] can be modelled as six magnetically coupled coils (three stator coils and three rotor coils). The coils on the rotor include a field winding and two damper windings. The state space model of the generator is given below.

$$\begin{aligned}
 X_e(k+1) &= A_m X_e(k) + B_m U(k) + w(k) \\
 Y_e(k) &= C_m X_e(k) + D_m U(k) + v(k) \\
 X_e(k) &= [X_1(k) X_2(k) \Sigma(k)]^T; Y_e(k) = X_1^T(k) \\
 \text{where} \\
 X_1(k) &= [i_{sd}(k) - \mu_a i_{fd}(k) i_{sq}(k) - \mu_b i_{fd}(k) i_{s0}(k) - \mu_c i_{f0}(k)] \\
 X_2(k) &= [i_{rfd}(k) i_{rkd}(k) i_{rkq}(k)]; \Sigma(k) = [\mu_a i_{fd} \mu_b i_{fd} \mu_c i_{f0}], \quad (1)
 \end{aligned}$$

where

$$A_m = \begin{bmatrix} A & O_{6 \times 3} \\ O_{3 \times 6} & I_{3 \times 3} \end{bmatrix}; B_m = \begin{bmatrix} B \\ O_{3 \times 6} \end{bmatrix}; C_m = [I_{3 \times 3} \ O_{3 \times 6}]; D_m = [O_{3 \times 3} \ O_{3 \times 3}];$$

O is a null matrix with the dimension indicated in the subscript and I is identity matrix.

$$\begin{aligned}
 A &= I + \begin{bmatrix} Z_1 & Z_2 \\ Z_3 & Z_4 \end{bmatrix}^{-1} \begin{bmatrix} Y_1 & Y_2 \\ Y_3 & Y_4 \end{bmatrix} * T_s; B = \begin{bmatrix} Z_1 & Z_2 \\ Z_3 & Z_4 \end{bmatrix}^{-1} * T_s \\
 Z_1 &= \begin{bmatrix} -(L_{md} + L_{al}) & 0 & 0 \\ 0 & -(L_{mq} + L_{al}) & 0 \\ 0 & 0 & -L_{al} \end{bmatrix}; Z_2 = \begin{bmatrix} L_{afd} & L_{akd} & 0 \\ 0 & 0 & L_{akq} \\ 0 & 0 & 0 \end{bmatrix}; \\
 Z_3 &= \begin{bmatrix} \frac{3}{2} L_{afd} & 0 & 0 \\ \frac{3}{2} L_{akd} & 0 & 0 \\ 0 & \frac{3}{2} L_{akq} & 0 \end{bmatrix}; Z_4 = \begin{bmatrix} L_{fdfd} & L_{fdkd} & 0 \\ L_{fdkd} & L_{kdkd} & 0 \\ 0 & 0 & L_{kqkq} \end{bmatrix}; \\
 Y_1 &= \begin{bmatrix} r_s & -(L_{mq} + L_{al}) & 0 \\ (L_{md} + L_{al}) & r_s & 0 \\ 0 & 0 & r_s \end{bmatrix}; Y_2 = \begin{bmatrix} 0 & 0 & L_{akq} \\ 0 & -L_{afd} & -L_{akd} \\ 0 & 0 & 0 \end{bmatrix}; \\
 Y_3 &= Z_3; Y_4 = Z_4 - [\text{diag}[r_{fd} \ r_{kd} \ r_{kq}]]; \\
 U(k) &= [v_{sd}(k) \ v_{sq}(k) \ v_{s0}(k) \ v_{rfd}(k) \ v_{rkd}(k) \ v_{rkq}(k)]^T
 \end{aligned}$$

The definition of various constants and their values used in this work is given in Table 1. As seen from the above state space model the state variables are not directly the stators currents so the estimates of the H_∞ filter have to be processed further to

Table 1 Definition of various constants and their values used in the model

Parameter	Value (pu)	Parameter	Value (pu)
r_s stator resistance	0.018	L_{mq} quadrature axis inductance	0.75
r_{fd} field winding resistance	0.015	L_{fdfd} field inductance	1.2
$r_{kd} = r_{kq}$ rotor d -axis and q -axis resistance	0.2	L_{kdkd} d -axis inductance	1.32
L_{md} direct axis inductance	1.00	L_{kqkq} d -axis inductance	1.26
L_{md} leakage inductance	0.15	$L_{afd}, L_{akd}, L_{akq}$ stator to rotor field, d -axis and q -axis mutual inductance	1.0
v_f field voltage	0.018	L_{fdkd} field to d -axis mutual inductance	1.0

obtain the actual estimates of the stator currents. The next section describes the H∞ filter algorithm.

4 H∞ Filter

H∞ filters [5] are applicable in cases where there is lack of clarity regarding the disturbances that acts on the system and the system uncertainties. In this work, H∞ filter is used for generating the fault indicator. The presence of ‘ γ ’ in filter equations highlights the major difference between the H∞ filter from the traditional Kalman Filters. The H∞ filter like the traditional Kalman filter contains two steps the prediction step and the correction step. The flow chart shown in Fig. 2 contains the algorithm.

The state space model of the Synchronous Generator as given in the previous section is nonlinear but the H∞ filter formulation considers the system as linear so the system is linearized at each time step. The states of the state space model are different from the natural stator currents. As per the schematic shown in Fig. 1 the natural stator currents are to be estimated by the H∞ filter, hence the last three variables of the state vector are added to the first three variables to obtain the stator currents in the dq frame and an inverse dq transformation is applied to the currents in the dq frame to obtain the natural stator current this is done at each time step more details can be found in [15]. The next section describes the experiments and analyses the results.

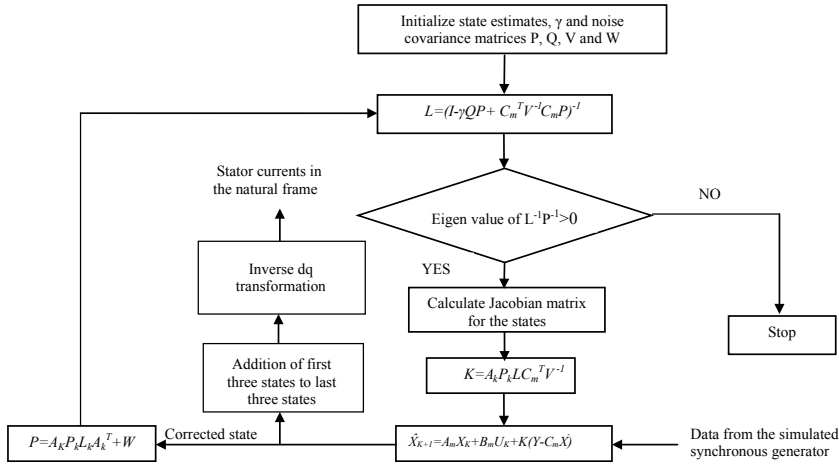


Fig. 2 H∞ filter algorithm used in this work

5 Simulation Experiment Results and Analysis

The simulated synchronous generator was run for a duration of 20 and at 10 s an inter turn fault with $\mu = 0.1$ (10% of the turns shorted) was introduced at phase A at 15 s and another interturn fault with $\mu = 0.2$ (20% of the turns shorted) was introduced in phase A. The H∞ filter was used to estimate the value of the three stator currents at each time step. The difference between the H∞ filter estimates and the actual stator currents from the simulation also called the residual was computed. The residual was further processed by calculating its moving variance and the RMS value of the resultant signal called the “RMS variance” was computed. This was done as the variance of the residual signal was sinusoidal in shape. This RMS variance is the fault indicator. Here the moving 10,000 time points (6 s) RMS variance of the residual was computed and plotted. The plot obtained is shown in Fig. 3. The plot was zoomed to the time interval of interest for better clarity. From the figure, it can be seen that at 10 and 15 s the RMS variance of phase A is different from the RMS variance of phase B and phase C. From this we can conclude that the fault is in Phase A. It can be concluded from the figure that the three RMS variance signals can indicate the location of the phase in which the fault has occurred.

Another simulation experiment was done to observe the effect of load imbalance similar to interturn fault on the residual. This is to make sure that the indicator does not respond to other spurious effects. The synchronous generator simulation was modified to incorporate a load imbalance and the H infinity filter was used to compute the fault indicator. Here a load imbalance of similar magnitude of the interturn fault was introduced at 10 and 15 s. The plot of 10,000 point RMS variance is shown in Fig. 4. By comparing Figs. 3 and 4, it is clear that the residual is sensitive only to interturn fault and not to the load imbalance. In the above experiments the

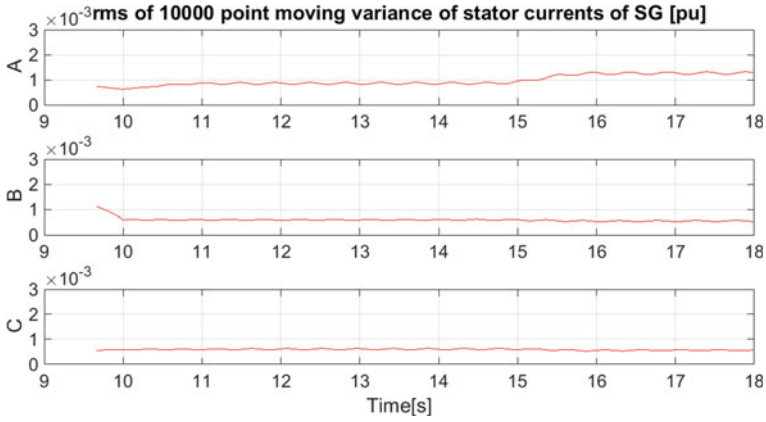


Fig. 3 10,000 point RMS variance of residuals with stator interturn fault

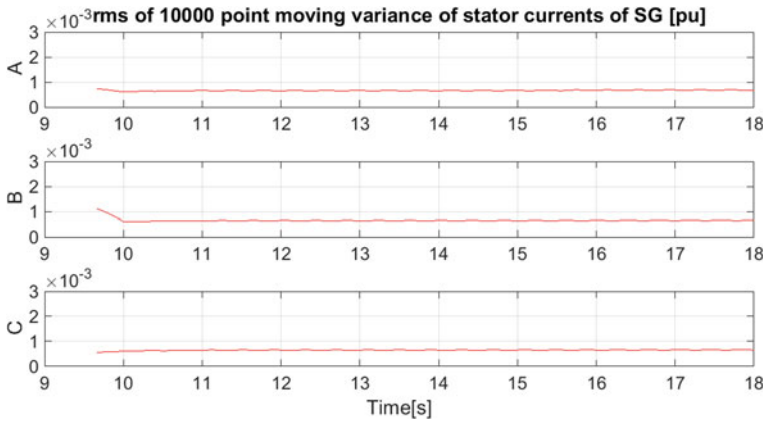


Fig. 4 1000 point moving variance of residuals with load imbalance

value of γ was taken as 0.01. It can also be seen that the magnitude of the RMS variance depends on the magnitude of the interturn fault. To use this phenomenon for the obtaining the fault magnitude an experiment was conducted with different values of fault parameters and the RMS variance values were computed. The results are shown in the Table 2. A straight line fit was obtained for the data in the table. The equation of the straight line is

$$R = 0.4534 \times 10^{-3} + \mu * 4.253 \times 10^{-3}, \tag{2}$$

where μ represents the percentage of fault parameter and R is the RMS variance. This equation can be used to compute the value of the fault parameter given the value of the RMS variance.

Table 2 Different fault parameter values with the corresponding RMS variance values

Fault parameter	Corresponding RMS variance (10^{-3})
0.1	0.8787
0.125	0.9641
0.15	1.055
0.175	1.134
0.2	1.304
0.225	1.365
0.25	1.569

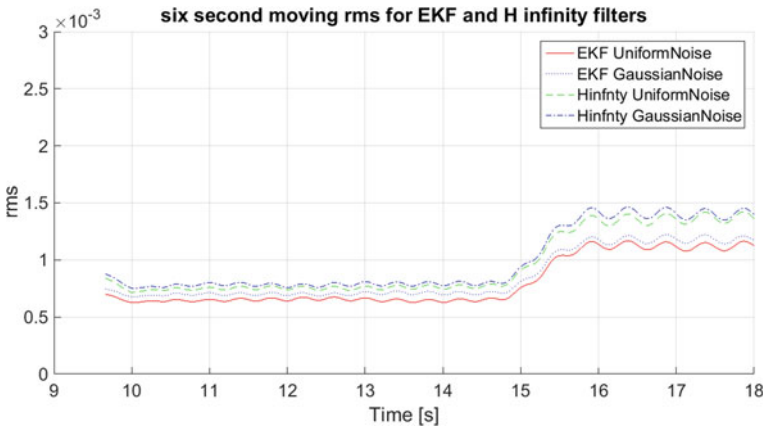


Fig. 5 Comparison of EKF and H_{∞} for different kinds of noise

An experiment was conducted to observe the effect of different types of noise on the performance of the H_{∞} filter for fault indication. A similar experiment was conducted using an Extended Kalman Filter (EKF) for the same purpose. The performance of both the filters was compared. The results in graphical form are presented in Fig. 5. The figure shows the RMS variance values with respect to time. Some relevant numerical data are tabulated in Table 3. It can be observed from the figure and the tabulated data that the change in the RMS variance values compared to the no fault case of the H_{∞} filter are larger than the change in RMS variance values of the EKF. Hence H_{∞} filter is better suited to be a fault indicator. Further the type of noise has not affected the performance the filter. Hence we can conclude that H_{∞} filter is better suited a fault indicator as it is more sensitive to the fault and performs independent of the type of noise affecting the system.

Table 3 Numerical results extracted from the Fig. 5

Filter	RMS variance (in 10 ⁻³)	
	$\mu = 0$ (no fault)	$\mu = 0.2$
H∞ filter with uniform noise added	0.7188	1.39
H∞ filter with Gaussian noise added	0.7572	1.456
EKF filter with uniform noise added	0.6268	1.1161
EKF filter with Gaussian noise added	0.7464	1.201

6 Summary, Conclusion and Future Work

The work presented tests the performance of a H∞ filter based fault indicator that can be used on any component of a smart grid. Here a synchronous generator was chosen as the component and a fault indicator for stator interturn fault was developed. This fault was chosen as it is the basis for almost all kinds of electrical faults in a synchronous generator. Here a simulated fault injection capable synchronous generator was used to test the performance of the fault indicator. The fault indicator was the RMS variance of the residual signal obtained from the H∞ filter. It was observed that the fault indicator will respond to the fault and will not respond to similar effects like load imbalance. Further the indicator can also be used to compute the fault parameter by using the Eq. (2) obtained from the experimental data. It was also proved that the proposed H∞ filter based fault indicator is more sensitive to the fault than a traditional EKF based fault indicator. It is also functions in the same manner independent of the noise and uncertainties effecting the system. The main contribution of this work is to use H∞ filter based residual as indicator for faults especially for a wound rotor synchronous generator. To the best of the author’s knowledge this approach has not been tried previously. This work can be extended by applying to other components of the smart grid provided the model of the component along with the model parameter values that are available. Hence this approach provides a very reliable real-time health monitoring system for a smart grid and this can be implemented in software without any extra hardware. As a further extension this system can be tested on a lab-scale setup [16].

References

1. Zhang, H.-T., & Lai, L.-L. (2012). Monitoring system for smart grid. In *2012 International Conference on Machine Learning and Cybernetics*. <https://doi.org/10.1109/icmlc.2012.6359496>.
2. Aubert, B., Regnier, J., Caux, S., & Alejo, D. (2015). Kalman-filter-based indicator for online interturn short circuits detection in permanent-magnet synchronous generators. *IEEE Transactions on Industrial Electronics*, *62*, 1921–1930.
3. Gopinath, R., Santhosh Kumar, C., Ramachandran, K., Upendranath, V., & Sai Kiran, P. (2016). Intelligent fault diagnosis of synchronous generators. *Expert Systems with Applications*, *45*, 142–149. <https://doi.org/10.1016/j.eswa.2015.09.043>.
4. Gopinath, R., Kumar, C., & Ramachandran, K. (2016). Scalable fault models for diagnosis of synchronous generators. *International Journal of Intelligent Systems Technologies and Applications*, *15*, 35–51. <https://doi.org/10.1504/IJISTA.2016.076103>.
5. Simon, D. (2006). *Optimal state estimation*. Hoboken, NJ: Wiley.
6. Ahmad, H., & Namerikawa, T. (2011). H_∞ filter-SLAM: A sufficient condition for estimation. *IFAC Proceedings Volumes*, *44*, 3159–3164. <https://doi.org/10.3182/20110828-6-IT-1002.00260>.
7. Yang, W., Yu, D., & Kim, Y. (2013). Parameter estimation of lithium-ion batteries and noise reduction using an H_∞ filter. *Journal of Mechanical Science and Technology*, *27*, 247–256. <https://doi.org/10.1007/s12206-012-1203-z>.
8. Vikalo, H., Hassibi, B., Erdogan, A. T., & Kailath, T. (2005). On robust signal reconstruction in noisy filter banks. *Signal Processing*, *85*, 1–14. <https://doi.org/10.1016/j.sigpro.2004.08.011>.
9. Guo, L., Mao, Y., & Song, C. (2012). H infinity filter in maneuvering target tracking of military guidance field. In *International Conference on Automatic Control and Artificial Intelligence (ACAI 2012)*. <https://doi.org/10.1049/cp.2012.1173>.
10. Boulkroune, B., Pages, O., Aitouche, A., Zemouche, A., & Hajjaji, A. (2014). H_∞ -based fault diagnosis for diesel engines. In *2014 IEEE Conference on Control Applications (CCA)*. <https://doi.org/10.1109/cca.2014.6981414>.
11. Nohra, C. (2013). Online stator and rotor fault diagnosis in induction machines by H_∞ observer and sliding mode estimator. In *2013 25th Chinese Control and Decision Conference (CCDC)*. <https://doi.org/10.1109/ccdc.2013.6561511>.
12. Henry, D. (2008). Fault diagnosis of microscope satellite thrusters using H-infinity/ H_∞ filters. *Journal of Guidance, Control, and Dynamics*, *31*, 699–711. <https://doi.org/10.2514/1.31003>.
13. Duvvuri, S., & Detroja, K. (2015). Model-based stator interturn short-circuit fault detection and diagnosis in induction motors. In *2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE)*. <https://doi.org/10.1109/icitee.2015.7408935>.
14. Rabbi, A. (2016). *Detection of stator interturn fault of synchronous machine by rotor current analysis* (Master thesis project, School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden).
15. Nag, P. V. S., Kumar, C. S., Sindhu Thampatty, K.C., & Isha T. B. (2018). A modified approach for application of augmented extended Kalman filter for stator interturn fault diagnosis of a synchronous generator. In *4th International Conference on Electrical Energy Systems (ICEES-18)*.
16. Gopinath, R., Nambiar, T., Abhishek, S., Pramodh, S., Pushparajan, M., & Ramachandran, K., et al. (2013). Fault injection capable synchronous generator for condition based maintenance. In *2013 7th International Conference on Intelligent Systems and Control (ISCO)*. <https://doi.org/10.1109/isco.2013.6481123>.

A Survey on Intelligent Transportation System Using Internet of Things



Palak Patel, Zunnun Narmawala and Ankit Thakkar

Abstract An Intelligent Transportation System (ITS) can reduce traffic congestion on roads through reduced use of private vehicles. For the same, we need to expand existing infrastructure for the identified region, but considerable time and resources are required to set up state-of-the-art infrastructure from scratch. However, technologies like Internet of Things (IoT) can be used with the existing infrastructure for the design of an efficient public transportation system. This paper surveys a set of solutions available in the literature to design of an ITS system using IoT along with challenges and future scope for the improvement of the existing solutions.

Keywords Internet of Things (IoT) · Intelligent Transport System (ITS) · Survey

1 Introduction

Intelligent Transportation System (ITS) is a smart way of providing transportation with the help of various technologies [1]. It deals with various modes of transportation and traffic management system using Internet of Things (IoT). IoT is a network which connects people, things, application, and data through the Internet to enable remote control and management of the devices [2]. IoT works on the concept of connecting people to connecting everything. Basically, it collects the data from sensors, actuators and other devices to compute the data using different algorithms. The computed data will be sent to application layer so that it is available to the end user. This put forth requirements to design an embedded system using sensors and mobile phones with

P. Patel (✉) · Z. Narmawala · A. Thakkar
Institute of Technology, Nirma University, Ahmedabad 382481, Gujarat, India
e-mail: 15mcei22@nirmauni.ac.in

Z. Narmawala
e-mail: zunnun80@gmail.com

A. Thakkar
e-mail: ankit.thakkar@nirmauni.ac.in

communication and computing capabilities. Intelligent transport system is mainly useful in Metro Cities and Urban areas.

Public Transportation should be a part of the solution for sustainable development in transport. But most of the people prefer private vehicles to travel which leads to traffic congestion as well as high utilization of non-renewable resources. However, in order to keep and attract more passengers, public transport must have high service quality to satisfy and fulfill a more wide range of different customer base as needed.

1.1 Different Application Areas of Intelligent Transportation System

Following are various fields in ITS useful to common people in their daily lives:

Advanced Traveler Information System (ATIS) provides information regarding arrival and departure of a transportation vehicle, predicting availability of routes, traffic congestion, pre-trip information to reduce waiting time in traffic and frustration due to lack of information [3].

Advanced Public Transportation System (APTS) is concerned with enhancing the efficiency of public transportation using intelligent schedule management according to the congestion level in various areas [4].

Advanced Traffic Management System (ATMS) is used by traffic regulation authorities to monitor and control traffic flow. It is also useful to take appropriate decision in real-time conditions to provide effective traffic management during the emergency situations by giving higher priority to the emergency vehicles [5].

Emergency Management System (EMS) is helpful in emergency conditions like tracking accident spots and providing services. It is effective in providing emergency services [6].

1.2 The Current Scenario of Public Transportation

- People are not preferring public transportation these days because of unavailability of transportation at the scheduled time. Hence, people travel by their private vehicles that increase number of vehicles on roads. It also causes environmental degradation.
- Due to heavy traffic, more time is required to reach the desired place. This may result in incomplete sleep and various kind of health issues to the human and sometimes accidents too.
- People prefer public transportation less due to non-maintenance of the public transportation vehicle. ITS using IoT can help in monitoring and maintaining public transport facilities which can also improve safety.

1.3 Benefits of Using Intelligent Transportation System

Nowadays more than half of the population reside in urban areas which results in traffic congestion and wastage of environmental resources. Awareness must be spread among people to use public transportation that helps to provide overall sustainable development of the country [7]. If more people start using public transportation instead of preferring their own private vehicles, it can be helpful in reducing pollution and indirectly helpful in saving the environment.

The following facilities can be provided to the customers using ITS.

- It introduce reliable public transportation system so that people can easily rely on the provided services.
- It has improved safety features for the betterment of people.
- Highly efficient schedule management so that people can know the exact timings for the transportation.
- Convenience with respect to facilities, availability of seats and other services.
- A smart system can be introduced which could generate a quicker response to user's query and could predict the availability of seats, whether the bus is fully occupied or still, some seats are remaining. Keeping such records may help in building a system which could ease lifestyle of the people and can provide a comfortable journey for travelers.

2 Literature Survey of Intelligent Transportation System

According to the prediction of Barcelona tourist guide, IoT has potential to increase global profits by 19% with the help of effective asset utilization, employee productivity, supply chain, customer experience, and innovations [8]. There are approximately 800 million vehicles in the world which are maintained by Vehicle Dynamic Control (VDC) system which collects the data of vehicles related to the performance and quality of vehicles for analyzing, processing, and remote monitoring. According to a survey, around 40% of traffic is due to parking problem [8]. Hence a smarter way of parking is implemented in Barcelona city which helps users to locate space for parking by their vehicles.

There is a prediction from Cisco that the number of the connected devices will increase to 50 billion by the end of the year 2020. This leads to salable, energy saving, and a portable devices which consist of accurate and secured architecture [9].

In developed countries, the transportation system is well-structured and cost-effective. All stops are clean and hygienic as resources are more compared to the population. But in India, people have a tendency to rush towards the bus as soon as it arrives, and as a result of this, elders, women and physically unfit people fail to catch the required bus. India is a highly populated country resulting in lack of resources for the increasing population. In order to serve large number of people with available resources, an effective solution must be implemented that attracts people to use public

transportation, e.g., current location of the bus can be provided by an ITS [10–16]. An ITS may also provide the number of seats vacant in a plying vehicle in real time.

2.1 Various Existing Intelligent Transportation System [17]

Electronic Fare Collection (EFC) that automate the ticketing system of the public transportation network and collects electronic toll at a toll station.

Highway Data Collection (HDC) provides information of traffic scenario based on positioning devices (Global Positioning System-GPS, Global System for Mobile communication-GSM, General Packet Radio Service-GPRS) attached to road network. It also collects wind velocity data, direction, etc., to predict weather condition for the traveler.

Traffic Management Systems (TMS) plays a central role by gathering various information from different hardware to improve the overall efficiency of transportation.

Vehicle Data Collection (VDC) collects parameters of vehicles for predictive maintenance, durability, and tracking purpose.

2.2 Challenges in Vehicle Tracking Using ITS

According to Verma et al. [10], Global Positioning System (GPS) is one of the most emerging technology and Global System for Mobile communication (GSM) network are easily available at most places. Hence, to get current location of any vehicle, authors have proposed a simple design using Liquid Crystal Display (LCD), GPS, General Packet Radio Service (GPRS) modules with ATmega16 micro-controller. They have also designed a web-based application. Using the system, one can track vehicle remotely. This system is mainly used to track the stolen vehicle.

Farooq et al. [11] proposed a bus tracking system using Arduino, Radio-Frequency Identification (RFID), Infrared (IR) sensors to track the bus. This user-friendly system consists of RFID reader which is located near the gate of the bus station. When the bus with RFID tag enters the premise, a signal is sent to control room that confirms the arrival of the authenticated bus to the bus station. IR sensors are attached to parking slot for the proper parking of the authenticated buses.

Mundada et al. [12] discussed a survey done in Bangalore city that shows that IoT is evolving continuously as Broadband connections are getting cheaper. In this system, GPS, GSM, and cameras are used for counting vehicles to get traffic situation. Camera is attached at various nodes in the road network and with the help of Zigbee protocol, data of the various nodes are sent to the server which can be accessed by the consumer.

Kumar et al. [13] proposed a low-cost design for passengers using bus transportation. GPS is attached to the bus so that one can track its current location. This system consists of four modules: BUS Station Module, In-BUS Module, BASE Sta-

tion Module and BUS Stop Module. In each module, different types of equipments like GPS, GSM, micro-controller, etc., are used to integrate information. Matrix display is attached at the station to show the location of the bus so that passengers can know arrival time of the bus.

Hassan et al. [14] presented a case study of Tanzania, where there is no public transport monitoring system. It reveals that many deaths and injuries occur mainly due to passenger overloading and not following traffic rules in this East-African country. To overcome this problem, a system is proposed for monitoring and reporting excessive passengers in public transportation. It includes Passive Infrared Sensor (PIR) to count passengers and uses GPS/GSM to transmit location data. The system sends alerts to the server room if bus carries excessive passengers. This condition delays the departure of the bus. This system provides safety to passengers and reduces accidents.

Tarapiah et al. [16] proposed a system for vehicle tracking using GPS/GSM module. Vibration/shock sensors are connected in air-bags of the vehicle to send an alert to nearby hospital and police station in case of an accident. The application is composed of GPRS and Google map API with various web services using HTTP REST protocol. “Ray” algorithm is proposed to support online and offline mode of location.

2.3 *Seat Occupancy Detection*

Youngki et al. [18] Lee has proposed a system in which seat sensor has been used. The system is applied on 36 seats of the library in their university. In this system, the presence of a human is detected by the capacitive sensor which works by sending continuous data stream in voltages, which is further connected to micro-controller. This system works on the fundamentals of capacitive coupling where object interrupts proximity of capacitive foil used in seats. Further, the foils are connected to capacitive chips and attached with Raspberry pi. When foils come in physical contact, the shape of the foil gets changed changing the capacitance. But the system gives inaccurate results and requires frequent recalibration of the sensor. There is a possibility of hardware failure also (Table 1).

George et al. [19] has used capacitive and inductive proximity sensor. The system involves capacitive sensor which is able to detect the presence of humans and the inductive sensor is used to detect all the conductors like laptops. The drawback with this type of hybrid system is if the laptop is kept on the seat with power supply to the system and when a person comes in contact with this laptop, then the status of the seat changes to occupied. Table 2 summarizes various seat occupancy detection methods found in the literature.

Zeeman et al. [20] proposed a low-cost system to measure occupancy in a minibus taxi. In sub-Saharan Africa, there is an informal public transportation with less passenger mobility. To improve understanding among passengers, the capacitive proximity sensor is used to detect occupancy of seats and it is integrated with cellular communication to provide real-time information. Capacitive sensing is a technique to

Table 1 Summary of ITS solutions available in the literature

Paper reference	Advantages	Disadvantages	Suggestions
Verma et al. [10]	It is mainly used for tracking purpose only	It uses CMOS 8-bit micro-controller which has less features	Micro-controller like Arduino can be used for efficient results
Farooq et al. [11]	Convenient for passengers as they can track bus, also parking available for buses can be known	People cannot track the location of a bus when it is not in the range of RFID	GSM/GPS technology can be used
Mundada et al. [12]	Overcome customers' difficulty by providing various facilities	ZigBee is having short range and integration of area wise count is time-consuming	System collects data from every vehicle terminal and customers can get public transit information which leads security issues, hence only the result can be displayed
Kumar et al. [13]	It has various types of alarms like battery availability, stoppage, getting late, route deviation	SMS are sent but if there is a network problem, passengers are unavailable to get the message	Real-time application can be made
Hassan et al. [14]	Monitoring and reporting excessive passengers in public transportation	PICAXE system is used as an embedded system so advanced features cannot be implemented	Micro-controller like Arduino can be used to implement advanced features
Tarapiyah et al. [16]	Vibration sensors are attached to an airbag. In case of an accident, it will notify surrounding areas within the range of the accident	Useful in particular range	Ultrasonic sensors can be used to avoid accidents

measure the capacitance between the electrode and its surrounding. The capacitance of unoccupied seat is less than occupied seat based on various results. A mathematical model is proposed to measure the capacitance of sensor electrode.

In a system by ETA Info-Tech in Dubai [21], passenger seat sensor are attached in the taxi. It is made by European Union which are very high quality, reliable and durable. It is attached to the center of the seat where the sensor can be placed flat and its wire is passed from the backside of a seat which is integrated with meter connection.

Table 2 Papers related to seat occupancy detection

Paper reference	Advantages	Disadvantages	Suggestions
Youngki et al. [18]	It works on the fundamentals of capacitive coupling attaching foil in a seat	When foils come in physical contact, its shape gets changed and gives an inaccurate result	Mechanical support capacitive sensor can be used
George et al. [19]	New approach of the combined sensor is used to detect occupancy	Complex system	Only one concept can be used
Zeeman et al. [20]	Simple and low-cost capacitive sensor system is ideal for occupancy detection in multiple-seat vehicles accurately	Parallel plate capacitive sensor is used with diameter 22 cm and height 10 cm	Small mechanical sensor can be used which can be studded in seat easily
ETA Info-Tech. [21]	Simple and secure system	Sensors need to be placed on flat surface. They get damaged under pressure in an uneven surface. Further, current ratings must not vary	Better quality of seat sensors should be used like capacitive sensors

2.4 Traffic Monitoring

The government of Gujarat [22] has implemented City Surveillance and Intelligent Traffic Monitoring System wherein cameras are used to survey movement of vehicles. If someone breaks the rules, the camera will detect vehicle’s number plate and E-challan will be sent to the registered address of the vehicle. These cameras are connected via optical fiber cables or wireless nodes which sends signals to the monitoring room. Table 3 summarizes few traffic monitoring ITS solutions.

World Sensing System [23] focuses on city mobility management including various traffic flows, smart parking, etc. It uses real-time information to make smarter and faster decisions. In Barcelona, smart parking system is available where an electromagnetic sensor is placed in each parking slots. When a vehicle enters that slot, the sensor will send signals of occupancy. This status can be viewed in-app which guides users to proper parking space.

Table 3 Existing implementations for traffic monitoring

Paper reference	Advantages	Disadvantages	Suggestions
The government of Gujarat [22]	Data capture by the camera are sent through fiber optics or wireless network and if someone breaks rules then E-Challan gets generated	Issuing E-Challan is a manual process	By doing automatic number plate recognition, it can be automated
World sensing system [23]	Electromagnetic sensors are used to detect the arrival of cars	Electromagnetic sensors sometimes miss objects like steel ramps and also deflect acoustic waves	Use of other sensors like IR, ultrasound, etc.

3 Future Research Directions

No research is whole and complete. There are chances to improve the existing research work. The literature review presented so far have implemented ITS with different features and each of them has been implemented in isolation. There is a strong need to design an ITS by integrating all these features in a single system that caters the need of different users. Safety, accuracy, and cost-effectiveness can be considered as parameters of interest while designing smart ITS. Also below challenges faced by IoT can be considered:

- **Security:** As the number of devices will be connected, their security will be a big challenge. Also, centralized IoT server needs to be secured.
- **Connectivity:** If the number of devices is moderate then the system is manageable. But, if billions of devices are involved then managing network is a challenging task.
- **Compatibility:** Every device connected for IoT is diversified in its own protocol and operating system, which needs to be synced in order to make it usable for IoT architecture.

For an IoT architecture, following basic hardware is required:

- **Sensors:** To fetch data which would be on the sensor layer of an architecture
- **Micro-controller/ Microprocessor:** To integrate the data. This would be in communication layer of an architecture.
- **Device:** To display data. This would be in an application layer of an architecture.

This technology leads in the business world with major impact on the environment. A better world can be achieved by optimizing the use of IoT technology by solving its open challenges. Most of the people are dependent on the public transit. Hence, a traveler should get the level of convenience same as his/her private vehicle, i.e., amount of waiting time should be minimal and assurance of a seat in the public

transport vehicle. Tracker sensors will be able to track the vehicle, seat sensors can track the number of people through which prediction of the crowd can be calculated. Gateways and mediators can be used to merge various sensors and actuators which leads to storage of data in a particular location via signals or communicators. This leads to a better approach for an Intelligent Transportation system.

4 Conclusion

Public transport has the potential to reduce traffic congestion if it is implemented in intelligent and efficient manner. Existing efforts for the same lack a holistic approach. There are numerous transit system ideas using low-cost IoT technology but each of them lacks in consistency. A complete architecture for Public Transport System using IoT should be devised and standardized for its wide adaptation.

References

1. van Arem, P. B. (2015). In *Intelligent Transportation System*. Intelligent Transportation System Society.
2. Saleh, I. (2012). *Internet of Things*. Department of Civil Engineering, Wiley.
3. Khattak, A. J., Targa, F., & Yim, Y. (2004). *Advanced traveler information systems*.
4. Uchimura, K., Takahashi, H., & Saitoh, T. (2002). *Demand responsive services in hierarchical public transportation system* (Vol. 51).
5. Hancke, G. P., & Nellore, K. (2004). *Advanced traffic management system*
6. Singh, A. G. B. (2015). *The journal of Transport Literature*. International Transport Planning Society.
7. Das, J. H., & Tom, S. (2016). Futuristic intelligent transportation system architecture for sustainable road transportation in developing countries
8. *Barcelona tourist guide*. 2015.05.04.
9. Yaqoob, I., Ahmed, E., Hashem, I. A. T., Ahmed, A. I. A., Gani, A., Imran, M., & Guizani, M. (2017). *Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges* (Vol. 1). IEEE Wireless Communications.
10. Verma, P., & Bhatia, J. (2013). Design and Development of GPS-GSM based tracking system with google map based monitoring system. *International Journal of Computer Science, Engineering and Application*, 3.
11. Vidyasagar, K. S. K., & Farooq, M. A. (2015). Public road transport guiding system using arduino microcontroller. *International Journal of Computer Applications*, 132, 12–16.
12. Mundada, M. R., & Selvapriya, P. R. (2015). IoT based bus transport system in banglore. *International Journal of Engineering and Technical Research*, 3.
13. Kiran Kumar, G., & Prasad, M. (2012). Public Transportation Management Service using GSM-GPS. *International Journal of Research in Computer and Communication Technology*, 1.
14. Machuve, D., Hassan, K., & Sam, A. (2013). A system for monitoring and reporting excessive passengers in public buses case study: Tanzania. *International Journal of Engineering And Computer Science*, 1.
15. Rane, K. P., Dukare, S. S., & Patil, D. A. (2015). Vehicle tracking, monitoring and alerting system: review. *International Journal of Computer Applications*, 119.

16. AbuHania, R., Tarapiah, S., & Atalla, S. (2013). Smart on-board transportation management system using GPS/GSM/GPRS technologies to reduce traffic violation in developing countries. *International of Digital Information Communication*.
17. Abdullah, A. H., & Qureshi, K. N. (2013). A survey on intelligent transportation system. *Middle-East Journal of Scientific Research*, 15.
18. Youngki, L., Rajesh Krishna, B., Nguyen Huy, H. H., & Hettiarachchi, G. (2013). *Small scale deployment of seat occupancy detectors* (Vol. 1). Singapore Management University.
19. Bretterkieber, T., Brasseur, G., George, B., & Zangl, H. (2010). A combined inductive capacitive proximity sensor for seat occupancy detection. *IEEE Transactions on Instrumentation and Measurement*, 59.
20. Booysen, M. J., & Zeeman, A. S. (2014). Simple capacitive seat sensing for occupancy detection and passenger counting in minibus taxis. Research Gate.
21. Eta-info tech. (2016). Retrieved from <http://www.etastarinfotech.com/products/fleetmanagement/vehicle.htm>. Accessed December 03, 2015.
22. *City surveillance and intelligent traffic monitoring system*. No. Sep 2016. Accessed December 03, 2015.
23. *World sensing system*. <http://www.worldsensing.com/industries/cities-governments/>, 2016. Accessed December 03, 2015.

CRUST: A C/C++ to Rust Transpiler Using a “Nano-parser Methodology” to Avoid C/C++ Safety Issues in Legacy Code



Nishanth Shetty, Nikhil Saldanha and M. N. Thippeswamy

Abstract CRUST, a language translator (transpiler) has been developed which converts programs in C/C++ to programs in Rust. Rust created by Mozilla has become popular as a systems programming language as it provides constructs and tool support for developing safe and secure programs without hassles. Safe Rust ensures that the programs developed using these programming constructs are safe and secure by enforcing compiler restrictions, concurrency without data races, memory safety without garbage collection, and abstractions with low overheads. CRUST is a semi-automated transpiler, to automatically convert a subset of “C/C++” code base into Rust without much effort. This is done using a unique Nano-Parser Methodology. CRUST also enhances the readability and understandability of the program by adding extensive documentation to the translated code. It yields a code which is guaranteed to be thread safe and proven to be faster.

Keywords C++ · Memory safety · Nano-parser methodology · Rust · Transpiler · Type safety

1 Introduction

C and C++ programming languages provide support for easily and efficiently interacting with the underlying hardware using high level program constructs, when compared to most of the other high level programming languages [1–3]. However, it takes a lot of effort in order to ensure that the developed programs are safe and secure as there is no tool support to enforce this during compilation. This has to be handled

N. Shetty · N. Saldanha (✉) · M. N. Thippeswamy
Department of CSE, Nitte Meenakshi Institute of Technology, Bangalore 560043, India
e-mail: saldanhanikhil@gmail.com

N. Shetty
e-mail: nishanthspshetty@gmail.com

M. N. Thippeswamy
e-mail: thippeswamy.mn@nmit.ac.in

by the programmer, which is generally a tedious task and such patches added to the code as an afterthought are generally buggy.

Programming mistakes such as buffer overflow, null pointer dereferences, and incorrect type casts causes the system to break down. Moreover, the lack of safety restrictions can be exploited by an attacker to compromise the security of these systems [4, 5]. These issues that need to be addressed in order to develop safe and secure programs have been studied extensively and numerous solutions have been proposed to alleviate them [6–10]. However, all these require quite a bit of manual intervention. Nowadays, there has been a dramatic rise in the popularity of functional programming languages such as Clojure, Haskell, OCaml, Scala, and support for functional programming principles in existing languages such as JavaScript (ECMAScript 6), Java 8. Functional Programming Languages by principle, solve concurrency and memory issues which plague programs written in C and C++. Rust is at its heart a functional programming language, which has mechanisms to ensure type and memory safety at its very core.

Rust, is Systems programming language designed by Mozilla to achieve safe systems and painless concurrency. The result is safe programming with highly enforced compiler restrictions, memory safety without garbage collection, and zero-cost abstraction. Rust achieves what C/C++ does not, by enforcing safety through compiler restrictions. Due to these restrictions, it is guaranteed to be safer and more efficient. It is a concept of borrowing and ownership that creates a well-defined object scope, which avoids disaster.

In Rust, the problem of data race is overcome by employing mutable and immutable references, where only mutable references can be borrowed. These along with a few other principles have been built into Rust so that the programmer does not have to take care of it explicitly. There is sufficient proof that Rust combines the best of both worlds, low-level control and type/memory safety [11–13].

There is an opportunity to permanently solve problems associated with systems programming by simply moving from C and C++ to Rust, which is a safer and more predictable language. CRUST provides the means to do this efficiently and effortlessly. Though we do not claim to be able to convert every construct, a fairly large subset of code has been tried and tested. The sections that cannot be converted currently, are well documented so that the user can manually convert these to a Rust equivalent code with the aid of the information provided in the associated documentation.

In the process of realizing a tool such as CRUST, we have devised an elegant manner in which to build a Parser. Aptly named, The Nano-Parser methodology, it involves designing several cooperatively functioning tiny parsers (nano-parsers) each designed for very specific grammars. Using one master parser calling the nano-parsers, it is able to set off a chain of nano-parser calls, ending in the parsing of the complete input.

Once CRUST realizes its full potential, legacy code bases that have been abandoned because of their sheer size and safety may be transpiled and deployed.

In this paper, the novel contributions are as follows:

- Translation of unsafe legacy C/C++ code into safe and manageable Rust code with minimal manual involvement.
- We augment the translated code with helpful documentation to ease the process of understanding the Rust code for those who are unfamiliar with the language.
- Provision of different translation modes controlling memory safety and package management.

There are certain assumptions made in this paper, they are the following:

- The C/C++ code is syntactically correct.
- The external API calls must be handled manually after translation.
- C/C++ code does not include any headers or modules.

2 Related Work

In this section, languages, which are considered as type safe are discussed. Type safety is ensured to a large extent by static analyses of the source code as a part of the compilation process, and traps are set for capturing and handling runtime type exceptions that might occur dynamically [2]. As C/C++ compilers, in general, have no such built-in enforcement mechanisms, the programmer has to manually ensure that such conditions do not arise by performing code reviews before compilation and also augment the code by enveloping the section of code with type checkers inserted into the code to catch type mismatches dynamically. This is definitely a cumbersome and error-prone process.

2.1 Smart Pointers

The use of ‘smart pointers’ is proposed in [6] to solve issues regarding memory leaks associated with deallocated ‘raw pointers’ (regular pointers). The wrapper class has attributes and methods that help to manage the lifetime of an object by building in reference counters. Even with these mechanisms in place adoption of bad programming practices like cyclic references could still lead to memory leaks.

2.2 Probabilistic Memory Safety

The authors in [7] have proposed a runtime system called “DieHard”, which achieves memory safety by using probabilistic methods of randomization and replication and creating a virtual infinite-sized heap. Randomization and Replication increase the safety of the program as the effect of errors across the replicated versions will have a very low probability of corruption of the same variable/entity in the replicated versions.

2.3 *Memory Safety*

The work done in [8] proposes the memory safety for embedded software without Garbage Collection or Runtime Checks. The authors propose a pool of free memory from which the random allocation of objects and their replicates can be done. This will minimize the probability of runtime errors affecting the replicated versions in the same manner and hence enhances the safety of the system.

2.4 *Manual Safe Memory Management*

Gay et al. have proposed HeapSafe, which is a tool that has been developed by them in order to trap “bad” memory free calls in programs written using “C” language. Runtime checks are performed in order to identify dangling pointers and free memory locations and also ensure that “bad” free function calls are treated mostly as no-ops. Standard techniques of reference counting are adapted to verify the C programs and also to ensure their reliability when manual memory management techniques are adopted [9]. It relies on an extension of the malloc and free APIs. HeapSafe cannot introduce new bugs since it only checks the existing memory management API rather than augmenting or changing it [9]. Depending on the programs the HeapSafe scheme can incur an execution time overhead of about 30% and designers might get discouraged from using it for real-time systems with tight budgets.

2.5 *Ironclad C++*

Ironclad C++ [10] is a subset of the C++ language, which contains constructs of C++ which can be verified to be safe by static syntactic analysis of the program segments, and also uses the concept of “smart pointers” to ensure freedom from runtime errors that might be introduced by using simple pointers. Nearly 50,000 lines of code have been successfully translated to Ironclad C++ [10].

3 **Rust as a Successor to C/C++**

In this paper, we believe that the solutions described in the previous are all valid, they do not provide a complete solution and they each come with their own drawbacks. Programmers, who prefer C/C++ do so because of the control it provides, which is taken away when any of these solutions are used.

Figure 1 shows the trade-off between low-level control and safety for various programming languages. It describes how Rust combines low-level control along

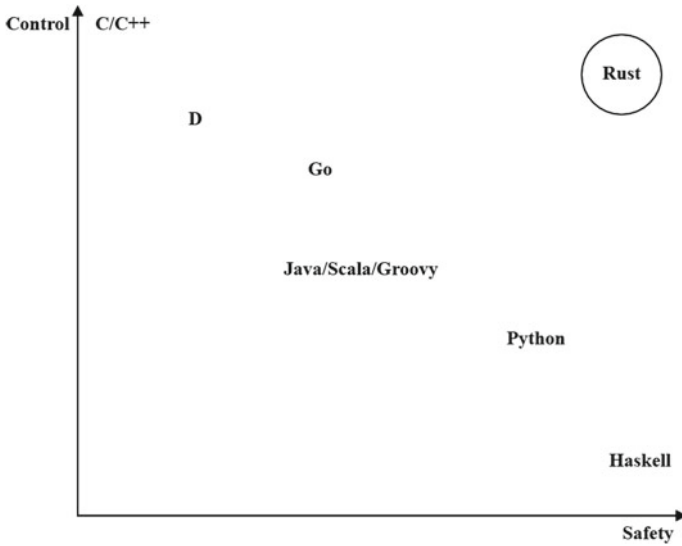


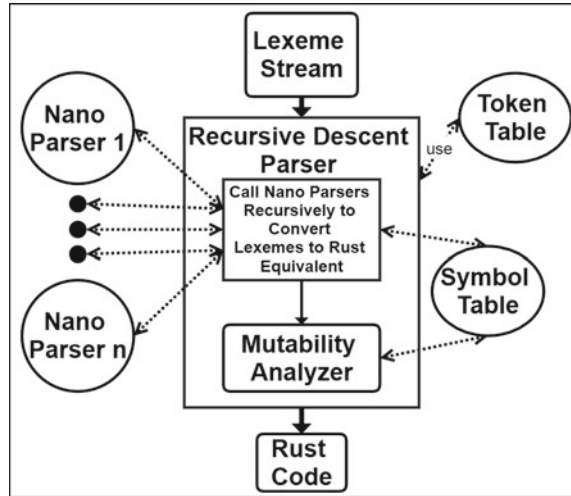
Fig. 1 Tradeoff between control and safety of popular languages

with safety to solve the safety implications that come with using C like languages. Languages like Haskell tried to provide the kind of safety that is desired for hardware applications. A Go is comparable to C and C++ in terms of low-level control and efficiency, but is not so safe as Haskell. This is where Rust comes in with its compile-time enforcement of safety rules and C like low-level control. It does prove that Rust is a worthy successor to C and C++.

It is believed that these results indicate that Rust has overcome the drawbacks that plague C/C++ while still allowing low-level control and efficiency that C is popular for. There is nothing to lose from moving on from C/C++ except for an initial learning curve more importantly in the context of this paper, translating an existing C code base. CRUST solves both these problems by allowing one to translate C/C++ to safe Rust and also provide formatted and documentation in the translated code to ease the learning curve.

In this subsection, the design of Transpiler is discussed. Transpiler essentially is constructed out of two functional blocks, which are executed sequentially in order to translate the program written in “C/C” to “Rust”. Typically, like any other compiler, the main blocks are the Lexical Analyzer and the Parser followed by the Code Generator. The “Match” function which is provided by Rust is used. Multiple DFAs are launched in parallel with each constructed to recognize a template of a valid regular expression belonging to the language. The longest lexeme will be accepted and tokenized. It is assumed that the input source code is syntactically correct. These identified lexemes or tokens are inserted into the Symbol Table. The Parser is mainly composed of two modules: “Global Construct Identifier” (GCI) and “Nano-Parser” (NP). The Lexeme streams as delivered by the Lexer are taken as input to the GCI.

Fig. 2 Design of the parser with the nano-parser methodology



Based on an initial analysis of the streams, it identifies the possible candidate grammatical expressions of the chosen subset of C/C++ language. Once this is identified, the GCI triggers the appropriate NP. Each NP generates a code snippet in “Rust” corresponding to the C/C++ source code snippet which was passed onto it by the GCI (see Fig. 3a, b). The Rust code snippets so generated are then collated to compose the program in Rust which is equivalent of the input C/C++ source code. The parser itself has been architected as a “recursive descent parser”.

We believe that this scheme called “The Nano-Parser Methodology” of using a GCI along with specific NPs is a novel and efficient way of building compilers. Rust inherently supports the concept of mutable and immutable variables. This allows restrictions to be imposed on the type of functions that can be performed on the variables by the functions to which this variable is passed. Therefore, strict control can be enforced on the scope and the operations on variables by the functions to which it is exposed. In order for the user to exploit this feature of Rust to enhance the safety of the program, an option has been given where the user can tag the particular variable as mutable or immutable. This is handled by the Mutability Analyzer (see Fig. 2). This analyzer is invoked during the transcription process and identifies operations which violate the immutability property tagged to a particular variable. These operations are flagged and appropriate error messages are generated for information to the programmer.

In this context, two modes of translation have been proposed:

- **Strict mode:** CRUST will treat all variables as immutable and identifies all operations which violate this criterion.
- **Loose mode:** CRUST will treat all variables as mutable (Fig. 3).

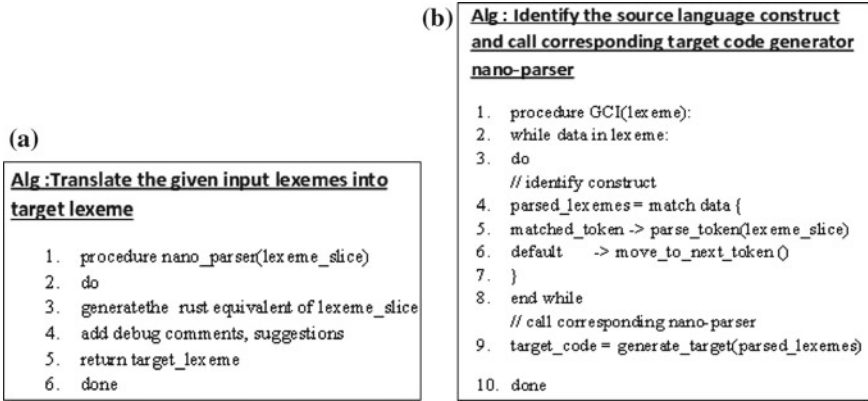


Fig. 3 a Design of the parser with the nano-parser methodology, b global construct identifier algorithm

4 Nano-Parser Scheme

This scheme which has been defined above which comprises of a GCI and several Nano-parsers working cooperatively is quite novel. Nesting of Nano-parsers is allowed in order to support complex programming snippets. For example, an “if” statement inside a “for” loop would be processed in the following manner. The GCI based on the “for” lexeme triggers the NP associated with it. As this NP is processing and translating the said input the “if” statement is encountered. This will lead to the “for-NP” triggering the “if-NP” for further processing leading to the nesting of the NPs (see Fig. 2). Recursive triggering can occur in case an “if” statement is encountered inside another “if” statement.

Salient features of this methodology are listed below:

- (a) As the design of the “Transpiler” is highly modular in nature, it is very easy to implement the same.
- (b) The modular design enabled test coverage of about 95% of the code in Unit testing phase itself and the rest were checked after integration. This highly simplified the task of testing the transpiler implementation.
- (c) The modular nature of the implementation helps in easily identifying and localizing the errors.
- (d) Upgrades and extensions to the allowed set of language constructs can easily be accommodated by building an “NP” specific to this new construct and integrating this with the GCI.

5 Results and Analysis

This section provides sample results for the proposed CRUST. CRUST is able to translate a subset of legacy C/C++ code into Rust code, with appropriate hints in case of compile errors. The portions of the code which do not conform to the defined translatable subset will be retained as such but flagged so that the user can translate it manually.

Subsequently, when the translated code is passed onto the Rust compiler, it will identify statements which are likely to violate the “safety” norms. This whole process enables an experienced “C/C++” programmer to write “safe and secure” programs without having to become proficient with a new programming language like “Rust”. A large number of C/C++ programs have been translated by us using CRUST to prove the translation process. Table 1 shows a few “C” program constructs and the equivalent “Rust” program statements.

In Table 2, we have shown the “C” code developed for the implementation of the matrix addition function and have also shown the translation of the same in both the “strict” and “loose” modes. The “For loop” construct is not supported in Rust, and must be implemented using “while loops”. The documentation that is inserted into the code has been omitted for brevity. It can be observed that the keyword “must” is added in the declaration of variables in the loose section. This allows modification of the variables.

In Sl. No. 2, we see a function being converted to Rust. Sl. No. 3 shows the conversion from C++ classes to Rust equivalent. There are no classes in Rust and hence this must be taken care of using structures with “impl” block. The constructors must be handled manually using static builder functions since Rust structures do not have constructors.

The results show that very little manual translation is required in compiling the translated code. This is very easy to do as the translated code contains appropriate suggestions and additionally the Rust compiler generates very precise error messages. The resultant Rust programs are very efficient and optimized. These processes generate “safe” programs without appreciably compromising on the execution speed. With no aliasing and mutation, Rust avoids memory leaks and data races.

The code was running a “for” loop for 10^9 iterations. The output code was tested by adding output statement to Rust code, I/O is a costly operation but Rust excelled. It performed better than C++ code without any I/O doing the same task. C++ program takes about 2.3 s to run while the translated Rust completes the same program in

Table 1 Comparison of the performance of C/C++ code and its equivalent in rust, translated using the crust

Timing statistic	Average performance	
	C/C++	Rust
Real time (s)	2.3	0.005
User time (s)	2.3	0.002
System time (s)	0.006	0.002

Table 2 C/C++ code translation to Rust using CRUST in strict and loose modes

Sl. No.	C/C++ Code	Rust Equivalent Code	
		Strict Translation	Loose Translation
1.	<pre>int main() { int i=0; int a[4]={1,2,3,4}; int b[4]={1,2,3,4}; int c[4]; for(;i<4;i++) { c[a]=a[i]+b[i]+100; } }</pre>	<pre>fn main() { let i: i32 = 0; let a: [i32; 4] = [1, 2, 3, 4]; let b: [i32; 4] = [1, 2, 3, 4]; let c: [i32; 4]; while i < 4 { c[a]=a[i] + b[i] + 100; i+=1; } }</pre>	<pre>fn main() { let mut i: i32 = 0; let mut a: [i32; 4] = [1, 2, 3, 4]; let mut b: [i32; 4] = [1, 2, 3, 4]; let mut c: [i32; 4]; while i < 4 { c[a] = a[i] + b[i] + 100; i += 1; } }</pre>
2.	<pre>int add_it(int a,int b) { return a+b; }</pre>	<pre>fn add_it(a: i32, b: i32) ->i32 { a + b }</pre>	
3.	<pre>class A { int a; int b; private: float getfloat() { return 1.23; } public: int getInt(int a) { return a; } A() { a = 5; b = 6; } };</pre>	<pre>struct A { a: i32, b: i32, } impl A { /** Rust structures do not support constructors * Please handle them with static builder functions* >>> A () { a = 5 ; b = 6 ; }**/ fn getfloat(&self) -> f32 { 1.23 } pub fn getInt(&self, a: i32) -> i32 { a } }</pre>	

about a 1/2000ths of a second, a 1000 times improvement in runtime speed (see Table 2).

6 Conclusion

The evaluation results prove that we are able to successfully translate the given C/C++ code into Rust. Rewriting existing codebase in Rust requires lots of time, money, and effort. With the help of an automated translation tool, we can get this task done with minimal cost and time. CRUST’s translated code is easy to debug with suggestions and hints provided in the translated code. The code is safe and runs faster than the original C/C++ code. Performance tests have shown that Rust runtime is very efficient and much smaller than that of the source C/C++ code. This is yet another reason to shift to Rust. The use of the Nano-Parser Methodology in the implementation of parser makes it very easy to implement and test the Parser modules. This approach may lead to significant changes in parser design and implementation.

References

1. Stroustrup, B. (2012). Software development for infrastructure. *Computer*, 45(1), 47–58. <https://doi.org/10.1109/MC.2011.353>.
2. Shen, R. (2017). *Why-is-C++-important*. Retrieved from <https://www.quora.com/>. Accessed April 11, 2017.
3. O'Neal, B. (2017). *What is the role of C++ today?* Softwareengineering.stackexchange.com. Retrieved from <http://softwareengineering.stackexchange.com/questions/61248/what-is-the-role-of-c-today>. Accessed April 11, 2017.
4. Pincus, J., & Baker, B. (2004). Beyond stack smashing: Recent advances in exploiting buffer overruns. *IEEE Security and Privacy Magazine*, 2(4), 20–27.
5. CSE341 Lecture Notes 26. (2017). Unsafe language (C). Courses.cs.washington.edu. Retrieved from <http://courses.cs.washington.edu/courses/cse341/04wi/lectures/26-unsafe-languages.html>. Accessed 11 April, 2017.
6. Colvin and Adler, D. (2012). *Smart pointers—boost 1.48.0. Boost C++ libraries*. Retrieved from www.boost.org/docs/libs/1_48_0/libs/smart_ptr/smart_ptr.htm. Accessed 11 April, 2017.
7. Berger, E., & Zorn, B. (2006). DieHard. *ACM SIGPLAN Notices*, 41(6), 158–168.
8. Dhurjati, D., Kowshik, S., Adve, V., & Lattner, C. (2003). Memory safety without runtime checks or garbage collection. *ACM SIGPLAN Notices*, 38(7), 69–80.
9. Gay, D., Ennals, R., & Brewer, E. (2007). Safe manual memory management. In *International Symposium on Memory Management*.
10. DeLozier, C., Eisenberg, R., Nagarakatte, S., Osera, P.-M., Martin, M. M. K., & Zdancewic, S. (2013). Ironclad C++: a library-augmented type-safe subset of C++. *MS-CIS-13-05*. University of Pennsylvania CIS.
11. Turon, A. (2017). Fearless concurrency with rust—the rust programming language blog. *Blog.rust-lang.org*. Retrieved from <https://blog.rust-lang.org/2015/04/10/Fearless-Concurrency.html>. Accessed 11 April, 2017.
12. Turon, A. (2017). Abstraction without overhead: Traits in Rust—The Rust programming language blog. *Blog.rust-lang.org*. Retrieved from <https://blog.rust-lang.org/2015/05/11/traits.html>. Accessed 11 April, 2017.
13. Turon, A., Matsakis, N. (2017). Stability as a deliverable—The Rust programming language blog. *Blog.rust-lang.org*. Retrieved from <https://blog.rust-lang.org/2014/10/30/Stability.html>. Accessed 11 April, 2017.

Species Environmental Niche Distribution Modeling for Panthera Tigris Tigris ‘Royal Bengal Tiger’ Using Machine Learning



Shaurya Bajaj and D. Geraldine Bessie Amali

Abstract Biodiversity loss due to habitat degradation, exploitation of natural deposits, rapid change of environment and climate, and various anthropogenic phenomenon throughout the last few decades in the quest of development have led to rise in safeguarding species ecological domain. With natural habitat of the endangered Panthera Tigris Tigris fast declining, coupled with factors such as loss in genetic diversity and disruption of ecological corridors, there is an urgent need to conserve and reintroduce it to newer geographic locations. The study aims to predict and model the distribution of the species Panthera Tigris Tigris by combining various climatic, human influence, and environmental factors so as to predict alternate ecological niche for the already dwindling tiger habitats in India. 19 Bioclimatic variables, Elevation level, 17 Land Cover classes, Population Density, and Human Footprint data were taken. MAXENT, SVM, Random Forest, and Artificial Neural Networks were used for modeling. Sampling bias on the species was removed through spatial thinning. These variables were tested for Pearson correlation and those having coefficient greater than 0.70 were removed. Kappa statistic and AUC were used to study the results of the methodology implemented. Testing data comprises 25% of the presence only points and test AUC value of MAXENT was found to be the highest at 0.963, followed by RF at 0.931, ANN at 0.906, and lastly SVM at 0.898. These indicated a high degree of accuracy for prediction. The most recent datasets were taken into consideration for the above variables increasing accuracy in both time and spatial domain.

Keywords Artificial neural networks · Bioclimatic · Elevation · Environmental niche · Human influence · Land cover · MAXENT · MODIS · Panthera Tigris Tigris · Random forest · Royal Bengal tigers · Species distribution modeling · Support vector machines

S. Bajaj (✉) · D. Geraldine Bessie Amali
SCOPE, Vellore Institute of Technology, Vellore 632014, Tamil Nadu, India
e-mail: shauryabajaj1@gmail.com

D. Geraldine Bessie Amali
e-mail: geraldine.amali@vit.ac.in

1 Introduction

As a result of various human activities, over-exploitation of natural resources and climate change, various conservation measures need to be taken for the already dwindling species of flora and fauna. The necessity of habitat suitability estimation and species distribution in an ecological niche is closely linked to the conservation and management of biodiversity. Ecological Niche can be described as a set of environmental circumstances which are not hostile for species persistence and offspring production [1]. A productive method for knowing the geographical locations so as to reintroduce the species is species distribution models (SDMs), known by different terms such as habitat suitability models, climate envelope models, and ecological niche models. The main aim is to detect environmental circumstances which are suitable and at the same time not hostile for the particular species to persist using species occurrence (presence) data from biodiversity sources as well as several other factors for the prediction of an output (response) variable, such as habitat suitability for a geographical location. Satellite data with high resolution together with other factors are seen to increase model accuracy [2].

Different SDMs have their own benefits and drawbacks. Performance varies depending on various models, the variables chosen as well as the data chosen (presence only/presence–absence data). The behavior of different SDMs and the influence of various predictor sets on them need to be studied. Performance of the models can be measured through parameters such as area under the receiver operating characteristic curve (AUC), overall accuracy, sensitivity, specificity, and Kappa statistic most widely used being Kappa and AUC. The research uses AUC and Kappa statistic to measure performance.

1.1 Motivation

The study to locate new geographical potential habitats is motivated by various causes. First, declining prey bases and human activities such as deforestation, encroachment, and development have led to a decline in Royal Bengal Tiger's natural habitat. At geographical locations where tigers are increasing in number, there is a tussle for territory among the tigers themselves [3]. Another motivation is to maintain genetic diversity. As per **Wildlife Institute of India** and studies conducted by the **Centre for Cellular and Molecular Biology**, the country is home to three different tiger populations which are genetically connected. They thrive in different parts of India mainly the Terai-Duar Savanna, the northeastern region of India, South India, and parts of north India especially the Ranthambore. It was found that among these, the **Ranthambore** Tiger population has highest threat of isolation and least genetic diversity. This is verified by another study [3]. Thus, the species requires genetic flow. The third major motivation is disrupted ecological corridors. Research indicates that the population of Tiger species outside their natural habitat is 35%

[3]. Thus, it is the ecological corridors between the natural habitats that maintain the flow of the gene pool. Thus, it becomes necessary to link these habitats through ecological corridors.

2 Related Works

Although the concept of SDMs has been used in modeling habitat suitability for various species yet little work has been done for *Panthera Tigris* species. Kywe [4] performed habitat suitability modeling in the Hukaung Valley Tiger Reserve, Northern Myanmar. Covering a landscape of around 1700 km², the research was done for the year 2003. Land classification dataset was developed through segmentation based classification leading to 14 land cover classes being developed. Additional data in the form of topographical and human influence were taken from other resources. The research found that the species of tiger usually are found close to river beds and grass area while avoiding evergreen closed areas and settlements. The reserve was found to have 42% suitable areas for tiger habitats which was done with the help of Ecological Niche Factor Analysis. Hernandez et al. [5] found Maxent to give highly accurate results unbiased of the count of instances of species and the extent of the study area or geographical distribution in comparison to Mahalanobis and Random Forests models. Phillips et al. [2] made use of Maxent to model the potential geographical locations of tiger populations during the time period of Late Pleistocene and Holocene, thus providing new insights into the evolutionary history and interconnectivity between populations of this endangered species. Kulloli and Kumar [6] found out the potential habitats for *Commiphora wightii* (Arnt.) Bhand in the arid zones. The SDM was implemented using MAXENT distribution taking bioclimatic, NDVI, and elevation data in consideration. These variables were tested for correlation and those having coefficient values >0.80 were removed. AUC was greater than 0.9 in all the test cases with different predictor variables which indicates that prediction precision is high. Jaisalmer and Barmer regions of Northwest India were predicted to be geographical areas suitable for the species to be introduced.

Maxent can be modeled on both presence and pseudo-absence data, utilizing both categorical, and continuous dataset. It uses an exponential method to predict values and thus needs additional attention when we shift to an alternate geographical location or a time stamp. Duan et al. [7] considered five species of trees consisting of one coniferous and four broad-leaved species. These were the *Quercus wutaishanica*, the Japanese white birch, the *Mongolian oak*, the Chinese cork oak, and *P. massonia* which then made use of 13 different environmental variables to predict alternate habitats. A total of six different species distribution model mainly were then compared and their accuracy was evaluated using the coefficient of variation, Kappa statistic, and area under the operating curve. The map closest to the mean value of the curve was transformed into a binary map and displayed. Qing et al. [8] implemented a Maxent model making use of presence-only records for predicting

the winter ecological niche for the scaly-sided merganser in China. AUC indicated high accuracy for both training and testing cases.

3 Methodology for Alternate Ecological Niche Prediction

3.1 Study Species and Distribution Data

The tiger species is classified in the genus *Panthera* with the lion, leopard, jaguar, and snow leopard. Royal Bengal Tiger is one of the nine subspecies three of which are extinct. An endangered species as listed by the International Union for Conservation of Nature (IUCN) Red Book, India provides habitat to 2226 Royal Bengal Tigers also known as *Panthera Tigris Tigris*. It is one of the subspecies of the Felidae family, the largest of the ‘big cats’ in the genus *Panthera*. Records of occurrence for *Panthera Tigris Tigris* were collected from Global Biodiversity Information Facility which provides combination from various sources such as the Wildlife Institute of India, iNaturalist and mainly the field museum of natural history for different temporal and spatial data. The locations depicting the physical presence of *Panthera Tigris Tigris* in terms of latitude and longitude along with 233 other attributes were extracted. Figure 1 represents the architectural design of the steps undertaken. Data from GBIF is preprocessed and plotted on Google Maps. Different datasets representing various ecological, climatic, and human factors are taken from a variety of sources. These are tested for correlation and sampling bias is removed. Following this, different machine learning algorithms are applied and the results are studied and described.

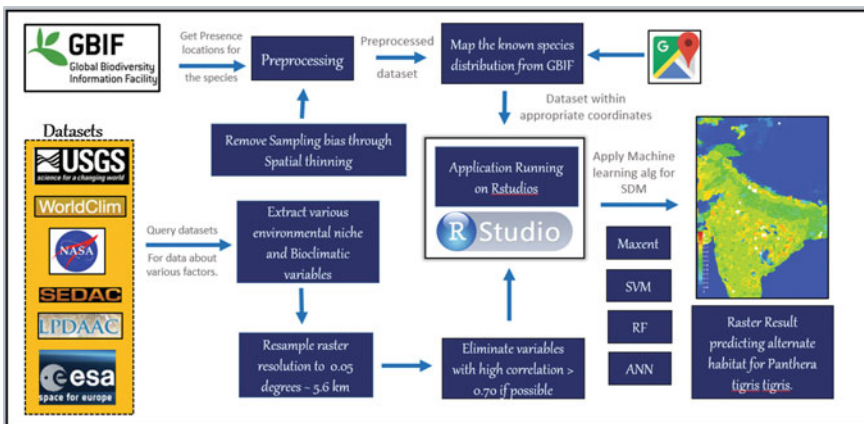


Fig. 1 Architecture to represent species distribution modeling for *Panthera Tigris Tigris*

3.2 *Preprocessing of Distribution Data*

From GBIF, the Darwin core Archive was downloaded as it is directly given to the publisher consisting of 422 global occurrences. After extracting the coordinate positions of *Panthera Tigris Tigris*, there is a need to preprocess this data which contains presence only locations. It becomes necessary to avoid errors in coordinate locations, misleading information about the species as well as records having null attributes. The following steps were followed and after their conclusion, the preprocessed data was displayed to depict the geographical coordinates in R. The above preprocessing filters the number of occurrence to 130 from 442.

- **Selection of important attributes**—From the gbif file of the above species, we select only attributes of importance from the 235 attributes such as species latitude, longitude, country, name, and any other which seems necessary.
- **Removing duplicate records**—For the same spatial (in real time geographical) coordinates, a species may have presence only data at multiple time stamps. Thus, it is necessary to remove them as it may bias the output.
- **Georeferencing**—There may be records with no coordinates but with locality descriptions. These were georeferenced. This could be done with the help of geocode function present in the *dismo* package in R if necessary.
- **Map the dataset within the required spatial coordinates**—The species records were limited to the Indian subcontinent by imposing restriction on the raster extent. The extent was decided to be (64.84885, 97.71038, 5.758113, 35.43777) as per the standard (x_{\min} , x_{\max} , y_{\min} , y_{\max}) to represent the Indian subcontinent.

3.3 *Removing Sampling Bias*

Some presence only data is derived from random sampling. These may be collected near frequent forest treks and reserve park roads and these spatial sampling biases are not included in presence-only data. To remove this, spatial thinning was implemented using the R package *spThin*. The idea is to preserve most effective information while removing bias through removing records on the basis of a thinning distance of 3 km for a geographical area. Finally, from the 442 occurrences, we focus down on 48 occurrences localities after performing spatial thinning.

3.4 *Environmental Data*

The environmental data consists of various datasets representing various environment variables, land cover, and human influence on the environment among others so as to model suitable habitats for *Panthera Tigris Tigris*. The resolution chosen was 0.05° as this is suitable for prediction at the national level. At the equator, this comes out

to be 5.6 km². The Coordinate reference system (CRS) chosen is the EPSG: 4326, WGS 84.

- **Land cover**—Many recent researches make use of The University of Maryland's (UMD) global land cover classification at a one kilometer resolution [9]. This classification was done using the 1992–1993 satellite images. These datasets have become outdated with time and for rapidly developing country like India these may not represent the actual spatial land cover data in the current scenario. To minimize such differences, one of the MODIS products, the Land Cover Type Climate Modeling Grid product MCD12C1 consisting of seventeen different land cover classes was taken to represent land cover for the Indian subcontinent. MCD12C1 is at 0.05° resolution for which the pixel size is around 5.6 km². The Royal Bengal tiger usually thrives in a wide variety of habitats such as tropical and deciduous rainforests, swamps, shrublands, grasslands, and high altitude. In the Indian region, they are found along the Western Ghats, in deltas of Bengal, the Terai highlands, Northeast region and parts of central India among others. Thus, the classes depicting the above landscape were taken to represent their habitat for more accurate prediction.
- **Bioclimatic**—Worldclim contains a set of nineteen climatic factors. The raster dataset was then averaged for all the months which depicted the average over the above time period. These were resampled to 0.05° using bilinear interpolation.
- **Elevation**—For elevation, the GTOPO30 dataset was utilized. It is a part of the elevation schema of the United States Geological Survey (USGS) updated and released in 2000. The dataset which was available in 30 arc second resolution was resampled to using bilinear method to 0.05°. The dataset was then cropped to respective extent so as to represent the Indian subcontinent.
- **Global Human Footprint and Population density**—GHF is a dataset under the Last of the Wild, v2 (2004) from SEDAC depicting human impact in terms of population pressure, human access in terms of distance to roads, coastline, rail lines, and navigable waterways and infrastructure development in terms of night lights and built up area. The human population density raster dataset was taken from GPW, v4 estimated for the year 2020.

3.5 Correlation Test

The test for correlation is done with the help of Pearson Correlation and environmental factors coefficient with correlation coefficient greater than 0.7 were excluded. The environmental variables finally selected are shown in Fig. 2 based on the coefficient values in Fig. 4. For any two pixels x and y between two corresponding raster images representing the same spatial domain of environmental variables in the current scenario, the Pearson correlation coefficient is calculated as per the below equation in Fig. 3. It is performed over the 390,258 pixels each representing 5.6 km² in the Indian subcontinent.

Environmental predictor	Code	Unit	Primary data source
Bioclimatic Variables(1970-2000)			
Annual mean temperature	biocl1	°C	Worldclim (Hansen and el.)
Mean Diurnal Range	biocl2	°C	Worldclim (Hansen and el.)
Temperature Seasonality	biocl4	%	Worldclim (Hansen and el.)
Annual Precipitation	biocl12	mm	Worldclim (Hansen and el.)
Precipitation seasonality	biocl15	%	Worldclim (Hansen and el.)
Land cover(2012)			
Mixed Forest	Mixed Forest	Proportion of cell area	MODIS MCD12C1
Deciduous Broadleaf	Deciduous Broadleaf	Proportion of cell area	MODIS MCD12C1
Open Shrubland	Open Shrubland	Proportion of cell area	MODIS MCD12C1
Woody Savanna	Woody Savanna	Proportion of cell area	MODIS MCD12C1
Cropland	Cropland	Proportion of cell area	MODIS MCD12C1
Evergreen Broadleaf	Evergreen Broadleaf	Proportion of cell area	MODIS MCD12C1
Urban	Urban	Proportion of cell area	MODIS MCD12C1
Population density (2020)	Population density	persons/ km ²	SEDAC
Elevation(2000)	Elevation	m	GTOP030
Global Human Footprint(2004)	GHF	Percentage	Last of the Wild, V2

Fig. 2 Environmental variables chosen after preprocessing and performing a correlation test

$$r = \frac{N \sum xy - (\sum x)(\sum y)}{\sqrt{[N \sum x^2 - (\sum x)^2][N \sum y^2 - (\sum y)^2]}}$$

Where N = number of pair of scores, $\sum xy$ = sum of the products of paired scores
 $\sum x$ = sum of x scores $\sum x^2$ = sum of x² scores
 $\sum y$ = sum of y scores $\sum y^2$ = sum of y² scores

Fig. 3 Formula for calculation of Pearson correlation coefficient where x and y represent any two pixels

	B1	B12	Cr1d	Def	ESF	GO30	HF	Os	Pd	WS
B1	1.00000000	0.42528652	0.62540550	0.11484995	0.145850514	0.13896451	0.04660219	0.05151043	0.18287629	0.30715357
B12	0.42528652	1.00000000	0.01682204	0.14118547	0.597951244	-0.51276503	0.33338969	-0.33054482	0.09854120	0.47931982
Cr1d	0.62540550	0.01682204	1.00000000	-0.04212255	-0.097994467	-0.20888261	-0.09311255	-0.07263524	0.26622702	-0.07742462
Def	0.11484995	0.14118547	0.04212255	1.00000000	0.003038760	-0.03364891	1.5051381	-0.02468341	-0.03507811	0.21929451
ESF	0.14585051	0.59795124	0.09799447	0.00303876	1.00000000	-0.03891621	0.03672597	-0.05006048	-0.08221077	0.06947487
GO30	0.13896451	-0.51276503	0.20888261	-0.03364891	-0.038916207	1.00000000	0.05536160	0.05847102	-0.22721055	-0.08111393
HF	0.04660219	0.33338969	0.09311255	0.15051381	0.036725965	0.05536160	1.00000000	-0.05918374	-0.07043501	0.19191061
Os	0.05151043	-0.33054482	0.07263524	-0.02468341	-0.050060479	0.05847102	-0.05918374	1.00000000	-0.09949359	-0.07070212
Pd	0.18287629	0.09854120	0.26622702	-0.03507811	-0.082210773	-0.22721055	-0.07043501	-0.09949359	1.00000000	-0.06601678
WS	0.30715357	0.47931982	0.07742462	0.21929451	0.069474872	-0.08111393	0.19191061	-0.07070212	-0.06601678	1.00000000

Fig. 4 Correlation between environmental variables after removing those with a correlation coefficient greater than 0.70

4 Models and Result

Modeling was done using machine learning algorithms mainly random forest, artificial neural networks, and support vector machines along with Maxent based on the literature survey done. Since there may yet exist some spatial sampling bias hence spatial partition was implemented using Block scheme where $k = 4$ whose results are shown in Fig. 7. The maximum number of iterations was set to be 500. This was

implemented using ENMeval. Based on longitudes and latitudes, the spatial extent is divided into four. The presence points were also divided randomly into 75% training (36) and 25% testing (12) in another instance. 10,000 background points were randomly chosen. Performance is measured through AUC and Kappa statistic which is calculated as shown in Figs. 5 and 6 ROC is calculated as the ratio between sensitivity and 1—specificity (False positive rate).

Maxent—Maxent model performs better in accuracy and prediction than other approaches which already exist as conveyed by Elith et al. [10]; Phillips et al. [2]. The maximum entropy model calculates the training gain for each of the variable for the 48 presence points which are split into training and testing subsamples. The gain for each variable here represents the maximum information extracted from each factor keeping others constant. This is done by jackknife as shown in Fig. 8. For Maxent, the AUC indicated a high degree of accuracy and for training, it was measured to be 0.922 while test AUC is 0.963. This is based on sensitivity which represents whether the position is favorable for Panthera Tigris Tigris to occur when environmental factors are present (true positive rate) and similarly the specificity which here represents the true negative rate. Figure 7 indicates the predicted geographical range with red being the highest probability of occurrence and blue with no occurrence. The contribution of each variable as per the gain is also calculated and shown in Fig. 8.

Random Forest—As per Petitpierre et al. [11], RF has one of the highest accuracy among the machine learning models available for the majority of the datasets regardless of its size. It can be implemented for both presence and pseudo-absence

Test	Present	n	Absent	n	Total
Positive	True Positive (TP)	a	False Positive (FP)	c	a + c
Negative	False Negative (FN)	b	True Negative (TN)	d	b + d
Total		a + b		c + d	

	Positive	Negative	Total
Positive	(a+b)(a+c)/N	(a+b)(b+d)/N	a+b
Negative	(c+d)(a+c)/N	(c+d)(b+d)/N	c+d
	a+c	b+d	N

Fig. 5 Contingency table and expected frequency for each cell

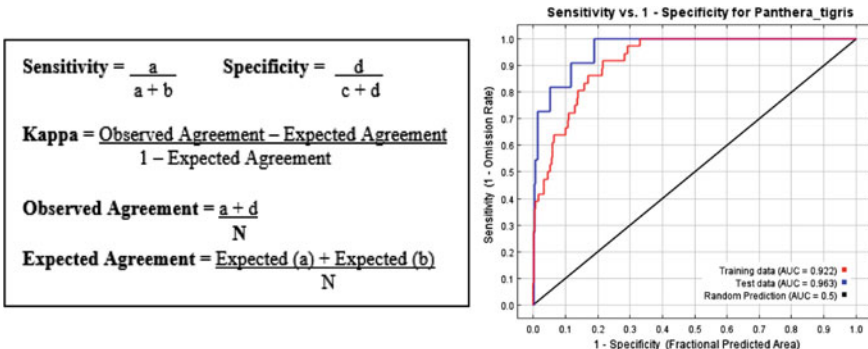


Fig. 6 Calculations for various performance measures and area under the receiver operating characteristic curve for 48 presence points (36 training points and 12 testing points) for Maxent

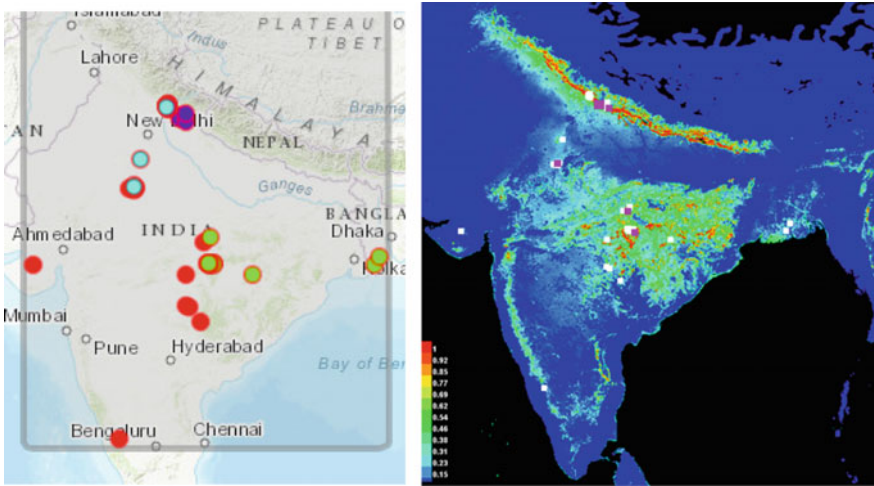


Fig. 7 Spatial partition using block ($k = 4$) and predicted potential geographical locations for Panthera tigris

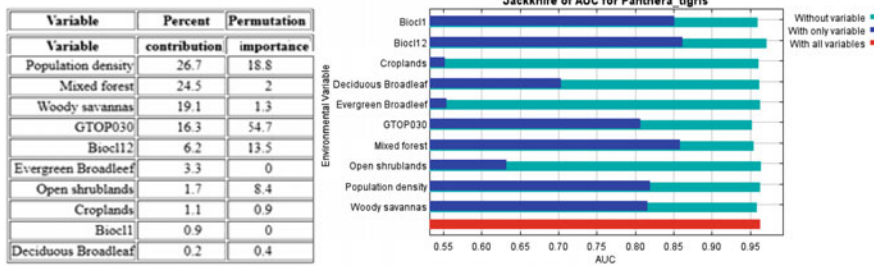


Fig. 8 Contribution of environmental variables and training gain for each variable (jackknife) for Maxent

data and corrects many of the flaws in classification and regression tree including overfitting. It also mentions the low reliability for a dataset consisting of categorical factors with varying levels. RF being a regression tree model which does classification, it comes to the result using various tree predictors in a forest. The AUC was found to be 0.93 along with the Kappa statistic at 0.86.

Artificial Neural Networks—These are biologically inspired models comprising of neurons and their corresponding weights. The AUC was found to be 0.906 along with the Kappa statistic at 0.809. Although ANN are easy to model for species with more presence locations yet it can be influenced by a large number of classes.

Support Vector Machines—These are multidimensional classifiers used for regression and classification through supervised learning. The AUC is 0.898 with a Kappa statistic of 0.795. Since they have a good generalization ability along being

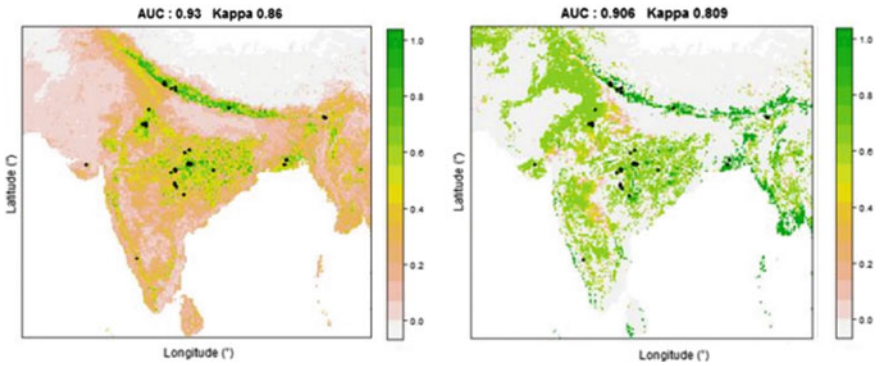


Fig. 9 Predicted geographic distribution using RF and ANN, respectively

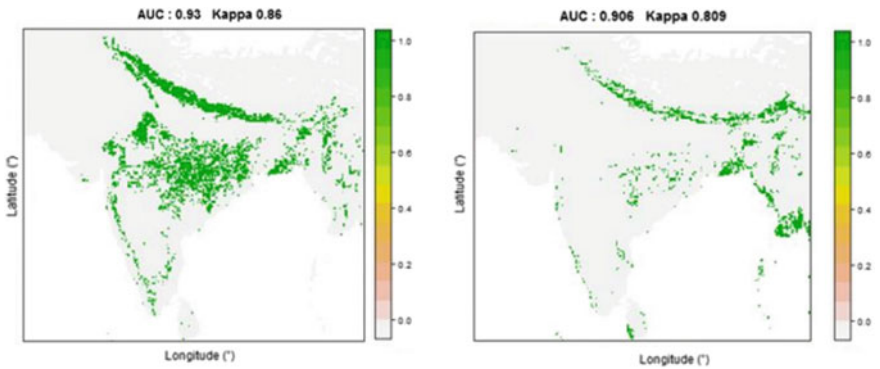


Fig. 10 Binary distribution for RF and ANN modeled on a threshold calculated using TSS

able to handle high dimensional data such as raster layers, SVM was chosen for modeling.

The potential geographical distribution for various models is shown in Figs. 9, 10, 11, 12. An ensemble model consisting of all the three models was also formed. The contribution of each of the variables was calculated using Pearson. To generate the binary distribution, sum of specificity and sensitivity as represented in Figs. 5 and 6 also known as TSS (True skill statistic) was taken to derive the threshold value as it represents the point of optimization under the receiver operating characteristic curve and is also found to minimize the mean error rate [12]. These represent locations representing potential predicted habitats in binary terms (Figs. 10, 12).

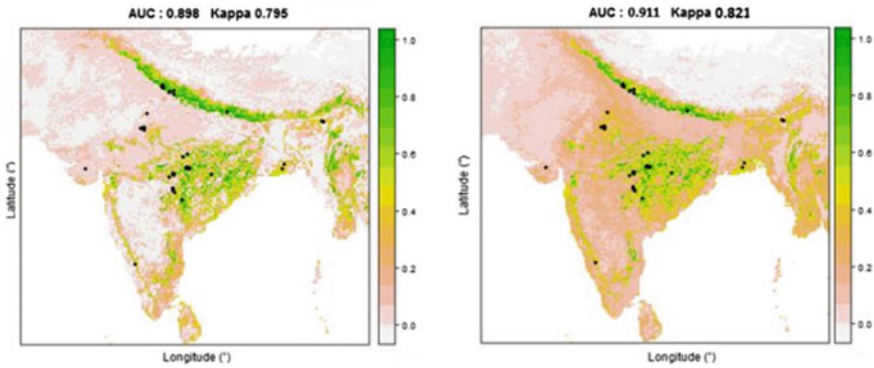


Fig. 11 Predicted geographic distribution using SVM and Ensemble modeling, respectively

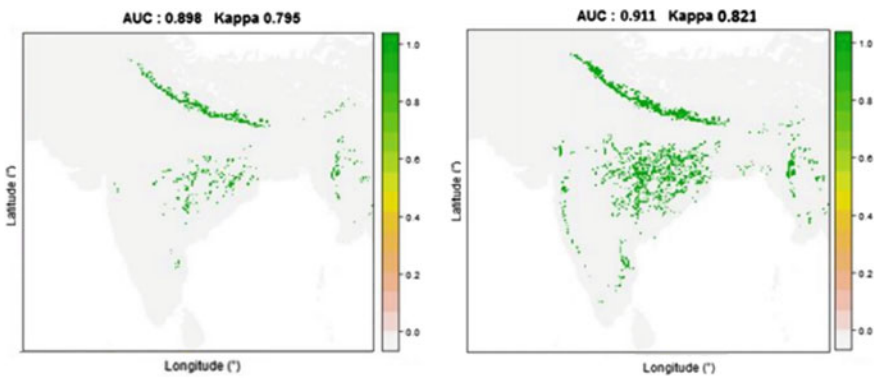


Fig. 12 Binary distribution for SVM and ensemble modeling modeled on a threshold calculated using TSS

5 Conclusion

The high AUC values of 0.963, 0.931, and 0.906 for Maxent, random forest, and ANN indicate a greater chance for ranking a positive instance which is chosen randomly by these classifiers as compared to a negative instance with the threshold being TSS. The Kappa statistic comes into play whenever we need to measure inter-rater agreement and reliability. From the above models, it can be concluded that the data collected is a true representation of the variables measured as it is above 0.80 in all these. It can be seen that population density plays a major role which shifts the prediction to unhabituated areas. Woody savannas which in India is represented by the Terai region of Uttarakhand, Uttar Pradesh, Bihar, and parts of Assam is also an important variable along with Mixed Forest as they provide suitable ecological habitat. Altitude plays a major role and lowers down the prediction range geographically. Annual precipitation and temperature together contribute around 14% on average. Human footprint has

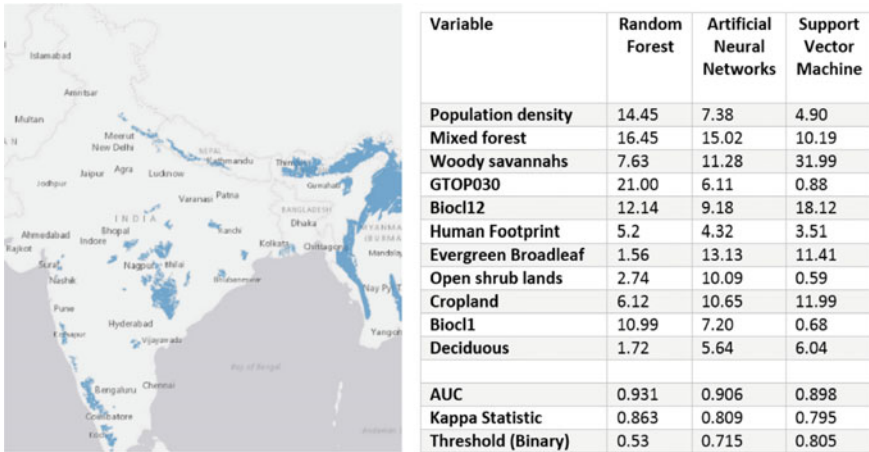


Fig. 13 Tiger conservation landscape (TCL) in the Indian subcontinent and variable contribution along with performance measures for RF, ANN, and SVM

little impact on prediction as compared to population density. Finally, shrublands, evergreen, and deciduous forests provide suitable conditions for prey capture and hence are important.

Also, it can be seen that the predicted areas lie in and around the tiger conservation landscape (TCL) zones as declared by WWF (Fig. 13). These are found basically in Madhya Pradesh, Uttarakhand and parts of Rajasthan, and a large area of predicted zones by all the models lie under these. Maxent provided a high level of AUC at 0.964 and was chosen since it is independent of the number of occurrences and performs better than various other models. Care was taken to proceed with correct practices for MAXENT as mentioned by Yackulic et al. [13]. RF has one of the highest accuracy while SVM performs generalization well although accuracy is lower. The above results can be used to study regions around the present tiger reserves in India as they will provide locations suitable for reintroduction as per environmental factors free from human influence. Each pixel of the raster images represents an area of around 5.6 km². These geographical locations can then be used either as a part of ecological corridors to prevent genetic isolation or as independent reserves. Although several methods exist for modeling the distribution of various species yet there exist many limitations. Resampling usually involves predicting cell values from the existing through various techniques such as bilinear, averaging, etc., which may lead to decrease in accuracy since each cell value is depicting the predicted value rather than the actual value. The factors incorporating and describing the habitat and needs of the particular species must be included after proper study of the requirement of the species. Future work requires testing on an extensive dataset, mapping the predicted geographical locations with an actual land survey and comparing with other present models so as to strengthen and make use of the results of this research.

References

1. Grinnell, J. (1917). The niche-relationships of the California thrasher. *Auk*, 34, 427–433.
2. Phillips, S. J., Anderson, R. P., & Schapire, R. E. (2006). Maximum entropy modeling of species geographic distributions. *Ecological Modelling*, 190, 231–259.
3. Thatte, P., Joshi, A., Vaidyanathan, S., Landguth, E., & Ramakrishnan, U. (2018). Maintaining tiger connectivity and minimizing extinction into the next century: Insights from landscape genetics and spatially-explicit simulations. *Biological Conservation*, 218, 181–191.
4. Kywe, T. Z. (2012). Habitat suitability modeling for tiger (*Panthera tigris*) in the Hukaung Valley Tiger Reserve, Northern Myanmar, Niedersächsische Staats (157 pp.). Germany: Nund Universitätsbibliothek Göttingen.
5. Hernandez, P. A., Graham, C. H., Master, L. L., & Albert, D. L. (2006). The effect of sample size and species characteristics on performance of different species distribution modeling methods. *Ecography*, 5, 773–785.
6. Kulloli, R. N., & Kumar, S. (2014). Comparison of Bioclimatic, NDVI and elevation variables in assessing extent of *Commiphora wightii* (Arnt.) Bhand. *ISPRS—International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-8, 589–595.
7. Duan, R.-Y., Kong, X.-Q., Huang, M.-Y., Fan, W.-Y., & Wang, Z.-G. (2014). The predictive performance and stability of six species distribution models. *PLoS ONE*, 9(11), e112764. <https://doi.org/10.1371/journal.pone.0112764>.
8. Qing, Z., Zhang, Y., Sun, G., Duo, H., Wen, L., & Lei, G. (2015). Using species distribution model to estimate the wintering population size of the endangered scaly-sided merganser in China. *PLoS ONE*, 10, e0117307. <https://doi.org/10.1371/journal.pone.0117307>.
9. Hansen, J., Sato, M., Ruedy, R., Lacis, A., & Oinas, V. (2000). Global warming in the twenty-first century: An alternative scenario. *Proceedings of the National Academy of Sciences*, 97, 9875–9880. <https://doi.org/10.1073/pnas.170278997>.
10. Elith, J., Graham, C. H., Anderson, R. P., et al. (2006). Novel methods improve prediction of species' distributions from occurrence data. *Ecography*, 29, 129–151.
11. Petitpierre, B., Kueffer, C., Broennimann, O., Randin, C., Daehler, C., et al. (2012). Climatic niche shifts are rare among terrestrial plant invaders. *Science*, 335, 1344–1348.
12. Freeman, E. A., & Moisen, G. G. (2008). A comparison of the performance of threshold criteria for binary classification in terms of predicted prevalence and kappa. *Ecological Modelling*, 217, 48–58.
13. Yackulic, C. B., et al. (2013). Presence-only modelling using MAXENT: When can we trust the inferences? *Methods in Ecology and Evolution*, 4, 236–243.
14. Adhikari, D., Barik, S. K., & Upadhaya, K. (2012). Habitat distribution modelling for reintroduction of *Ilex khasiana* Purk., a critically endangered tree species of northeastern India. *Ecological Engineering*, 40, 37–43.
15. Elith, J. H., Graham, C. P., Anderson, R., Dudík, M., Ferrier, S., et al. (2006). Novel methods improve prediction of species' distributions from occurrence data. *Ecography*, 29, 129–151.

Organizational Digital Footprint for Traceability, Provenance Approach



Sheetal Arya, Kumar Abhishek and Akshay Deepak

Abstract In this age of instant noodles, instant messaging, and tweeting, we have all we want at the tips of our finger whether it be the delivery of a food product, buying a property, or going for a date. All but a click away. As quoted, “The control of information is something successful person does.” If one could control the information, he would be able to make worthy decisions. Now the question arises, how does one know which information is relevant and which is fake, for making the foundation of decisions? Here comes the role of provenance. In this paper, the characteristics of a data such as why, when, where, what, how, and who are determined using PROV family of documents, giving full structural format to data and determining all its characteristics and its existential source. Now all this provenance information describing all aspects of data when connected will generate a footprint. Apache NiFi has been used to show the flow of data and its manipulation. The digital footprint as it is all in digital format can be traceable to determine data trustworthiness, i.e., its provenance, hence determining the reliability of an organization for merger, acquisition, data analysis, stock market, etc.

Keywords Big data · Provenance · Digital footprint · Knowledge representation

1 Introduction

As noticed by E. O. Wilson, “We are drowning in information while starving for wisdom.” It could be seen there is too much information available but irrelevant. This big data of real-time information, i.e., raw data, could be used to seek queries, suggestion, and solutions, i.e., meaningful data, knowledge. The more relevant the

S. Arya (✉) · K. Abhishek · A. Deepak
Department of Computer Science and Engineering, NIT Patna, Patna 800005, Bihar, India
e-mail: sheetal.shatabdi@gmail.com

K. Abhishek
e-mail: kumar.abhishek@nitp.ac.in

A. Deepak
e-mail: akshayd@nitp.ac.in

© Springer Nature Singapore Pte Ltd. 2019
N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_23

trustworthy knowledge one has, the more reliable the decisions one would be able to make; hence come provenance, semantics, Apache NiFi, and digital footprint in the picture.

In Sect. 2, the related works have been described which lead to the invention of the proposed system model. In Sect. 3, how provenance has been used in this model has been described. In Sect. 4, the proposed workflow of the system has been described. In Sect. 5, the methodology has described which consists of dataset used, Sect. 5.1 describes the implementation of data flow and PROV-N, and finally, Sect. 5.2 describes digital footprint formation, i.e., result. In Sect. 6, the major achievements and limitations are described. Section 7, finally, consists of conclusion and future scope.

2 Related Works

The 14 papers used in this paper for provenance data implementation are summarized in Table 1 having references [1–13]. Table 1 consists of five columns, where **column 1** represents serial number. **Column 2** represents the year in which paper was published. **Column 3** represents the purpose of the published paper. **Column 4** represents the scope of data, which can be used in the implementation; sometimes, *format* is written as data scope because paper simply describes the syntax/format for implementation purpose. **Column 5** represents the concept described in the paper and the algorithm used in the implementation of the framework.

3 Provenance

According to the dictionary, provenance is “The place of origin or earliest known history of something” [14]. The structure of provenance data model is represented in Fig. 1. Arrows are read in the backward direction, such as an entity was used by an Activity, etc. In this diagram, symbols are used to represent attributes, and properties are represented by arrowheads. An *Entity* is used to represent an object, which is a source of information. An *Activity* is processed, which are involved with data manipulation, i.e., converting data from one form to another. An *Agent* represents the living or nonliving object responsible for initiating processes for data manipulation. There are various other core and extended, attributes and properties of the provenance data model [1].

The granularity with which provenance of a data is to be maintained is determined according to the requirement of provenance information for implementation, because it could be as specifically defined as you want it to be. Yet again if that much granularity is not required, then it is useless.

Table 1 Literature survey

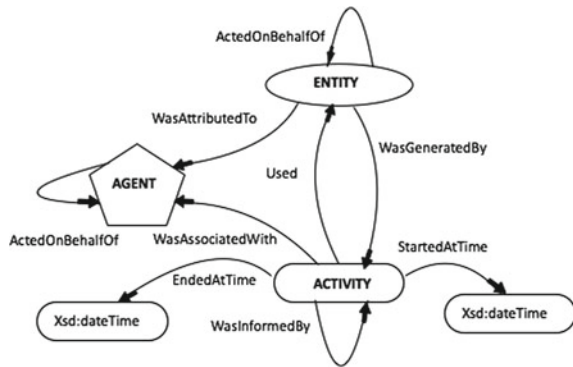
S. No.	Year of publication	Purpose	Data scope	Concepts/ algorithms used
1	2013	PROV introduction	Format	Provenance introduction, summary is described
2	2013	Ontology for provenance	Web, format	Provenance ontology syntax for provenance data model is described
3	2013	Data model for provenance	Web, format	Structural format for attributes and properties is described
4	2013	Notations for provenance	Web, format	Notational representation of provenance data model is described
5	2013	Constraints for provenance	Web, format	Implementation of constraints is described
6	2013	Accessing and querying of provenance	Web, format	Accessing and querying mechanism for provenance is described
7	2013	XML schema for provenance	Web, format	XML implementation of provenance is described
8	2013	Dictionary representation of provenance	Web, format	Collective structural representation of provenance is described
9	2013	Dublin core terms representation of provenance	Web, format	Mapping to DC terms vocabulary is described
10	2013	Semantics for provenance	Web, format	Semantics implementation in first-order logic for provenance is described

(continued)

Table 1 (continued)

S. No.	Year of publication	Purpose	Data scope	Concepts/ algorithms used
11	2013	Link connectivity for provenance	Web, format	Connectivity of different bundles of provenance data using PRO LINKS is described
12	2010	Capturing and usage of provenance	Web	Framework, models for provenance implementation
13	2016	A blockchain ontology for supply chain provenance	Supply chain, blockchain, web pages	Blockchain, solidity, language, ethereum, TOVE traceability ontology

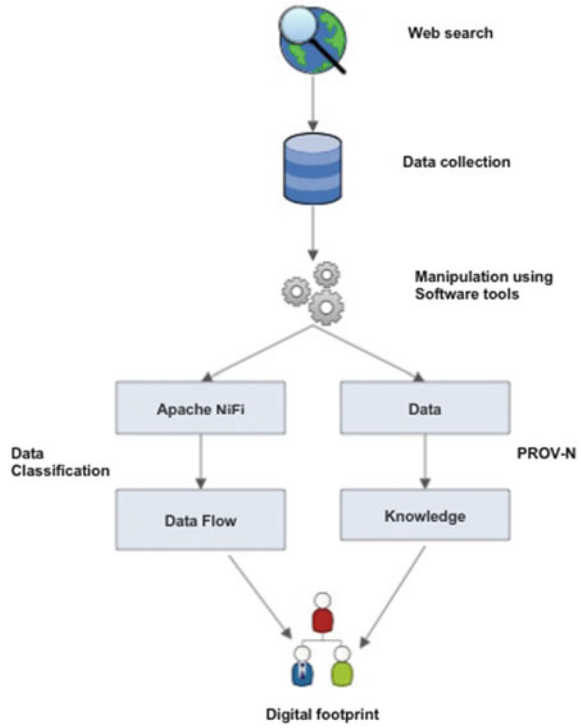
Fig. 1 Provenance data model



4 Proposed System Workflow

The steps to be followed for fetching digital footprint are determined here. First, data is searched from websites, repositories, and Wikipedia about organizations in CSV format. That data is collected in Microsoft Excel files. This classified data is first imported in Apache NiFi [15]. In NiFi, processors are used, connections are established, and flow files are derived for final data flow derivation. NiFi will give us data flow of information from different sectors of organizations. The data collected will be written in provenance notation PROV-N deriving knowledge. Thus, finally Fig. 2 shows, the workflow of digital footprint created using visualization tools.

Fig. 2 Workflow diagram



5 Methodology

In methodology, the technical implementation part of the paper is described. Here, the datasets used, software tools used, implementation criteria used, and the final result will be described. The dataset used for this work has been taken from the following website:

- Wikipedia [16],
- Quandl [17], and
- Google [18].

5.1 Implementation

In this paper, finding provenance of petroleum companies' framework has been divided into two phases as follows:

Phase I: Dataflow using Apache NiFi

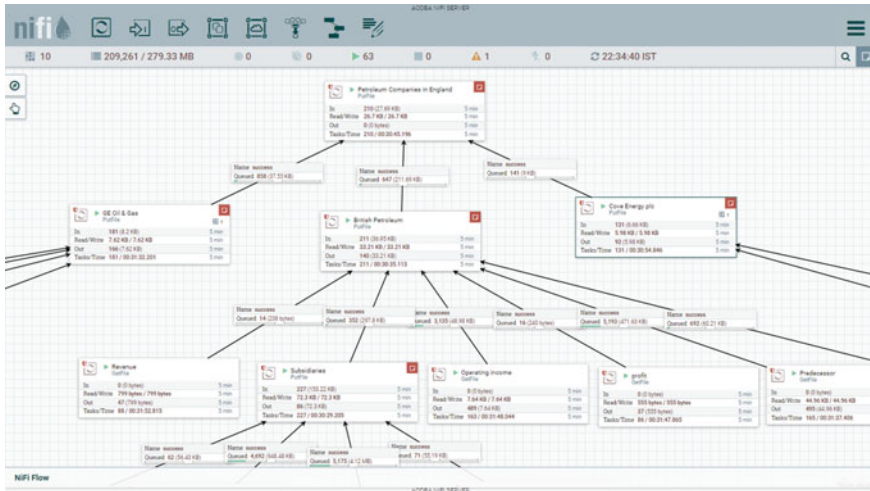


Fig. 3 Data flow of all collected information from distributed organizations to forming flow of data of petroleum companies in England

Information (such as location, revenue, acquisition, etc.) of petroleum companies in England is collected from sources such as Wikipedia, Quandl, etc. The information obtained is treated as flow files, and a data flow of these files is generated in Apache NiFi as shown below. Through this data flow, information is transferred through different levels. As shown, processors *British Petroleum*, *GE Oil & Gas*, and *Cove Energy Plc*. represent information collected by respective petroleum companies, headquartered in England, whereas sub-processors such as processor *Revenue* fetch the data of revenue generated by the company, processor *Subsidiaries* collects information relating to subsidiaries of a company, processor *Operating income* describes dataset of income on which the company operates, processor *Profit* describes the data of a company profits, processor *Predecessor* describes data of company that was replaced and now is known as mentioned company, processor *Parent* describes the parent company of the present mentioned company, processor *Acquisition* describes datasets of the companies that were acquired by the mentioned company, and processor *Successor* describes the company which has acquired the present mentioned company.

All the information collected in these workspaces (*GE Oil & Gas*, *British Petroleum*, *Cove Energy plc*) is transferred to “Eng_Petroleum” workspace, which is represented by processor *Petroleum Companies in England*. So, all information of these three petroleum companies is stored in *Petroleum Companies in England* processor as shown in Fig. 3. Also, in the workflow, information details at each level can be found.

British Petroleum gets information from processor *Revenue*, *Subsidiaries*, *Operating income*, and *Profit*. It has four subsidiaries located in *Pakistan*, *Serbia*, *New*

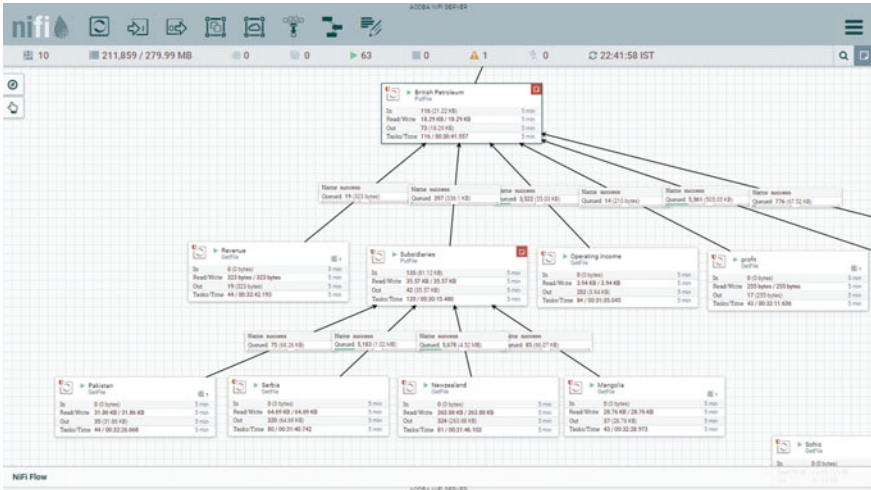


Fig. 4 Data flow of British petroleum from its attributes

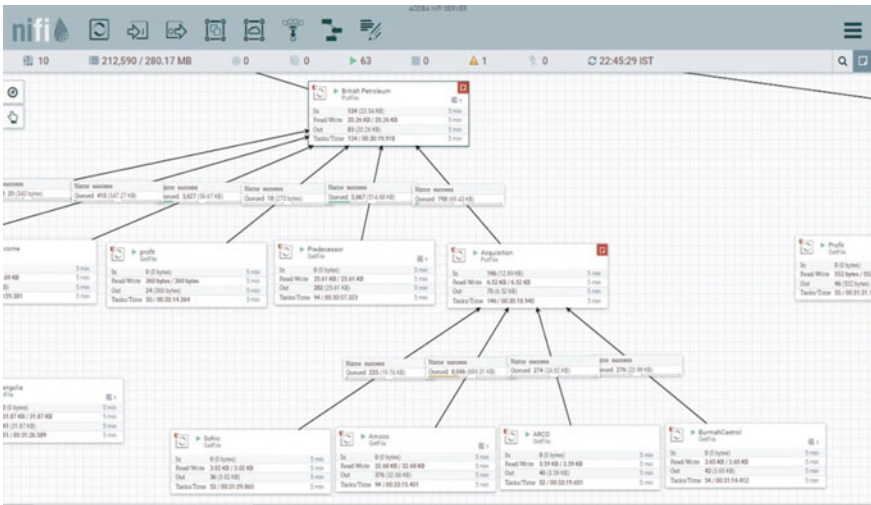


Fig. 5 Data flow of British petroleum further from its acquisition

Zealand, and Mangolia. So, processor *Subsidiaries* gets its information from processors named after its subsidiaries as shown in Fig. 4.

British Petroleum has also acquired four companies named *Sohio*, *Amoco*, *ARCO*, and *Burmah Castrol*. Information of all these acquisitions described by a respective processor is collected by the *Acquisition* processor as shown in Fig. 5.

Similarly, information related to GE Oil & Gas is transferred to processor *GE Oil & Gas*, and information related to company Cove Energy Plc. is collected by processor *Cove Energy plc*.

Phase II: Knowledge for digital footprint formation using provenance

Metadata formation using provenance notation PROV-N: PROV-N is a W3C recommended notation for provenance model. It is invented to represent information about the data of the PROV data model as per these implementation principles:

- Readability for humans: Provenance notations determine syntax which is human-readable and machine understandable so it can be used for illustration of examples.
- Independence of technology: Provenance notations provides format so simple that it can easily be mapped to several technologies.
- Formal representation: Provenance notations are defined in a formal grammar representation which is acceptable to be used with parser generators.

In phase II, we identify entities, activities, agents, and relationships among them through data flow generated in phase II, and write code using PROV-N syntax to obtain the provenance metadata, describing knowledge in ontologies, hence showing digital footprint in visualization generated.

5.2 Result

The following figures in this section describe visualization generated using provenance notation PROV-N, describing digital footprint generated by European Petroleum Organizations. All the petroleum industries in England data are collected in a directory, which is represented by **entity** *Petroleum Companies in England* as shown in Fig. 6.

The **agents** are organizations attributed to **entities** *BP*, *Cove*, *GE* which are represented by data directories as shown above. The **activities** *SendInfo1*, *SendInfo2*, etc., are associated with **agent** *ApacheNiFi* as it is basically the backbone to collect and transfer all data information. The **activity** *SendInfo8* used information from **entities** *BP*, *Cove*, *GE* to **generate** **entity** *PetroleumCompaniesInEngland*.

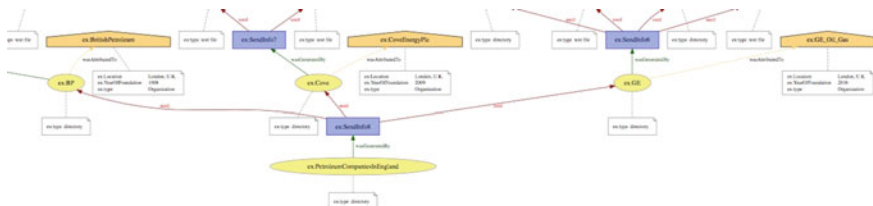


Fig. 6 Digital footprint generated by Europe petroleum industries



Fig. 7 Digital footprint generated by British petroleum describing its attributes

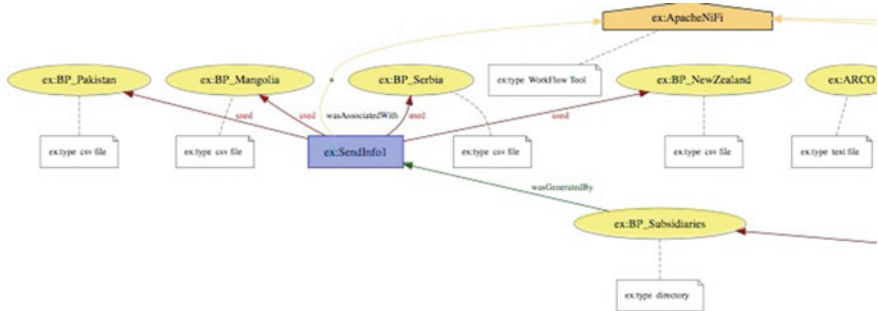


Fig. 8 Digital footprint generated by British petroleum further describing its subsidiaries

The **entity** *BP used* information from **entities** *BP_Predecessor*, *BP_Profit*, *BP_Revenue*, *BP_Operating_Income*, *BP_Acquisition*, and *BP_Subidiaries*. **Entities** *ARCO*, *BurmahCastrol*, *Sohio*, *Amoco* represent the information of companies that have been acquired by the company British Petroleum. So, information from these **entities** has been **used** by **activity** *SendInfo2* to **generate** entity *BP_Acquisition* as shown in Fig. 7.

Similarly, British Petroleum industry has subsidiaries in Pakistan, Magnolia, Serbia, and New Zealand and their collected data are represented by **entities** *BP_Pakistan*, *BP_Mangolia*, *BP_Serbia*, and *BP_NewZealand*. These **entities** are **used** by **activity** *SendInfo1* to **generate** entity *BP_Subidiaries*. Hence, *BP_Subidiaries* represents information relating to subsidiaries of British Petroleum as shown in Fig. 8.

The *Cove_Profit* **entity** was **used** by **activity** *SendInfo7* to **generate** entity *Cove*, which is finally **attributed** to similarly for *Cove_Profit* **entity** which is **attributed** to **agent** *CoveEnergyPlc* an organization and for **entity** *GE* that is attributed to **agent** *GE_Oil_Gas* also an organization. Hence as could be seen, this is all a digital footprint representation of companies catching all milestones of their journey.

6 Major Achievements and Limitation

The limitation discovered from recent papers is that the data about organizations is not available publicly; hence, further granularity is not possible easily.

The major achievements are that Trustworthiness of Organization are fetched, which has many benefits such as lower cost of transactions as valid decisions could be made for reducing expenditure. Apart from reliability, digital footprint could also be used to validate data quality, i.e., increasing automatic data processing and analyzing using scientific workflows could be influenced for automated collection, processing, and analysis of provenance information. The provenance metadata generating digital footprint will be used to track how market information is modified and used across the financial system.

7 Conclusion and Future Scope

This paper presented a model for data flow and digital footprint using provenance from W3C Provenance family of documents and related papers. The 14 published papers were summarized using tables; they were used in contributing to model for digital footprint formation. After going through these papers, it is clear that the enhancements of provenance are an open field for research in a financial institution, as where else authorization, validation, and trustworthiness are more required other than where money is involved.

Paper have explained, provenance approach for deriving dataflow and additionally creating a digital footprint for tracing of required information for merger, acquisition, subdivision, and other financial matters. This knowledge will help in determining the source of default or identifying whether a subsidiary is working efficiently or not, when to merge two departments, during acquisition how perfect decisions could be made by derived knowledge, etc.

In the next phase of this data flow, using open-source Java API for OWL Ontologies and other API implementation is proposed. So that automatic implementation of the model could be done. Hence, automatic retrieval of data from sources could be done and their footprint will be derived, without manual labor for any domain.

References

1. PROV-PRIMER. (2013). *World wide web consortium W3C*. Retrieved from: <https://www.w3.org/TR/2013/NOTE-prov-primer-20130430/>.
2. PROV-O: The PROV Ontology. (2013). *World Wide web consortium W3C*. Retrieved from <https://www.w3.org/TR/2013/REC-prov-o-20130430/>.

3. PROV-DM: The PROV Data Model. (2013). *World Wide web consortium W3C*. Retrieved from <https://www.w3.org/TR/2013/REC-prov-dm-20130430/>.
4. PROV-N: The PROV Notation. (2013). *World Wide web consortium W3C*. Retrieved from <https://www.w3.org/TR/2013/REC-prov-n-20130430/>.
5. PROV-CONSTRAINTS: The PROV Constraints. (2013). *World Wide web consortium W3C*. Retrieved from <https://www.w3.org/TR/2013/REC-prov-constraints-20130430/>.
6. PROV-AQ: The PROV Access and Query. (2013). *World Wide web consortium W3C*. Retrieved from <https://www.w3.org/TR/2013/NOTE-prov-aq-20130430/>.
7. PROV-XML: The PROV XML Schema. (2013). *World Wide web consortium W3C*. Retrieved from <https://www.w3.org/TR/2013/NOTE-prov-xml-20130430/>.
8. PROV-DICTIONARY: The PROV Dictionary. (2013). *World Wide web consortium W3C*. Retrieved from <https://www.w3.org/TR/2013/NOTE-prov-dictionary-20130430/>.
9. PROV-SEM: The PROV Semantics. (2013). *World Wide web consortium W3C*. Retrieved from <https://www.w3.org/TR/2013/NOTE-prov-sem-20130430/>.
10. PROV-LINKS: The PROV Link Across Bundles. (2013). *World Wide web consortium W3C*. Retrieved from <https://www.w3.org/TR/2013/NOTE-prov-links-20130430/>.
11. Allen, A., & David, M. et al. (2010). *Provenance capture and use: A practical guide*. The MITRE Corporation.
12. Kim, H. M., & Laskowski, M. (2016). Towards an ontology-driven blockchain design for supply chain provenance.
13. Sheth, A., & Kapanipathi, P. (2016). Semantic filtering for social data. *IEEE Internet Computing*, 20(4), 74–78. Retrieved from <http://ieeexplore.ieee.org/abstract/document/7529038/>.
14. *Dictionary Meaning of provenance*. Retrieved from <https://en.oxforddictionaries.com/definition/provenance>.
15. *Apache NiFi overview*. Retrieved from <https://nifi.apache.org/docs.html>.
16. Wikipedia. Retrieved from https://en.wikipedia.org/wiki/GE_Oil_and_Gas.
17. Quandl. *Financial data repository*. Retrieved from <https://www.quandl.com>.
18. Google browser. Retrieved from <https://www.google.co.in/>.

Bidirectional Long Short-Term Memory for Automatic English to Kannada Back-Transliteration



B. S. Sowmya Lakshmi and B. R. Shambhavi

Abstract Transliteration is the key component in various Natural Language Processing (NLP) tasks. Transliteration is the process of converting one orthographic system to another. This paper demonstrates transliteration of Romanized Kannada words to Kannada script. Our system utilizes a bilingual corpus of around one lakh words, which comprise pairs of Romanized Kannada word with its corresponding word in Kannada script and employs orthographic and phonetic information. Recurrent Neural Networks (RNNs) are widely used Neural Networking model for text and speech processing as they better predict the next word based on past information. Long Short-Term Memory (LSTM) Networks are exceptional kind of RNNs which handles long-term dependencies. A Character level Bidirectional Long Short-Term Memory (BLSTM) paradigm which drives down the perplexity with respect to word-level paradigm has been employed. Knowledge of Characters uncovers structural (dis)similarities among words, thus refining the modeling of uncommon and unknown words. Test data of 3000 Romanized Kannada words is used for model evaluation and we obtained an accuracy of 83.32%.

Keywords Transliteration · Bilingual corpus · RNN, LSTM

1 Introduction

Transliteration is the task of mapping graphemes or phonemes of one language into phoneme approximations of another language. It has got various applications in the domain of NLP like Machine Translation (MT), Cross Language Information Retrieval (CLIR), and information extraction. Even though the task appears trivial, prediction of pronunciation of the original word is a crucial factor in transliteration process. Transliteration process among two languages is minimum if they share

B. S. Sowmya Lakshmi (✉) · B. R. Shambhavi
Department of ISE, BMS College of Engineering, Bangalore 560019, India
e-mail: sowmyalakshmibs.ise@bmsce.ac.in

B. R. Shambhavi
e-mail: shambhavibr.ise@bmsce.ac.in

© Springer Nature Singapore Pte Ltd. 2019
N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_24

identical alphabet set. However, for languages which practice nonidentical set of alphabets, words have to be transliterated or portrayed in the native language alphabets.

Majority of multilingual web users have a tendency to represent their native languages in Roman script on social media platforms. In spite of many recognized transliteration standards, there is an intense inclination to use unofficial transliteration standards in many websites, social media, and blog sites. There are ample of issues such as spelling variation, diphthongs, doubled letters, and reoccurring constructions which are to be taken care while transcribing.

Neural Networks is a rapidly advancing approach to machine learning [1, 2] and has shown promising performance when applied to a variety of tasks like image recognition, speech processing, natural language processing, cognitive modeling, and so on. It involves using neural networks for training a model for a specific task. This paper demonstrates the application of neural network for machine transliteration of English–Kannada, two linguistically distant and widely spoken languages.

The rest of this paper is arranged as follows. Section 2 describes prior work in this area. An introduction to LSTM and BLSTM is described in Sect. 3. The methodology adopted to build corpus is presented in Sect. 3. Proposed transliteration network is portrayed in Sect. 4. Section 5 provides details of results obtained. Section 6 communicates conclusion and future work of the proposed method.

2 Previous Research

Research on Indic languages within the perspective of social media is quite ample, with numerous studies concentrating on code-switching has become a quite familiar phenomenon. There are a few substantial works being done on Transliteration or, more precisely, back-transliteration of Indic languages [3, 4]. A shared task which included, back-transliteration of Romanized Indic language words to its native scripts was run in 2014 [5, 6]. In many areas, including machine transliteration, end-to-end deep learning models have become a good alternative to more traditional statistical approaches. A Deep Belief Network (DBN) was developed to transliterate from English to Tamil with restricted corpus [7] and obtained an accuracy of 79.46%. A character level attention-based encoder in deep learning was proposed to develop a transliteration model for English–Persian [8]. The model presented a good accuracy, with BLEU score of 76.4.

In [9], authors proposed transliteration of English to Malayalam using phonemes. English–Malayalam pronunciation dictionary was used to map English graphemes to Malayalam phonemes. Performance of the model was fairly good for phonemes in pronunciation dictionary. However, it suffered from out-of-vocabulary issue when a word is not in pronunciation dictionary.

The most essential requisite of transliterator is to retain the phonetic structure of source language after transliterating in target language. Different transliteration techniques for Indian languages were proposed. In [10], input text was split into

phonemes and was classified using Support Vector Machine (SVM) algorithm. Most of the methods adopted features like n -grams [11], Unicode mapping [12], or a combination-based approach by combining phoneme extraction and n -grams [13, 14].

Antony et al. [15–17] have proposed Named Entities (NE) transliteration techniques from English to Kannada. In [15, 16], authors adopted a statistical approach using widely available tools such as Mosses and Giza++ which yielded an accuracy of about 89.27% for English names. System was also evaluated by comparing with Google transliterator. A training corpus of 40,000 Named Entities was used to train SVM algorithm [17] and obtained an accuracy of 87% for 1000 test dataset.

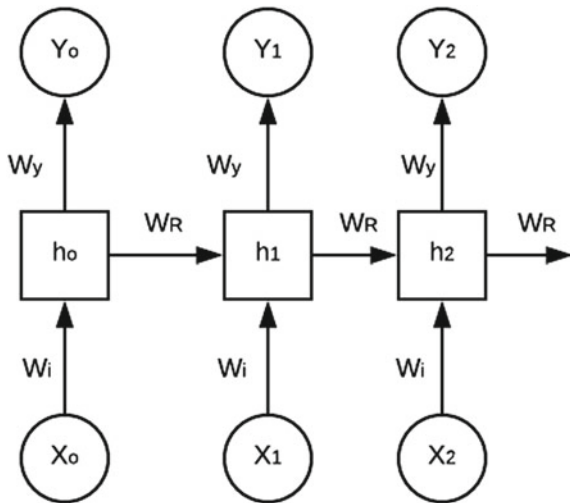
3 LSTM and BLSTM Network

RNNs have been employed to produce promising results on a variety of tasks including language model [18] and speech recognition. A RNN foresees present output based on the preserved memories of its past information. RNNs are designed for capturing information from sequences or time series data.

RNN networks [19] comprises an input layer, hidden layer, and an output layer where each cell preserves a memory of previous time. Figure 1 demonstrates a simple RNN model where X_0, X_1, X_2 are inputs at timestamps t_0, t_1, t_2 and hidden layer units are h_0, h_1, h_2 :

The new state (h^t) of RNN at time t is a function of its previous state at time $t - 1$ (h^{t-1}) and the input at time t (x^t). Output (y^t) of the hidden layer units at time t are

Fig. 1 A simple RNN model



calculated using new state calculated and the weight matrix. The maths behind RNN to calculate output from hidden and output layers are as follows:

$$h^{(t)} = g_h(W_i X^{(t)} + W_R h^{(t-1)} + b_h) \tag{1}$$

$$Y^{(t)} = g_y(W_Y h^{(t)} + b_y) \tag{2}$$

where W_Y , W_R and W_i are weights which are to be calculated during training phase, g_h and g_y are activation functions computed using Eqs. (3) and (4) respectively and b_h and b_y are bias. RNN uses backpropagation algorithm, but it is applied for every time stamp. It is commonly known as Backpropagation Through Timestamp (BTT). The dimensionality of output layer is same as labels and also characterizes likelihood distribution of labels at time t .

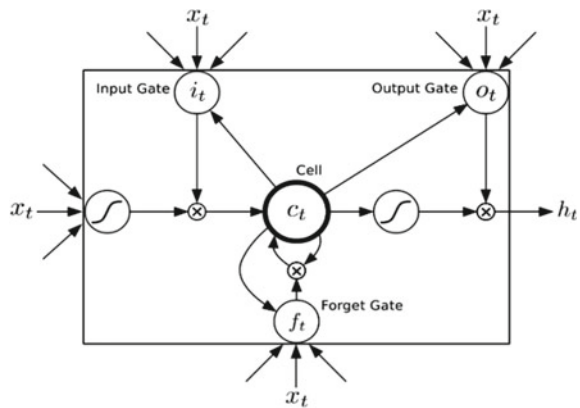
$$g_h = \frac{1}{1 + e^{-z}} \tag{3}$$

$$g_y = \frac{e^{z^m}}{\sum_k e^{z^k}} \tag{4}$$

where g_h is a sigmoid function and g_y is softmax activation function which maps input to the output nonlinearly.

LSTM Networks are special kind of RNNs and these RNNs are accomplished to learn long-term dependencies. Consider a language model of text prediction, which predicts a next word based on the previous word. In order to predict German as the last word in the sentence “I grew up in Germany. I speak fluent German” recent information suggests that next word might probably be the language but in order to narrow down the language context of German is needed which is quite long back in the sentence. This is known as long-term dependencies. LSTMs are capable of handling this type of dependencies where RNNs fail. Figure 2 shows a LSTM cell.

Fig. 2 A LSTM cell



where

σ = logistic sigmoid function

i = input gate

f = forget gate

o = output gate

c = cell vectors

h = hidden vector

W = weight matrix

LSTM is implemented as the following:

- Primary step in the LSTM is to decide the data to be neglected from the cell state which is made by forget gate layer. This decision is made by a sigmoid layer called the “forget gate layer”. It is a function of h_{t-1} and x_t as shown in Eq. (5), and outputs a number between 0 and 1 for each number in the cell state c_{t-1} . A 1 represents “completely keep this” while a 0 represents “completely get rid of this.”

$$f_t = \sigma(W_f[h_{t-1}, X_t] + b_f) \tag{5}$$

- The second step is to decide the new data to be stored. This has two steps, input obtained from the previous timestamp and the new input are passed through a sigmoid function called “input gate layer” to get i_t as shown in Eq. (6). Next, input obtained from the previous timestamp and the new input are passed through a tanh function. Both the steps are combined with f_t passed from the previous step as in Eq. (7).

$$i_t = \sigma(W_i[h_{t-1}, X_t] + b_i) \tag{6}$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_c[h_{t-1}, X_t] + b_c) \tag{7}$$

- Last step is to obtain output using Eqs. (8) and (9), which is based on cell state. First, a sigmoid layer decides what parts of the cell state are to be outputted. Then, the tanh function pushes cell state values between -1 to 1 , which is multiplied by the output of the sigmoid gate, so that only decided parts happen to be the output.

$$o_t = \sigma(W_o[h_{t-1}, X_t] + b_o) \tag{8}$$

$$o_t = \sigma(W_o[h_{t-1}, X_t] + b_o) \tag{9}$$

3.1 BLSTM Network

Sequence learning task requires previous and forthcoming input features at time t . Hence, BLSTM network is used to utilize previous features and upcoming features at a given time t . BLSTM [20] network hidden layer contains a sequence of forward and backward recurrent neural components connected to the identical output layer. Figure 3 shows a simple BLSTM network of four input units X_0 to X_4 . Network hidden layer has four recurrent components h_0 to h_3 in the forward direction and four recurrent components h_0 to h_3 in the backward direction to help predict output Y_0 to Y_3 by forming an acyclic graph. Most of the text processing task BLSTM would provide reasonable results in the prediction of sequence of data.

4 Dataset Collection

To develop a transliteration model using neural networks necessitates a significant amount of bilingual parallel corpus. For resource-poor languages like Kannada, it is hard to obtain or build a large corpus for NLP applications. Therefore, we built a training corpus of around 100,000 English–Kannada bilingual words. Bilingual words were collected from following various sources.

- Various websites were scraped to collect most familiar Kannada words and their Romanized words. Special characters, punctuation marks, and numerals were removed by preprocessing. This data contributed around 20% of the training data.

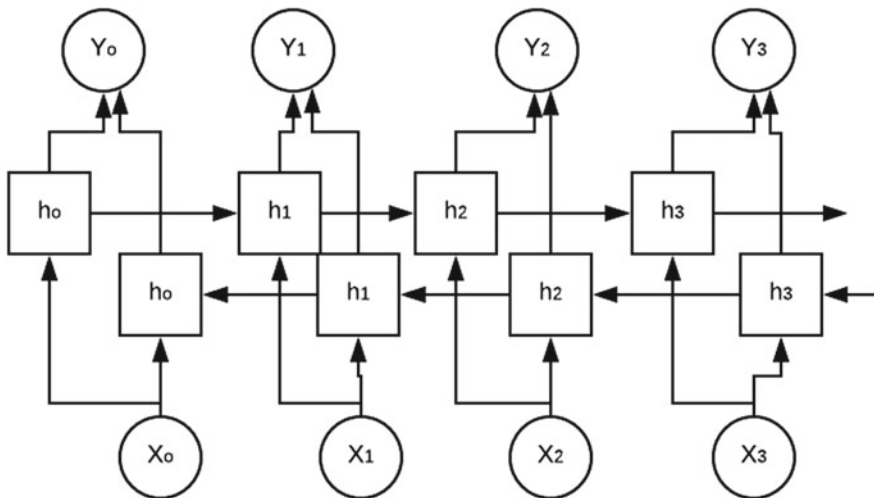


Fig. 3 A BLSTM network

- Majority of the bilingual words were collected from music lyrics websites which consist of song lyrics in Kannada script and its corresponding song lyrics in Romanized Kannada. Non-Kannada words, punctuations, and vocalize words in song lyrics were removed. Obtained list comprehends viable syllable patterns in Kannada and contributed around 70% of the training data.
- The subsequent share of the corpus was manually transliterated NEs.

5 Experiments

5.1 Setup

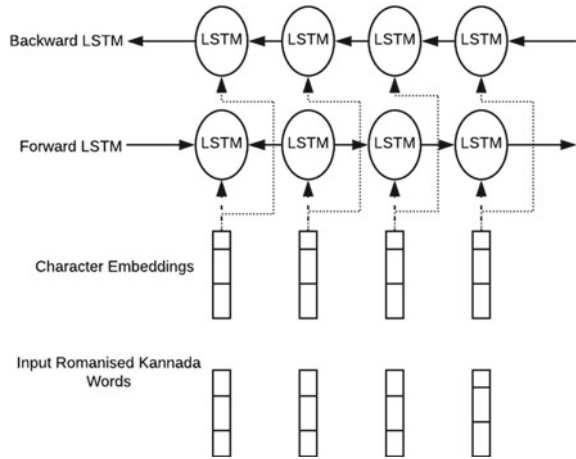
The proposed approach is implemented on python platform and packages used are numpy and neural network toolkit keras with Tensorflow in the backend. The network parameters are set up as in Table 1.

5.2 Training Procedure

The proposed model is implemented using simple BLSTM network as shown in Fig. 4. Paradigm is trained with the collected dataset which contained bilingual corpus Romanized Kannada words and its corresponding word in Kannada script. During Training, 20% of the training data is set as the validation data. Algorithm 1 describes network training procedure. In each epoch, entire training data is divided into batches and one batch is processed at given time t . Batch size determines number of words to be included in a batch. Characters in each input word are embedded and provided as input to forward and backward state of LSTM. Later, we backpropagate the errors from the output to the input to update the network parameters.

Table 1 Model parameters

Parameter	Value
No. of epochs	30
Batch size	128
Hidden units	128
Embedding dimension	64
Validation split	0.2
Output activation function	Softmax
Learning rate	0.001
Training model	Bidirectional LSTM

Fig. 4 BLSTM network**Algorithm 1** BLSTM Model Training Procedure

```

for each epoch do
  for each batch do
    1) bidirectional LSTM model forward pass:
      forward pass for forward state LSTM
      forward pass for backward state LSTM
    2) bidirectional LSTM model backward pass:
      backward pass for forward state LSTM
      backward pass for backward state LSTM
    3) update parameters
  end for
end for

```

6 Results

Model was tested for the dataset of around 3 K words collected from random websites. Test dataset contains Romanized words and its transliterated words in Kannada script which is kept as reference to compare with the result.

Snapshot of results obtained from the model is shown in Table 2. The correctness of the transliteration is measured by Accuracy (ACC) or Word Error Rate (WAR) yielded by a transliteration model. For completeness, other transliteration results obtained by RNN and LSTM networks which are trained for the same datasets are reported in Table 3.

Table 2 Snapshot of results

Romanized Kannada word	Gold standard transliterated word	Resultant word	Transliteration result
Tirugu	ತಿರುಗು	ತಿರುಗು	Correct
Setuve	ಸೇತುವೆ	ಸೇತುವೆ	Correct
Aggalikeya	ಅಗ್ಗಲಿಕೆಯ	ಅಗ್ಗಲಿಕೆಯ	Correct
Sadbhava	ಸದ್ಭಾವ	ಸಾಧಭವ	Incorrect
Anaupacharika	ಅನೌಪಚಾರಿಕ	ಅನೌಪಾಾರರರ	Incorrect

Table 3 Evaluation results

Model	Accuracy obtained (%)
RNN	74.33
LSTM	79.76
BLSTM	83.32

7 Conclusion and Future Work

Transliteration is the task of mapping graphemes or phonemes of one language into phoneme approximations of another language. It is the elementary step for most of the NLP applications like MT, CLIR, and text mining. English and Kannada language trail dissimilar scripts and also vary in their phonetics. Furthermore, Romanization of Kannada words does not go along a standard pattern as far as their pronunciation is concerned. Thus, a particular set of rules do not guarantee an effective back-transliteration. In this paper, we have presented a Transliteration model for English–Kannada language pair.

A character level BLSTM model was investigated, which utilizes character embedding for words. Along with BLSTM, model was also tested for LSTM and RNN for English and Kannada. The correctness of the transliteration was measured by Accuracy for a test data of 3000 Romanized Kannada words. As BLSTM has two networks, one access information in the forward direction and another access in the reverse direction, the output generated is from both the past and future context. Accuracy obtained by BLSTM was more when compared to LSTM and RNN for this test data.

There are several possible courses for future improvement. First, model would be further progressed by combining other algorithms with BLSTM. For example, combination of CNN with BLSTM or CRF with BLSTM would yield better results. Another improvement is to expand the training data by collecting data from other domains such as social media (Twitter and Weibo) which would include all pos-

sible orthographic variations. Since model is not restricted to specific domain or knowledge, social media text would also provide a fair share in training data.

References

1. Kalchbrenner, N., & Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.
2. Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*.
3. Sharma, A., & Rattan, D. (2017). Machine transliteration for indian languages: A review. *International Journal*, 8(8).
4. Dhore, M. L., Dhore, R. M., Rathod, P. H. (2015). Survey on machine transliteration and machine learning models. *International Journal on Natural Language Computing (IJNLC)*, 4(2).
5. Sequiera, R. D., Rao, S. S., Shambhavi, B. R. (2014). Word-level language identification and back transliteration of romanized text: A shared task report by BMSCE. In *Shared Task System Description in MSRI FIRE Working Notes*.
6. Choudhury, M. et al. (2014). Overview of fire 2014 track on transliterated search. *Proceedings of FIRE*, 68–89.
7. Sanjanaashree, P. (2014). Joint layer based deep learning framework for bilingual machine transliteration. In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE.
8. Mahsuli, M. M., & Safabakhsh, R. (2017). English to Persian transliteration using attention-based approach in deep learning. In *2017 Iranian Conference on Electrical Engineering (ICEE)*. IEEE.
9. Sunitha, C., & Jaya, A. (2015). A phoneme based model for english to malayalam transliteration. In *2015 International Conference on Innovation Information in Computing Technologies (ICIICT)*. IEEE.
10. Rathod, P. H., Dhore, M. L., & Dhore, R. M. (2013). Hindi and Marathi to English machine transliteration using SVM. *International Journal on Natural Language Computing*, 2(4), 55–71.
11. Jindal, S. (2015, May). N-gram machine translation system for English to Punjabi transliteration. *International Journal of Advances in Electronics and Computer Science*, 2(5). ISSN 2393-2835.
12. AL-Farjat, A. H. (2012). Automatic transliteration among indic scripts using code mapping formula. *European Scientific Journal (ESJ)*, 8(11).
13. Dasgupta, T., Sinha, M., & Basu, A. (2013). A joint source channel model for the English to Bengali back transliteration. In *Mining intelligence and knowledge exploration* (pp. 751-760). Cham: Springer.
14. Dhindsa, B. K., & Sharma, D. V. (2017). English to Hindi transliteration system using combination-based approach. *International Journal*, 8(8).
15. Antony, P. J., Ajith, V. P., & Soman, K. P. (2010). Statistical method for English to Kannada transliteration. In *Information Processing and Management* (pp. 356–362). Berlin, Heidelberg: Springer.
16. Reddy, M. V., & Hanumanthappa, M. (2011). English to Kannada/Telugu name transliteration in CLIR: A statistical Approach. *International Journal of Machine Intelligence*, 3(4).
17. Antony, P. J., Ajith, V. P., & Soman, K. P. (2010). Kernel method for English to Kannada transliteration. In *2010 International Conference on Recent Trends in Information, Telecommunication and Computing (ITC)*. IEEE.

18. Mikolov, T. et al. (2011). Extensions of recurrent neural network language model. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.
19. Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*(2), 179–211.
20. Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673–2681.

A Dominant Point-Based Algorithm for Finding Multiple Longest Common Subsequences in Comparative Genomics



Manish M. Motghare and Preeti S. Voditel

Abstract Finding the longest common subsequence is a classic and well-studied problem in the field of computer science and considered as an NP-hard problem. There are many application of LCS in the field of bioinformatics, computational genomics, image processing, file comparison, etc. There are many algorithms are present to find the similarity between the given strings and its special cases. As there is a tremendous increase in the biological data and it requires an efficient mechanism to deal with them, many efforts have been taken to reduce the time and space complexity of the given problem. In this paper, we presented a novel algorithm for the general case of multiple LCS problems, i.e., finding a longest common subsequence in the given two strings. Our algorithm works on dominant point approach to compute the LCS of the given string. When applied to multiple strings of length each 1000, 2000, 3000, 4000, and 5000, characters, it is found that our algorithm works two or three magnitude faster than existing algorithm and it requires less space compared to existing algorithms.

Keywords Longest common subsequence (LCS) · Dynamic programming · Np-hard problem · Dominant point · Problem complexity · Comparative genomics

1 Introduction

In biological computation, we need to compare two or more strings, in order to find a similarity between them. This is usually done in finding similarity between two organisms by comparing their DNA. A DNA comprises several molecules called bases. The DNA bases are adenine, cytosine, guanine, and thymine. All of these bases are represented as {A, C, G, T}, respectively. For example, we are having DNA of two organisms, let 'X = ACGGTGTCGTGCTATGCTGATGCTGACTTATATGCTA' denote DNA of one organism and 'Y = CGTTCCGGCTATCGTACGTTTC-

M. M. Motghare (✉) · P. S. Voditel
Department of Computer Application, Shri Ramdeobaba College of Engineering and Management, Nagpur, India
e-mail: motgharemm@rknc.edu

TATTCTATGATTTCTAA' denote DNA of other. We can say two organisms are similar to each other if one's DNA is a substring of another, but in this case, neither strings are a substring of each other. So another method to find similarity between the two organisms can be determined by generating a third string which is formed from the given two strings, the new string will contain common subsequence between them, it is necessary the bases of DNA will be in same organized order, but not possibly consecutively. In this case, the common subsequence between X and Y is $Z = \text{'CGTTCGGCTATGCTTCTACTTATTCTA'}$. The longer the common subsequence, the more the similarity between the given DNA strands. But up to what extent they are similar can be determined based on longest common subsequence. A subsequence is a part of a huge sequence which can be obtained from another by removing one or more elements without changing the order. For example, the sequence $\{C, B, C, E, D, B, G\}$ is obtained from the $\{A, C, B, D, E, G, C, E, D, B, G\}$ [1].

Table 1 shows the list of common structures in computational biology and their sizes [2]. The LCS algorithms are widely used in computational biology as well as in many traditional applications like file comparison, data compression, and FPGA synthesis.

The longest common subsequence (LCS) is a standard and well-studied problem of computer science. It is an NP-Hard problem with many applications in image processing, computational genomics, bioinformatics, etc [11–13]. Over many years, significant efforts have been made to find efficient algorithm to solve the problem of finding the longest common subsequence in the multiple strings, but the present algorithms having limitation as they work on only few cases of MLCS [14–17], or the problem's special case of two or three strings [18, 19]. Several methods have been proposed over general strings [19–22], and they could greatly be benefitted if there is improvement in computational time of the algorithms. As mentioned this method of MLCS [23–26] can be applied in bioinformatics and computational genomics which deals with biological data. Due to tremendous increase in biological data and widespread use of sequence analysis tools, we expect the usage of MLCS method in Computational genomics and their applications.

In this paper, we present an algorithm which is fast and space efficient for the multiple longest common subsequences (MLCS) problem. The algorithm works on larger strings and returns a longest common subsequence faster than existing algo-

Table 1 A details of biological data and its component

Biological data	Alphabet (Σ)	$ \Sigma $	Typical sequence length
DNA	{A C G T}	4	10^4 – 10^{11} [3–6]
RNA	{A C G U}	4	10 – 10^4 [7]
Genome	{gene ₁ , gene ₂ ...gene _k }	10^4	10 – 10^4 [8, 2]
Protein	{A, C, ..., W}	20	10^2 – 10^4 [9, 10]

rithms. Our algorithm works on the dominant point approach [18, 19, 25]. Procedure which uses a dominant point method is more efficient, which results in reducing the size of search space compared to other existing algorithm [19]. A dominant point represents matching character in the row, or simply k -dominant or dominant at level k and it is represented by circle. The key idea behind dominant point method is to recognize matching values of all levels in a matrix M , instead of all values in matrix [14, 26, 27]. In our algorithm, we are filling the values in the matrix only if match is found, i.e., “1”, reducing the space required to store other values. The place where the value “1” is printed is called dominant point. In this, there is no requirement to store the value of pointers to trace the longest common subsequence, result in saving of time. Whereas LCS based on dynamic programming require trace-back approach, which follows the arrow backwards, starting from the last entry of the matrix [28].

The paper is organized as follows: In the next section, we discussed some of the existing policies and related work. Later, we will discuss our new algorithm and its complexity and last we will compare our algorithm with other existing algorithm.

2 Problem Formulation and Existing Methods

In this paragraph, we will discuss existing techniques and number of ways to find MLCS of a given string.

2.1 Problem Definition

Definition 1 The longest common subsequence is defined as follows. Consider the string X which is of length ‘ L ’ and another string Y of length ‘ M ’. The main goal is to find the longest common subsequence present in the given two strings. The subsequence will be calculated from left to right order, which can be derived from another sequence by deleting some elements without changing the order of remaining elements. The longer the subsequence the more the similarity in between the strings.

Let us consider the following strings:

$$X = T G A C$$

$$Y = A G C T$$

In this case, longest common subsequence has length 2, i.e., subsequence ‘AC’. An alternative approach to look at this is that we are finding 1–1 matching between the character of X and Y strings.

Definition 2 Let $A = \{a_1, a_2 \dots a_n\}$ be a set of alphabets over Σ , then $B = \{b_1, b_2 \dots b_n\}$, are the subsequence generated from ‘ A ’, then multiple longest common

subsequence can be considered only if, b is generated from set 'A' without changing the order of other elements.

The subsequence 'b' is the longest sequence of all the available sequences present in 'A', satisfying point mentioned above.

For the given strings, there can be one or more subsequence will be present. Consider the given strings $X = \{A, B, C, B, D, A, B\}$ and $Y = \{B, D, C, A, B, A\}$, then the common subsequence of both the strings could be $Z = \{BDAB, BDA, BD, B\}$. So, the longest common subsequence present in the multiple string is $Z = BDAB$. For multiple longest common subsequences, finding LCS is declared as its special case.

2.2 Dynamic Programming Methods

The traditional method for multiple longest common subsequence is based on dynamic programming approach [17]. In this approach, two sequences are given n_1 and n_2 each of length L_1 and L_2 , respectively, the algorithm maintains two matrixes, one for storing the length of the string, i.e., 'b' and second matrix, i.e., c to store few pointers in a parallel array, $c[0\dots m, 0\dots n]$ value for particular i th row and j th column. Let us consider a string of protein sequences as characters. Given two sequences $L_1 = \{AATCCGCTAG\}$ and $L_2 = \{AAACCCTTAG\}$, the motivation is to compute the LCS of the two whole strings. In this, we will compute the LCS for every pair of prefixes and will store the value in $C[i][j]$. In the dynamic programming approach, we have come up with three observations.

First Approach: $b[i, 0] = b[j, 0] = 0$. If out of the given two sequences, any string found to be empty then LCS in between the string will be empty.

Second Approach: Assume $X_i = Y_j$. If the last characters of the given two sequences are same, then the LCS must end with same character. Hence $X_i = Y_j$ then $b[i, j] = b[i - 1, j - 1] + 1$.

Third Approach: Assume $X_i \neq Y_j$. In this approach, both X_i and Y_j will not appear in the longest common subsequence. From this, we can conclude either X_i is not a part of the LCS or Y_j . Hence, $X_i \neq Y_j$. Then $b[i, j] = \max(b[i - 1, j], b[i, j - 1])$.

By applying the dynamic programming approach, we get the matrix 'b'. The length of the longest common subsequence is calculated from the end point of the matrix $b[i, j]$ (Fig. 1).

3 Related Work

Since last decade, numerous algorithms have been proposed to find the longest common subsequence of the multiple strings, and most of them are based on dynamic

X/Y		A	B	C	B	D	A	B
		0	0	0	0	0	0	0
B	0	0 ↑	1 ↘	1 ←	1 ↘	1 ←	1 ←	1 ↘
D	0	0 ↑	1 ↑	1 ↑	1 ↑	2 ↘	2 ←	2 ←
C	0	0 ↑	1 ↑	2 ↘	2 ←	2 ↑	2 ↑	2 ↑
A	0	1 ↘	1 ↑	2 ↑	2 ↑	2 ↑	3 ↘	3 ←
B	0	1 ↑	2 ↘	2 ↑	3 ↘	3 ←	3 ←	4 ↘
A	0	1 ↘	2 ↑	2 ↑	3 ↑	3 ↑	4 ↘	4 ↑

Fig. 1 The LCS is calculated using all the three approaches of dynamic programming mentioned above. The MLCS present in the above string is “BDAB”. Arrows represent pointers which are stored in matrix ‘c’, which helps in trace-back of matching sequence from the end point, i.e., $b[i][j]$ to $b[0][0]$. The dynamic programming approach having time complexity of $O(m + n)$, as at least one of the ‘i’, or ‘j’, decrements at each stage. The resultant algorithm is having space complexity of $O(nl)$, for ‘l’, sequence of length ‘n’. Currently, there are many algorithms [14–16] present for finding the longest common sequence, but they are having large complexity and require more memory than our proposed algorithm

programming. But due to tremendous increase in the dataset of biological data, the algorithm which is based on dynamic programming scale badly and found to be less space and time efficient when dealing with larger strings to compare. Due to this, the focus has been shifted to heuristic techniques which are capable to deal with large data sets around gigabytes of data which are produced during genome sequencing [29]. The algorithm developed by Hirschberg [30] or Irving and Fraser [31], having complexity of $O(l_n)$, which shows they are exponential in nature. Many improvements have been done to reduce this complexity to $O(l_{n-1})$, like reducing the space by calculating dominant points [32] which are used in practice. Parallelization of this algorithm has been done by Chen et al. [33], i.e., Fast_LCS and Wang et al. [34], by running algorithm over multiple cores of processor and distributing the work among them. The implementation of this algorithm work over 10 strings ($n = 10$), with more strings their runtime becomes impractical. Various approximation algorithms like long run proposed by Jiang and Li [35] which create a longest string containing only a single character that is valid subsequence in all strings there which finds the optimal solution to NP-hard problems with reduced time complexity [36] as compare to heuristics approach, they have better solution and runtime. Heuristics algorithm does not provide good solution and runtime but they provide solution quickly which

is “good enough”. Jones et al. [37, 38] and Bryant [39, 40] proposed a GA algorithm for finding the longest common subsequence of an arbitrary string. They had done comparison of genetic algorithm with Fraser and Irving which are based on dynamic programming [31] and reported a shorter runtime for genetic algorithm.

4 New Fast_LCS Algorithm

Our algorithm works on the principal of point dominant approach. In our proposed algorithm LCS is calculated in row major order. Figure 2, shows a pseudocode for our proposed algorithm. The algorithm works parallel, i.e., it starts comparing first row and first column value of the said matrix, in row major order and also prints the value in the output array. Following observations are associated with the algorithm and combining all the observations jointly we have the following rule:

Rule 1: $C[i, j] = “1”$, if $x_i = y_j$, stop scanning for particular row and column and increments the value $C[x_i + 1, y + 1]$.

Rule 2: $C[i, j] = “0”$, if $x_i \neq y_j$, increment the value in row major order till the end of the row.

```

LCS-LENGTH(A, B)
BEGIN
    set m to length of String A
    set n to length of String B
    let C[[1...m],[1...n]] be Array()
    let LCS be String
    set i to 0
    set j to 0
    // Loop Counters
    FOR each letter element in Row A String
        FOR each letter element in Row B String
            IF Row letter of A String equals with Column letter of B String THEN // Comparing
                Strings
                    set Array(row, column) to 1
                    set LCS = LCS + A_i // Storing Matching LCS in LCS String
                    IF RowCount less than length of String A // Incrementing to Next Row and
                NextColumn
                    INCREMENT RowCount
                END IF
            ELSE
                set C[[i],[j]] := "0"
            END IF
        END FOR
    END FOR
    PRINT LCS
END

```

Fig. 2 Pseudocode of dominant MLCS algorithm

4.1 Quick Version of Algorithm to Find MLCS

In this section, we present a new memory-efficient dominant point-based algorithm for finding multiple longest common subsequence (Fig. 3).

First Step: Compare the first letter of ‘Y’ axis with all character of ‘X’ axis in row major order.

- a. Compare the first letter ‘B’ of Y-axis with first letter ‘A’ of X-axis, as there is no match, put value “0”.
- b. Compare the first letter ‘B’ of Y-axis with second letter ‘B’ of X-axis, as there is a match found, put value “1”.
- c. Compare the first letter ‘B’ of Y-axis with third letter ‘C’ of X-axis, as there is no match, put value “0”.
- d. Compare the first letter ‘B’ of Y-axis with fourth letter ‘B’ of X-axis, as there is a match found, put value “1”.
- e. Compare the first letter ‘B’ of Y-axis with fifth letter ‘D’ of X-axis, as there is no match, put value “0”.
- f. Compare the first letter ‘B’ of Y-axis with sixth letter ‘A’ of X-axis, as there is no match, put value “0”.
- g. Compare the first letter ‘B’ of Y-axis with seventh letter ‘B’ of X-axis, as there is a match found, put value “1”.
- h. The process will be same for the next letter ‘D’, ‘C’, ‘A’, ‘B’, and ‘A’.

Fig. 3 The matrix for two sequences $X = ABCBDAB$ and $Y = BDCABA$, the LCS found in the given string is “BDAB”, over $\sum = |L|$

X/Y	A	B	C	B	D	A	B
B	0	1	0	1	0	0	1
D	0	0	0	0	1	0	0
C	0	0	1	0	0	0	0
A	1	0	0	0	0	1	0
B	0	1	0	1	0	0	1
A	1	0	0	0	0	1	0

4.2 Computing Longest Common Subsequence

Step 1: The algorithm will search for match in the first row, if match is found, it will print the value of the matching character, i.e., “1”. And then it will stop searching for next matching character in that row. Now LCS is “B”.

Step 2: Again the algorithm will search for matching pair in the second row, but this time it will not consider that row and column from where first match pair was found and also it will not consider the left column values for the first matching pair while searching for next computation. In the second row, we are getting another matching pair so it will print “1”. Now LCS at this moment is “BD”.

Step 3: Discard Value of Row and Column after 2nd step.

Step 4: Search for matching pair at row “3”, here match between “C-C”, is found, i.e., value ‘1’, but this value falls in the left column of second matching pair of row, so this matching value will not be considered. Current LCS value is “BD”.

Step 5: Search for matching pair at fourth row, here match between “A-A” is found, so it will print value ‘1’, and LCS will be “BDA”. This time it will not consider upper row values and left column values.

Step 6: Search fifth row for matching pair, here two matches are found, but first matching pair will not be considered because it comes before the left column of the first selection, so it will be discarded. Then, next match is found, i.e., last “B-B”, whose value is ‘1’, so it will be considered. Now algorithm will print LCS “BDAB”.

Step 7: This will be the final Multiple LCS for the given sequence given in Table 3.

5 Results

In our experiment, the algorithm is executed in a system having Intel Core i5-2330 M Processor 2.20 GHz Windows 10 Home Basic (64-bit), Memory 4 GB/Hard Disk Drive 720 GB. The execution programming environment is GNU C++. The algorithm is tested on strings of length 1000, 2000, 3000, 4000, and 5000, each consists of four characters {A, C, G, T}, i.e., a nucleic acid sequence.

5.1 Dominant Point Multiple Longest Common Subsequence Algorithm

The developed algorithm is compared with Quick-DP [20] and Hakata et al. A and C algorithms [18]. The C algorithm proposed by Hakata et al. is designed for any number of strings. The A algorithm works for three strings and so far it is the fastest algorithm for three sequences. The QUICK-DP is a generalized algorithm which is developed to work with any number of sequences [20]. We implemented both algorithms as mentioned in their paper and the same environment was created for

Fig. 4 Longest common subsequence “BDAB” is found between the sequences

X/Y	A	B	C	B	D	A	B
B	0	1	0	1	0	0	1
D	0	0	0	0	1	0	0
C	0	0	1	0	0	0	0
A	1	0	0	0	0	1	0
B	0	1	0	1	0	0	1
A	1	0	0	0	0	1	0

execution of algorithm. As multiple longest common subsequence algorithms is having many application from finding of similarity among the biological sequences, computational genomics to other than biological domain, for every application there is a different representation of dataset. In our experiment, we considered a database which is generated from the four characters of nucleotide sequence, i.e., {A, C, G, T} and each dataset consists of sequence of length 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 2000, 3000, 4000, and 5000. In Table 1 and Fig. 4, the comparison is shown among Quick-Dp, Hakata and Imai algorithm, and our D_MLCS algorithm (Tables 2 and 3).

6 Conclusion

In this paper, we have proposed a new algorithm for finding the longest common subsequence, which is applied to both biological sequences which are randomly generated and real biological sequence and it is found that it is working efficiently compared to the existing algorithm with less time and space complexity. Recently, there is a tremendous change found in the study of protein and genomic sequences [20, 27], so our algorithm development and improvement will be on the basis of requirement by the bioinformatics community. In the current implementation of this algorithm, we focused on finding the longest common subsequence on genomic and protein sequences, each consists of combination of $\sum = 4$ and $\sum = 20$ characters, respectively. So in our future implementations of this algorithm, we will try to handle larger protein families. Our contribution in this paper is designing of new, efficient algorithm and its comparison with the existing algorithm.

Table 2 Comparison is done among the latest algorithm and our implementation mentioned above in terms of avg. running time of algorithm in seconds by giving inputs of 15 available random sequences of different size where input character is $\Sigma = |4|$

Sequence length	Quick-DP [28]	Hakata and Imai A algorithm [19]	Hakata and Imai C algorithm [19]	Our implementation of algorithm
100	0.0	0.6	1.7	0.03
200	0.1	4.2	6.6	0.04
300	0.3	13.0	26.0	0.14
400	0.8	62.4	71.5	0.16
500	1.3	174.2	203.3	0.17
600	NA	NA	560.3	0.18
700	NA	NA	NA	0.20
800	NA	NA	NA	0.22
900	NA	NA	NA	0.30
1000	NA	NA	NA	0.35
2000	NA	NA	NA	1.13
3000	NA	NA	NA	2.37
4000	NA	NA	NA	4.45
5000	NA	NA	NA	6.83

Table 3 Implementation results of dominant LCS algorithm

S. No.	X array sequence	Y array sequence	Time (s)	Processor			RAM	RAM type
				Execution time	Name	Speed		
1	100	100	0.03	Intel core i5-2330 M	2.2 GHz	2	4 GB	DDR3
2	200	200	0.04					
3	300	300	0.14					
4	400	400	0.16					
5	500	500	0.17					
6	600	600	0.18					
7	700	700	0.20					
8	800	800	0.22					
9	900	900	0.30					
10	1000	1000	0.35					
11	2000	2000	1.13					
12	3000	3000	2.37					
13	4000	4000	4.45					
14	5000	5000	6.83					

References

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. (2009). *Introduction to algorithm* (3rd ed.). The MIT Press.
2. Nekrutenko, A., & Li, W.-H. (2001). Transposable elements are found in alarge number of human protein-coding genes. *Trends in Genetics*, 17(11), 619–621.
3. Gregory, T. R. (2005). *Animal genome size database*. Retrieved from <http://www.Genomesize.com>.
4. Lodish, H. F. (2003). *Molecular cell biology*. WH Freeman.
5. Paterson, M., Dančik, V. (1994). Longest common subsequence's. In *Proceedings of the 19th International Symposium on Mathematical Foundations of Computer Science* (pp. 127–142). Springer.
6. Fortnow, L. (2009). The status of the P versus NP problem. *Communications of the ACM*, 52(9), 78–86.
7. Elbashir, S. M., Harborth, J., Lendeckel, W., Yalcin, A., Weber, K., & Tuschl, T. (2001). Duplexes of 21-nucleotide RNAs mediate RNA interference in cultured Mammalian cells. *Nature*, 411(6836), 494–498.
8. Blanchette, M., Kunisawa, T., & Sankoff, D. (1999). Gene order breakpoint evidence in animal mitochondrial phylogeny. *Journal of Molecular Evolution*, 49(2), 193–203.
9. Brocchieri, L., & Karlin, S. (2005). Protein length in eukaryotic and prokaryotic proteomes. *Nucleic Acids Research*, 33(10), 3390–3400.
10. Zastrow, M. S., Flaherty, D. B., Benian, G. M., & Wilson, K. L. (2006). Nuclear titin interacts with A- and B-type lamins in vitro and in vivo. *Journal of Cell Science*, 119(2), 239–249.
11. Luce, G., & Myoupo, J. F. (1998). Systolic-based parallel architecture for the longest common subsequences problem. *VLSI Journal Integration*, 25, 53–70.
12. Sankoff, D., & Blanchette, M. (1999). Phylogenetic invariants for genome rearrangements. *Journal of Computational Biology*, 6, 431–445.
13. Sheridan, R. P., & Venkataraghavan, R. (1992). A systematic search for protein signature sequences. *Proteins*, 14(1), 16–18.
14. Hirschberg, D. S. (1977). Algorithms for the longest common subsequence problem. *Journal of the ACM*, 24, 664–675.
15. Masek, W. J., & Paterson, M. S. (1980). A faster algorithm computing string edit distances. *Journal of Computer and System Sciences*, 20, 18–31.
16. Rick, C. (1994, October). *New algorithms for the longest common subsequence problem* (Technical Report No. 85123-CS). Computer Science Department, University of Bonn.
17. Smith, T. F., & Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, 147, 195–197.
18. Hakata, K., & Imai, H. (1992). Algorithms for the longest common subsequence problem. In *Proceedings of Genome Informatics Workshop III* (pp. 53–56).
19. Hakata, K., & Imai, H. (1998). Algorithms for the longest common subsequence problem for multiple strings based on geometric maxima. *Optimization Methods and Software*, 10, 233–260.
20. Chen, Y., Wan, A., & Liu, W. (2006). A fast parallel algorithm for finding the longest common sequence of multiple biosequences. *BMC Bioinformatics*, 7, S4.
21. Korkin, D. (2001). *A new dominant point-based parallel algorithm for multiple longest common subsequence problem* (Technical Report TR01-148). University of New Brunswick.
22. Xu, X., Chen, L., Pan, Y., He, P. (2005). Fast parallel algorithms for the longest common subsequence problem using an optical bus. In *Lecture Notes in Computer Science* (pp. 338–348). Springer.
23. Bork, P., & Koonin, E. V. (1996). Protein sequence motifs. *Current Opinion in Structural Biology*, 6, 366–376.
24. Korkin, D., & Goldfarb, L. (2002). Multiple genome rearrangement: A general approach via the evolutionary genome graph. *Bioinformatics*, 18, S303–S311.

25. Korkin, D., Wang, Q., & Shang, Y. (2008). An efficient parallel algorithm for the multiple longest common subsequence (MLCS) problem. In *Proceedings of the 37th International Conference on Parallel Processing (ICPP'08)* (pp. 354–363).
26. Bergroth, L., Hakonen, H., & Raita, T. (2000). A survey of longest common subsequence algorithms. In *Proceedings of International Symposium. String Processing Information Retrieval (SPIRE'00)* (pp. 39–48).
27. Chin, F. Y., & Poon, C. K. (1990). A fast algorithm for computing longest common subsequences of small alphabet size. *Journal of Information Processing*, 13(4), 463–469.
28. Wang, Q., Korkin, D., & Shang, Y. (2011, March). A fast multiple longest common subsequence (MLCS) algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 23(3).
29. Yang, J., Yun, X., Sun, G., & Shang, Y. (2013). A new progressive algorithm for a multiple longest common subsequences problem and its efficient parallelization. *IEEE Transactions on Parallel and Distributed Systems*, 24(5), 862–870.
30. Hirschberg, D. S. (1975, June). A linear space algorithm for computing maximal common subsequences. *Communications of the ACM*, 18, 341–343.
31. Irving, R. W., & Fraser, C. (1992). Two algorithms for the longest common subsequence of three (or more) strings. In *Proceedings of the Third Annual Symposium on Combinatorial Pattern Matching* (pp. 214–229). London, UK: Springer.
32. Wang, Q., Korkin, D., & Shang, Y. (2009). Efficient dominant point algorithms for the multiple longest common subsequence (MLCS) problem. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence* (pp. 1494–1499). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
33. Chen, Y., Wan, A., & Liu, W. (2006). A fast parallel algorithm for finding the longest common sequence of multiple biosequence. *BMC Bioinformatics*, 7, 4.
34. Wang, Q., Korkin, D., & Shang, Y. (2011). A fast multiple longest common subsequence (MLCS) algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 23(3), 321–334.
35. Jiang, T., & Li, M. (1994). On the approximation of shortest common supersequences and longest common subsequences. In *Proceedings of the 21st International Colloquium on Automata, Languages and Programming* (pp. 191–202). London, UK: Springer.
36. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms* (3rd ed.). Cambridge, MA, USA: MIT Press.
37. Jones, E., Oliphant, T., & Peterson, P. (2013). SciPy: open source scientific tools for python. Retrieved from <http://www.scipy.org/>. Accessed April 19, 2013.
38. Maier, D. (1978). The complexity of some problems on subsequences and supersequences. *Journal of the ACM*, 25, 322–336.
39. Julstrom, B. A., & Hinkemeyer, B. (2006). Starting from scratch: Growing longest common subsequences with evolution. In *Proceedings of the 9th International Conference on Parallel Problem Solving from Nature* (pp. 930–938). Berlin, Heidelberg: Springer.
40. Bergroth, L., Hakonen, H., & Raita, T. (2000). A survey of longest common subsequence algorithms. In *Proceedings Seventh International Symposium on String Processing and Information Retrieval SPIRE 2000* (pp. 39–48).
41. Attwood, T. K., & Findlay, J. B. C. (1994). Fingerprinting G protein coupled receptors. *Protein Engineering*, 7(2), 195–203.
42. Bourque, G., & Pevzner, P. A. (2002). Genome-scale evolution: Reconstructing gene orders in the ancestral species. *Genome Research*, 12, 26–36.

Fast and Accurate Fingerprint Recognition in Principal Component Subspace



S. P. Ragendhu and Tony Thomas

Abstract In the case of fingerprint-based person recognition, the most widely used discriminating features are minutiae (end points and bifurcations of ridges). Majority of fingerprint matching algorithms are dealing with comparing the parameters directly derived from or relative to minutiae points extracted from the templates. Hence eventually fingerprint matching based on minutiae can be reduced to a 2D point set matching problem. Various security pitfalls like impersonation using one's minutiae coordinates and performance issues related to enhancement as well as spurious minutiae removal are obvious in such a system. Certain non-minutiae based schemes are able to give acceptable performance at the cost of increased complexity which results in increased execution time. In order to overcome these issues, we propose a simple yet efficient and faster fingerprint alignment and matching scheme based on statistical features which will not reveal the unique local features of the template. Proposed matching technique is based on the weighted similarity score obtained by comparing the principal component subspaces of fingerprint templates. Proposed method also utilizes an alignment scheme based on principal components calculated for the 2D coordinates of fingerprint region with minimal overhead without any helper data.

1 Introduction

Fingerprint is the pattern formed by ridges and valleys at the palmar side of the finger tip, which is found to be unique for each individual. Fingerprint biometrics deal with extracting the distinguishing features of this pattern for each individual and identification and/or authentication by comparing the extracted features [28].

S. P. Ragendhu (✉) · T. Thomas
Indian Institute of Information Technology and Management-Kerala,
Research Centre of Cochin University of Science and Technology, Kazhakkuttom,
Thiruvananthapuram, Kerala, India
e-mail: ragendhu.res15@iiitmk.ac.in

T. Thomas
e-mail: tony.thomas@iiitmk.ac.in

A recent classification of fingerprint features into a three-level hierarchy has been proposed by experts [6, 29]. Even though there is difference of opinion among researchers regarding this classification, as a whole we can conclude that Level 1 features include global features like orientation field, ridge frequency field or singular regions (cores and deltas). Level 2 features include less global features like minutiae or ridge skeleton. More quantitative features like sweat pores, ridge contours etc. can be considered as Level 3 features. Level 1 feature can only be used for initial classification of fingerprints into different groups such as whorls, loops, or arches; whereas Level 2 features like minutiae can be used for more refined classification and person authentication as they represent local discriminating features even within the fingerprints with similar global structure. Different security issues associated with unauthorized reconstruction of Level 2 features from Level 1 features and vice versa while employing minutiae-based schemes are available in the literature [16, 22]. Nevertheless, non-minutiae-based features are also not exempted from the threat of unauthorized reconstruction of original templates [23].

In both minutiae-based and non-minutiae-based schemes, securing the features against the reconstruction of templates is inevitable. However, when existing template protection mechanisms such as cancellable biometrics and biometric cryptosystems are suffering from degradation of matching performance along with increased complexity when combined with state of the art fingerprint recognition systems [19]. Increased complexity slows down the recognition system. A simple, efficient, and secure feature representation for fingerprints, which can be secured easily is need of the hour.

Person authentication (one to one) and person identification (one to many) are the two major categories of biometric applications. As the query template will be compared with only one registered template in the case of one to one authentication, most of the algorithms perform efficiently in spite of complex calculations. However, for person identification, query template needs to be compared with all the templates present in selected template database. If complex calculations are involved, the execution time becomes the key factor in deciding the performance of matching technique. Hence person identification systems require comparatively simpler and faster method to match fingerprint templates.

In this paper, we propose principal components based alignment and matching for fingerprint templates, which reduces the execution time significantly without compromising the performance. Since we use only global statistical features (Level 1) for authentication, there is no need to store the intrinsic details (Level 2 and Level 3) of the biometric data in the system and it will ensure the security and privacy of the overall system. In the proposed alignment scheme, orientation estimation is done by calculating the principal components of the 2D coordinate points of fingerprint region and rotation by estimated angle is performed. Proposed matching technique compares fingerprints by analyzing the similarity of principal component subspaces of corresponding templates. Similarity of fingerprint images are calculated by estimating weighted cosine similarity of corresponding principal component subspaces. The calculations are quite simple and straight forward; which makes the method simpler and faster.

Rest of this paper is organized as follows. Section 2 discusses about some related literature in the area of fingerprint recognition. Introduction to the concept of principal components is given in Sect. 3. Principal components based orientation estimation and alignment of fingerprint is detailed in Sect. 4.1. Fingerprint matching scheme based on similarity and weighted similarity calculation of fingerprint templates using principal components are explained in Sect. 4.2. Section 5 describes the experimental results and performance of proposed matching scheme and Sect. 6 briefs the conclusion drawn and future directions for research.

2 Related Works

Vast varieties of approaches have been proposed in the domain of fingerprint recognition, in both minutiae-based [28] and non-minutiae-based [26] categories. Overhead associated with preprocessing techniques (spurious minutiae removal, minutiae verification, etc.) itself is really high in the case of minutiae-based algorithms; whereas complexity of calculations associated with non-minutiae-based schemes with acceptable performance are higher compared to minutiae-based systems.

Minutiae are coordinate dependent and will get affected largely by distortion and rotation. Most of the works in the literature are dealing with either relative distance or relative coordinates of minutiae points and the match is determined by the overlap between selected features of registered and query templates [24]. The most widely accepted and recent minutiae-based recognition system is based on Minutiae Cylinder Code (MCC) representation of fingerprint templates [4]. In MCC representation, a cylinder structure is used which encodes spatial and directional relationships between the minutiae and its neighborhood. In spite of high matching accuracy, higher execution time and higher memory requirement are the major disadvantages of MCC-based approach.

One of the popular non-minutiae-based approaches used in fingerprint recognition is Gabor features based fingercode (filterbank-based) representation of fingerprint [10]. Matching accuracy of this technique is found to be high, though at the cost of increased complexity. We have compared the performance of proposed work with filterbank-based approach and MCC-based approach, where the former is non-minutiae-based approach and latter is minutiae-based approach.

There are vast varieties of applications for principal components in different domains [13]. The most important application of principal components is principal component analysis (PCA); which is a statistical technique widely used mainly for linear dimensionality reduction. When we consider the domain of biometric recognition, conventional PCA is commonly used in face recognition. However, conventional PCA (similar to eigen fingerprints and weight calculation) is not able to give acceptable performance in the case of fingerprint recognition [27]. When compared to face images, the richness and variance of features are less in fingerprint templates as we only have some structures (singular points, endpoints and bifurcations) formed by the flow of ridges and valleys in fingerprints. The sensitiveness of principal components

to the outliers might be another reason for poor performance of PCA in fingerprints. There are certain works which use principal components for structural classification of fingerprints [2]. Principal components have been used for minutiae verification and spurious minutiae removal in some works [9].

When we use conventional PCA in biometric recognition [3], if there are m registered images ($N \times N$), the first step is to find mean image ($N \times N$) of the registered images and it will be subtracted from each of the registered image. After this, a matrix $A_{N^2 \times m}$ will be formed; where each column vector of A corresponds to each mean subtracted image. Each column of $A.A^T$ represents principal component of A . Each of the registered templates will be reconstructed using the obtained principal components. While reconstruction, weights associated with different principal components for each image will be calculated. This weight vectors are considered as the features. When query images are coming, weights of those images will also be calculated corresponding to the obtained principal components. Distance between weight vectors of query image and each of the registered images will be calculated. If there is a registered image with distance less than particular threshold, that will be considered as the matching template.

In the proposed scheme, instead of using conventional PCA technique, principal components are used directly for estimating similarity of the fingerprints. We are not performing dimensionality reduction or weight calculation for registered images. Instead, principal component subspace of each of the registered template will be compared with the principal component subspace of the query image. This technique has been used commonly in pattern recognition of historical data as well as in process control [5, 11, 25].

Matching techniques which deal with similarity measure based on less complex calculations can provide optimal tradeoff between performance and complexity. Almost all the similarity measures proposed in literature for fingerprint templates are based on minutiae points [7]. Proposed work discusses a different application of principal components, where the similarity between the biometric images is calculated based on the cosine similarity of principal components.

3 Principal Components

Principal Component Analysis (PCA) deals with finding out new variables (Principal components) that are linear functions of those in the original dataset, which successively maximize variance and that are uncorrelated with each other [13]. Principal components can be interpreted as the new uncorrelated variables extracted from a high dimensional dataset. The first principal component gives the axis of maximum variance and the second principal component represents the axis of second most variance. The problem of extracting the orthonormal vectors (principal components) can be reduced to an eigenvalue/eigenvector problem.

The basic idea behind principal components can be given by the Eq. 3:

$$x \approx \sum_{i=1}^m w_i a_i + c, \quad (3)$$

where, a_i , $i = 1, 2, \dots, m$ represent orthonormal vectors (principal components), w_i represent weights associated with each principal component and c is the minimal squared reconstruction error while reconstructing the data x . If the dimensionality of the data x is n , Eq. 3 shows how principal components can be used to reconstruct x in a dimensionally reduced space ($m < n$). Hence the most important application of Principal Component Analysis (PCA) is linear dimensionality reduction.

Different variants of applications have been proposed for principal component analysis in different disciplines. One of the applications of principal components in image processing is the estimation of orientation of objects in the images [18]. We have utilized this property of principal components for aligning the fingerprint images and detailed steps are given in Sect. 4.

Another application of principal components which has got recent attention from researchers is pattern recognition in two multivariate datasets based on the similarity of principal components of those two datasets. This technique is derived from the conventional factor analysis method [12] and it was first proposed by Krzanowski [14]. Multivariate data analysis is a well-researched area and many statistical techniques have been proposed for pattern recognition in multivariate datasets. Different fields such as finance, process control, multimedia and image processing are dealing with multivariate data. In the case of biometric systems, the biometric data collected for authentication in almost all the commonly used modalities (fingerprint, face, iris, finger vein, palmprint, palm vein, etc.) are in the form of images, which is a multivariate data. Hence we propose principal components based similarity measure commonly used for the pattern recognition in multivariate datasets for comparing the biometric images. When data is represented in high dimensions, it is difficult to analyze the similarity between them. Principal components basically represent optimized coordinate system to represent the given data. Hence it becomes comparatively easier to compare and analyze the data which is represented in terms of principal components.

3.1 Calculation of Principal Components

Principal components of a data matrix (biometric image) can be calculated either by singular value decomposition of a column centered data matrix or by the eigen value decomposition of a covariance matrix. We have used the latter approach for calculating the principal components. Different steps involved in the calculation of principal components are given in Algorithm 1.

Algorithm 1 CalcPC

Step 1: Normalize the input image $I_{(r \times c)}$, by subtracting the mean image of size $(r \times c)$ and obtain $\tilde{I}_{(r \times c)}$.

Step 2: Calculate the covariance matrix of $\tilde{I}_{(r \times c)}$ using either Equation 1 or Equation 2 (depends on the need of dimensionality reduction) as given below:

$$CovIm_{(r \times r)} = (\tilde{I})_{(r \times c)} \times (\tilde{I})_{(c \times r)}^T \quad (1)$$

$$CovIm_{(c \times c)} = (\tilde{I})_{(c \times r)}^T \times (\tilde{I})_{(r \times c)} \quad (2)$$

Step 3: Find out the eigen vectors ($e_i, i = 1, 2, \dots, c$ or r) corresponding to the eigen values ($\lambda_i, i = 1, 2, \dots, c$ or r) (sorted in descending order) of covariance matrix. Eigen vectors (e_i) represent the principal components and the amount of variance is given by corresponding eigen values (λ_i).

4 Proposed System

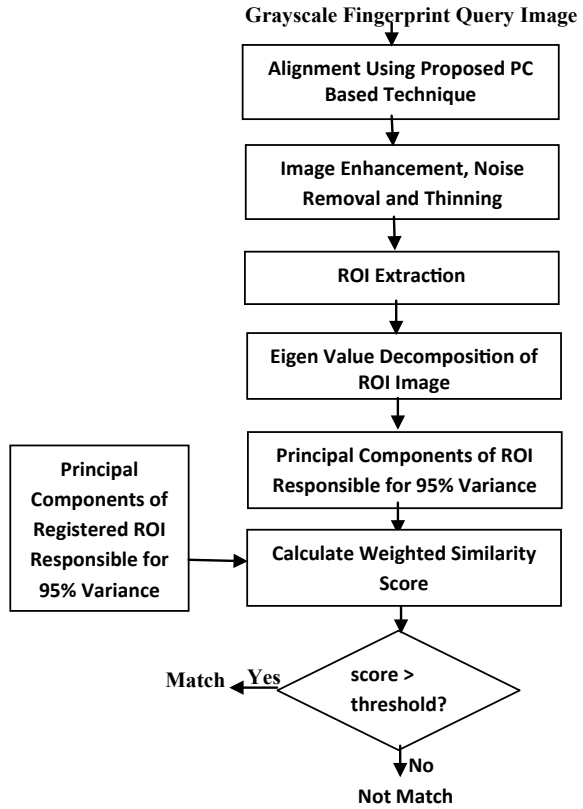
Different steps involved in proposed matching technique are shown in Fig. 1. Input grayscale fingerprint image will be aligned according to the alignment method explained in Sect. 4. Angular and orientation alignment will be achieved for the templates after this step. The enhancement technique proposed by Jain et al. [8] is applied on the aligned images and morphological operations are applied for smoothing the image. Noise removal and thinning are performed and region of interest (ROI) is extracted out based on the core point [1]. The results of the different preprocessing steps (enhancement, thinning and ROI extraction) are shown in Fig. 2. Principal components of the extracted ROI image are calculated by eigen value decomposition method as explained in Sect. 3.1.

4.1 Principal Components Based Fingerprint Alignment

Fingerprint alignment is the first step to be performed in the proposed system. There are various complicated procedures proposed in the literature for aligning the query template with the registered template [21]. Some of them are based on the core point which require executing different algorithms for core point detection [17]. Some others are based on the helper data stored along with the registered templates [20], which can become a security threat to the system. In both of these approaches, significant amount of processing time is required in the alignment process. We use a comparatively fast, secure and simple method based on the properties of principal components for aligning fingerprint templates.

In the proposed approach, orientation of fingerprint is estimated by finding out the axis of maximum variance of the fingerprint region. Fingerprint region is identified from the image and coordinates of fingerprint region are extracted. Hence the

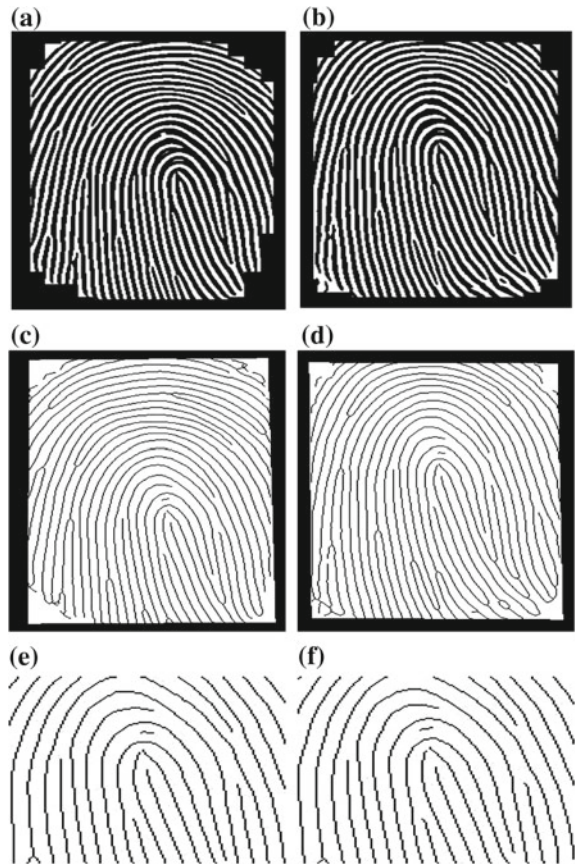
Fig. 1 Steps in proposed matching technique



entire fingerprint region, I is reduced to a set of 2D coordinate points, which can be represented as $I = \{(x_i, y_i) : I(x_i, y_i) < 230, i = 1, 2, \dots, n\}$ (where the value of n depends upon the fingerprint template). Principal components of this 2D data are calculated; wherein the first principal component($PC1$) will give the axis of maximum variance and the second principal component($PC2$) will give the axis of second most variance.

The orientation of fingerprint image w.r.t. current axis is calculated by measuring the angle between $PC1$ and vertical axis. Different steps in the alignment process is explained in Algorithm 2. Results of the alignment process are shown in Fig. 3. The red lines in the input images shown in Fig. 3a, b represent horizontal and vertical axes. The reference point calculated in Step 4 of the Algorithm 2 is used for visualization of principal components. The blue lines in the figures represent the principal components. The longest blue line represents $PC1$ which gives the axis of maximum variance. The second blue line orthogonal to $PC1$ is $PC2$, which represents the axis of second most variance. The angle estimated in Step 5 of the Algorithm 2 is the angle between $PC1$ and the vertical axis. We will perform a clockwise rotation if x -coordinate of $PC1$ is positive and we will perform counter clockwise rotation if

Fig. 2 Result of preprocessing steps in intra class images. **a** and **b** are the aligned images after enhancement. **c** and **d** are the corresponding thinned images. **e** and **f** are the images after ROI extraction



x -coordinate of $PC1$ is negative. The alignment is only based on the angle of rotation and no reference point is used for alignment. Results show that if the images have acceptable amount of information (if it is not latent), proposed alignment process can improve the performance of the system.

4.2 Fingerprint Matching

Proposed matching technique considers a fingerprint image as a multivariate data and hence quantifies the similarity of fingerprint templates based on the similarity of principal component subspaces. When data is represented in high dimensions, it is difficult to analyze the similarity between them. Principal components basically represent optimized coordinate system to represent the given data. Hence it becomes comparatively easier to compare and analyze the data which is represented

Algorithm 2 FPAlign

-
- Step 1: Input $I = \{(x_i, y_i) : I(x_i, y_i) < 230, i = 1, 2, \dots, n\}$
- Step 2: Form $n \times 2$ matrix $XY, XY = \begin{bmatrix} x_1 & y_1 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ x_n & y_n \end{bmatrix}$
- Step 3: Find the covariance matrix of XY . Calculate the principal components corresponding to the two eigen vectors of the covariance matrix using eigen value decomposition method as discussed in Section 3.1. Let $PC1 = \begin{bmatrix} \alpha_{x1} \\ \alpha_{y1} \end{bmatrix}$ and $PC2 = \begin{bmatrix} \alpha_{x2} \\ \alpha_{y2} \end{bmatrix}$ be the principal components corresponding to the eigen values λ_1 and $\lambda_2, \lambda_1 > \lambda_2$.
- Step 4: Calculate the mean values $\tilde{x} = \frac{\sum_{i=1}^n x_i}{n}, \tilde{y} = \frac{\sum_{i=1}^n y_i}{n}$ and set the reference point as (\tilde{x}, \tilde{y}) .
- Step 5: Calculate the angle(θ) between unit vector along Y axis (y) and $PC1$.
 $\theta = \cos^{-1} \frac{y \cdot PC1}{|y| \cdot |PC1|}$
- Step 6: In order to rotate the points XY , we need to first translate it to origin by subtracting the means. $\tilde{x}_i = x_i - \tilde{x}, \tilde{y}_i = y_i - \tilde{y}$.
- Step 7: Form the matrix of translated points $\tilde{X}\tilde{Y}, \tilde{X}\tilde{Y} = \begin{bmatrix} \tilde{x}_1 & \tilde{y}_1 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \tilde{x}_n & \tilde{y}_n \end{bmatrix}$
- Step 8: Rotate the matrix $\tilde{X}\tilde{Y}$ by the estimated angle $\theta, X'Y' = \begin{bmatrix} \tilde{x}_1 & \tilde{y}_1 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \tilde{x}_n & \tilde{y}_n \end{bmatrix} \times \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$
- Step 9: Translate the points back to original position, $x_i'' = x_i' + \tilde{x}, y_i'' = y_i' + \tilde{y}$.
-

in terms of principal components. Principal components obtained corresponding to each dimension will be compared with principal components of the query image. Two similarity measures are experimented for estimating the similarity between principal components of fingerprint templates.

4.2.1 Similarity Measure Based on Principal Components

If X_1 and X_2 represent two fingerprint ROI images of size $p \times q$, PCA can model X_1 and X_2 using k principal components each ($k \ll q$). If L and M represent corresponding subspaces of size $(p \times k)$ of each image, the similarity of these subspaces can be used as a measure of similarity between the images. The subspaces L and M also represent the eigen vector matrices corresponding to the first k eigen values, which can be obtained by the eigen value decomposition of covariance matrices of X_1 and X_2 . Hence PCA similarity factor S_{PCA} , compares these subspaces and is defined by the Eq. 4 [14]:

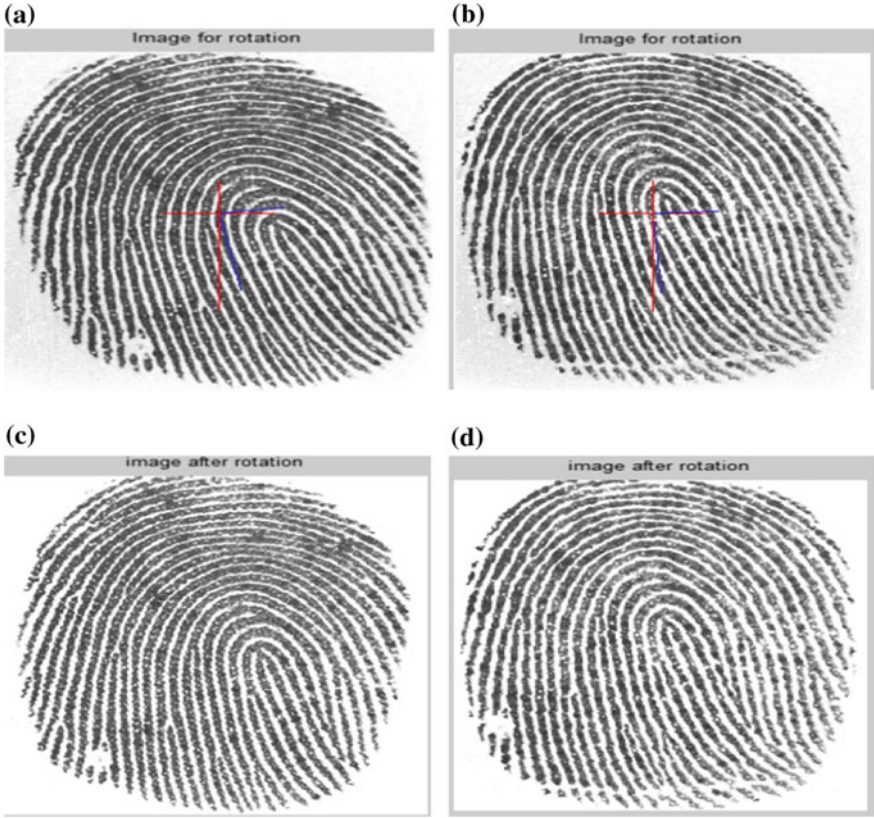


Fig. 3 Intra class fingerprint images before and after alignment. **a** and **b** are the input images. **c** and **d** are the corresponding aligned images

$$S_{PCA} = \frac{\text{trace}(L^T M M^T L)}{k}, \tag{4}$$

Basically we are checking only the similarity of orientation of principal components in the case of PCA similarity factor (S_{PCA}). Principal components responsible for providing at least 95% variance in each image are selected and similarity calculation is done in these subspaces formed by k principal components. Equation 4 can be represented as the squared cosine values of the angle between principal components of templates and is given in Eq. 5 [14]:

$$S_{PCA} = \frac{1}{k} \sum_{i=1}^k \sum_{j=1}^k \cos^2 \theta_{ij}, \tag{5}$$

where k is the smallest number of principal components responsible for at least 95% variance in each input biometric images and θ_{ij} is the angle between i th principal

component of first image and j th principal component of second image. The value of S_{PCA} will range between 0 and 1, where 1 corresponds to maximum similarity.

Performance of S_{PCA} was not desirable for fingerprint templates because of the high FAR value. The results of S_{PCA} based experiments were analyzed and it was found that templates having similar structure with comparatively similar orientations are the reason for increase in FAR . Hence we concluded that S_{PCA} is more suitable for structural classification of fingerprint images than person authentication. The reason for poor performance of S_{PCA} is that only orientation correspondences were involved in the calculation of match score.

4.2.2 Weighted Similarity Measure Based on Principal Components

Modified weighted PCA similarity(S_{PCA}^λ) measure considers the contribution or variance associated with each principal component along with orientation correspondence while calculating the similarity score [15]. Calculation of the modified similarity score is given by the Eq. 6:

$$S_{PCA}^\lambda = \frac{\sum_{i=1}^k \sum_{j=1}^k \lambda_i^1 \lambda_j^2 \cos^2 \theta_{ij}}{\sum_{i=1}^k \sum_{j=1}^k \lambda_i^1 \lambda_j^2}, \quad (6)$$

where k is the number of principal components responsible for at least 95% variance in each input biometric images, θ_{ij} is the angle between i th principal component of first image and j th principal component of second image and λ_i^1 represents the eigen value corresponding to i th principal component of first image and λ_j^2 represents the eigen value corresponding to j th principal component of second image. In Eq. 6 we have represented the similarity w.r.t. all possible combinations of eigen values of two images. Significant reduction of FAR to 6% is obtained against FRR of 11%.

5 Results and Discussions

Experiments are conducted using both $FVC2000$ and $FVC2004$ databases, wherein 8 images per person were present. We started experiments with fingerprint templates of 40 different people available in $Db2a$ of $FVC2000$ database ($40 \times 8 = 320$). After analyzing the results obtained from $FVC2000$ database, we extended our experiments to $FVC2004$ also. Fingerprint images of 60 different people available in $FVC2004$ fingerprint database (40 from $DB2A$ and 20 from $DB2B$) are also used in experiments ($60 \times 8 = 480$). Four images per person are used as training images and remaining four images are used for testing. Cross validation is performed using modified weighted similarity scores of all images. Figure 4 shows the values of FAR against GAR at different thresholds obtained for $FVC2000$ and $FVC2004$. Accuracy

Fig. 4 Calculation of area under curve (AUC)

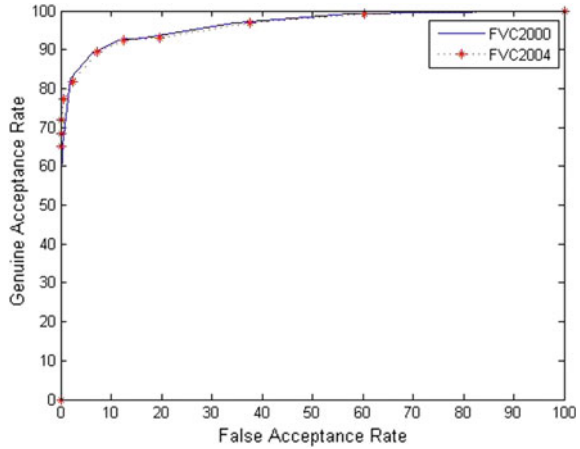


Table 1 Accuracy calculated from Fig. 4

Database	Accuracy (AUC) (%)
FVC2000	96.6
FVC2004	96.33

Table 2 Comparison of results

Method	Accuracy at 2%FAR (%)
Jain et al. [10]	96.78
Cappelli et al. [4]	97.25
Proposed	97.31

is computed through Area under curve(AUC) method from Fig. 4. It is observed that overall accuracy of proposed method is almost equal for the two databases, which is given in Table 1.

Performance of the proposed system is compared with two major approaches in fingerprint recognition. First one is filterbank-based system proposed by Jain et al. [10] and second one is Minutiae Cylinder Code(MCC) method proposed by Cappelli et al. [4]. Both the approaches are tested against the same set of FVC2004 images. The accuracy of the methods are calculated at different thresholds. Different FAR values are obtained corresponding to different thresholds. Number of genuine acceptance (True Positives) and genuine rejection (True Negatives) are calculated at each of the selected FAR levels. Accuracy is calculated using Eq. 7. Comparison of accuracy of methods at 2% FAR is shown in Table 2. We can observe that the proposed method is able to obtain better performance compared to filterbank-based system and slightly higher performance compared to MCC.

$$Accuracy = \frac{True\ Positives + True\ negatives}{Total\ number\ Templates\ Tested} \tag{7}$$

Table 3 Average matching time (ms) in one to one authentication

Method	Average matching time
Jain et al. [10]	18
Cappelli et al. [4]	15
Proposed	2

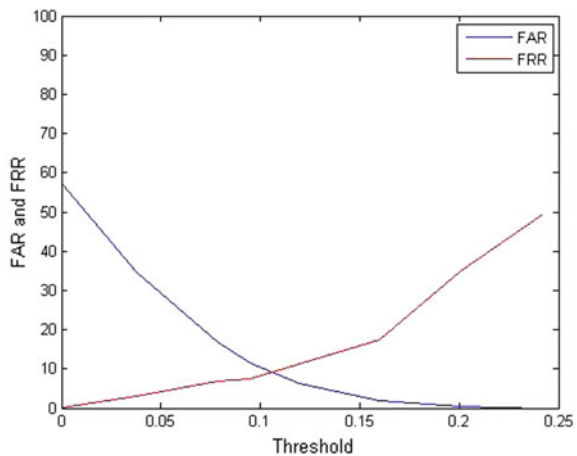
Table 4 Execution time (s) in one to many identification

Method	Execution time
Jain et al. [10]	6
Cappelli et al. [4]	4.5
Proposed	1.2

We also compared the time taken for the proposed method with the other two approaches. The two parameters we took for comparison are matching time and execution time. Matching time means the time required for calculating the similarity score and it does not include the time required for preprocessing techniques. It is evident from Table 3 that the proposed method is far better in terms of matching time. Execution time includes the time taken for entire steps in the approach, including the time required for preprocessing steps. We calculated the execution time of proposed method in case of one to many identification using a template database with 400 templates (4 images per person). Result is compared with the methods proposed by Jain et al. and Cappelli et al. and the result is given in Table 4.

The most appreciable feature of the proposed method is that the execution time for one to many identification is found to be very less compared to the other two methods in terms of execution time. One of the notable applications of biometric person identification is criminal identification by law enforcement agencies. In most of the cases, the template database will be large and hence the existing schemes will take pretty good amount of time to complete the searching. However, the performance

Fig. 5 Calculation of EER



of the proposed system show that it can reduce the search space, and converge to a small set of matching templates in almost no time.

Average value of *FAR* and *FRR* at different thresholds are calculated and Fig. 5 shows the average *FAR* and *FRR* values plotted against the normalized threshold range 0–0.25, and it is observed that *EER* of 10% is obtained at 0.10625.

6 Conclusion and Future Works

Fingerprint alignment and matching based on statistical properties of the templates is a promising direction of thought. Vulnerabilities of minutiae-based approach raise serious security and privacy concerns in fingerprint systems. Another serious drawback is the increased execution time due to highly complicated procedures. Results show that acceptable matching performance can be ensured even from statistical features based matching schemes without large computations. Analyzing the applicability and performance of the proposed technique in multi-biometric systems and designing suitable template protection mechanism for proposed system with minimal complexity are the major future directions for research.

Acknowledgements This research is supported by Kerala State Planning Board project CEPIA (2017–18).

References

1. Bahgat, G., Khalil, A., Kader, N. A., & Mashali, S. (2013). Fast and accurate algorithm for core point detection in fingerprint images. *Egyptian Informatics Journal*, 14(1), 15–25.
2. Ballan, M., & Gurgen, F. (1999). On the principal component based fingerprint classification using directional images. *Mathematical and Computational Applications*, 4(2), 91–97.
3. Bhattacharyya, S., & Chakraborty, S. (2014). Reconstruction of human faces from its eigen-faces. *International Journal*, 4(1).
4. Cappelli, R., Ferrara, M., & Maltoni, D. (2010). Minutia cylinder-code: A new representation and matching technique for fingerprint recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12), 2128–2141.
5. Damarla, S. K., & Kundu, P. (2011). Classification of unknown thermocouple types using similarity factor measurement. *Sensors & Transducers*, 124(1), 11.
6. Feng, J., & Jain, A. K. (2011). Fingerprint reconstruction: From minutiae to phase. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2), 209–223.
7. Ghany, K. K. A., Hassanien, A. E., & Schaefer, G. (2014). Similarity measures for fingerprint matching. In *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (ICIP)* (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).
8. Hong, L., Wan, Y., & Jain, A. (1998). Fingerprint image enhancement: Algorithm and performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8), 777–789.

9. Hsieh, C.-T., & Shyu, S.-R. (2007). Principal component analysis for minutiae verification on fingerprint image. In *Proceedings of the 7th WSEAS International Conference on Multimedia Systems & Signal Processing, Hangzhou, China*.
10. Jain, A. K., Prabhakar, S., Hong, L., & Pankanti, S. (2000). Filterbank-based fingerprint matching. *IEEE Transactions on Image Processing*, 9(5), 846–859.
11. Johannesmeyer, M. C., Singhal, A., & Seborg, D. E. (2002). Pattern matching in historical data. *AIChE Journal*, 48(9), 2022–2038.
12. Jolliffe, I. T. (1986). Principal component analysis and factor analysis. In *Principal component analysis* (pp. 115–128). Springer.
13. Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A*, 374(2065), 20150202.
14. Krzanowski, W. (1979). Between-groups comparison of principal components. *Journal of the American Statistical Association*, 74(367), 703–707.
15. Liao, T. W. (2005). Clustering of time series data survey. *Pattern Recognition*, 38(11), 1857–1874.
16. Liu, E., & Cao, K. (2016). Minutiae extraction from level 1 features of fingerprint. *IEEE Transactions on Information Forensics and Security*, 11(9), 1893–1902.
17. Msiza, I. S., Malumedzha, T. C., & Leke-Betechuoh, B. (2011). A novel fingerprint re-alignment solution that uses the tfcp as a reference. *International Journal of Machine Learning and Computing*, 1(3), 297.
18. Mudrova, M., Prochazka, A. (2005). Principal component analysis in image processing. In *Proceedings of the MATLAB Technical Computing Conference, Prague*.
19. Nandakumar, K., & Jain, A. K. (2015). Biometric template protection: Bridging the performance gap between theory and practice. *IEEE Signal Processing Magazine*, 32(5), 88–100.
20. Nandakumar, K., Jain, A. K., & Pankanti, S. (2007). Fingerprint-based fuzzy vault: Implementation and performance. *IEEE Transactions on Information Forensics and Security*, 2(4), 744–757.
21. Ramoser, H., Wachmann, B., & Bischof, H. (2002). Efficient alignment of fingerprint images. In *Proceedings. 16th International Conference on Pattern Recognition, 2002* (Vol. 3, pp. 748–751). IEEE.
22. Ross, A., Shah, J., & Jain, A. K. (2007). From template to image: Reconstructing fingerprints from minutiae points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4), 544–560.
23. Rozsa, A., Glock, A. E., & Boulton, T. E. (2015). Genetic algorithm attack on minutiae-based fingerprint authentication and protected template fingerprint systems. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 100–108). IEEE.
24. Shen, W., & Eshera, M. (2004). Feature extraction in fingerprint images. In *Automatic Fingerprint Recognition Systems* (pp. 145–181). Springer.
25. Wang, J., Zhu, Y., Li, S., Wan, D., & Zhang, P. (2014). Multivariate time series similarity searching. *The Scientific World Journal*.
26. Yang, J. (2011). *Non-minutiae based fingerprint descriptor*. InTech: In biometrics.
27. Yongxu, W., Xinyu, A., Yuanfeng, D., & Yongping, L. (2006). A fingerprint recognition algorithm based on principal component analysis. In *TENCON 2006. 2006 IEEE Region 10 Conference* (pp. 1–4). IEEE.
28. Zaeri, N. (2011). *Minutiae-based fingerprint extraction and recognition*. InTech: In biometrics.
29. Zhang, D., Liu, F., Zhao, Q., Lu, G., & Luo, N. (2011). Selecting a reference high resolution for fingerprint recognition using minutiae and pores. *IEEE Transactions on Instrumentation and Measurement*, 60(3), 863–871.

Smart Meter Analysis Using Big Data Techniques



Neha Pandey, Nivedita Das, Sandeep Agarwal, Kashyap Barua,
Manjusha Pandey and Siddharth Swarup Rautray

Abstract In the present-day, many government firms and global companies pay emphasis on energy conservation and efficient use of energy. The smart meter data have mapped a way to use energy efficiently. The need to use energy in an efficient way is very much required for developing countries like India. The emergence of smart meter gave us access to huge amounts of energy consumption data. It is an electronic component that records utilization of electric energy at regular intervals of time, be it hours, minutes, or seconds. This paper proposes a different method for grouping electricity consumption. Through smart meter, we get a huge amount of energy consumption data. These data are analyzed by various energy distribution companies which further leads to prediction of demand and consumption of user. Our paper uses a business intelligence tool such as map reduction to handle these data sets. Taking the advantage of this tool, energy distribution companies can reduce the investment by making the use of community hardware. Using distributed computing tools we can reduce the processing time appreciably to enable real-time monitoring and decision-making. Further, R is integrated to it to perform analysis. Various data sets are used to check the potential of the proposed models and approaches.

Keywords Smart meter data · Meter data analysis · Big data analytics · Hadoop · Map reduction · HDFS

N. Pandey (✉) · N. Das · S. Agarwal · K. Barua · M. Pandey · S. S. Rautray
School of Computer Engineering, KIIT University, Bhubaneswar, Orissa, India
e-mail: nehapandey47@yahoo.in

N. Das
e-mail: niveditads26@gmail.com

S. Agarwal
e-mail: sandygarg65@gmail.com

K. Barua
e-mail: Kashyapbarua@gmail.com

M. Pandey
e-mail: manjushafcs@kiit.ac.in

S. S. Rautray
e-mail: siddharthfcs@kiit.ac.in

© Springer Nature Singapore Pte Ltd. 2019

N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_27

1 Introduction

The basic energy demand is the consumption of electricity. The analysis is being made on energy consumption data to gain perception about the customer usage pattern. It is the job of smart meter to keep the record of data that has been generated for every minute. These data can be mined by the energy Utilities Company and new insights thus produced can lead to business benefit. The data from smart meter are raw stream and is also associated with high volume. For this, there is a requirement of framework which can accumulate and handle massive raw data stream generated by smart meter and to apply various logical and statistical techniques to correlate events with different conditions and finally predict the result. Well, the resolution to handle the above problem is Apache Hadoop. It is a tool which deals with distributed computing, which has large storage property along with computing capability. The framework of Apache Hadoop deals with the processing of large sets of data across a clutch of computers [1]. Parallel processing of data set is done by Hadoop and MapReduce. Unstructured and not originally intended for machine processing

- **Data:** A set of values of qualitative or quantitative variables is known as data. The idea of data is usually associated with scientific research. It is collected over by a massive range of institutions and corporation including governmental, non-governmental, and business firms. Data can be analyzed either by measuring and collecting the information or by visualizing the graphs and images.
- **RDBMS:** A database management system (DBMS) that is built upon the relational model is called a relational database management system (RDBMS). It has been one of the most popular choices for the storehouse of statistics. It is used for commercial and economical documentation, logical information, private data, and other applications for a long period of time [2]. Relational databases have often replaced other database techniques, because they are familiar in understanding and using. RDBMSs are used to store various kind of information.
- **BIG DATA:** Big data consists of data sets that are so massive and tangled that normal data processing applications are not sufficient. Managing data, storing, analyzing, finding, sharing, shifting, conceptualizing, querying, updating, information privacy, and data source [3]. Although big data is not specific to equate any amount of data, the term can be used for attending various bytes of data. Data might contain large variation in file types such as documents, sensors, images.
- **HADOOP:** The data generated by smart meter is sent by energy consumption data to the server at a regular interval of time leading to the generation of massive amount of data and existing tools were not functioning well in handling such massive data. It is an open architecture for the development of scattered application that can pass over very large amount of data [4]. It is a base which provides us with various cache and computing techniques. The advantages of Hadoop are: (1) It is an extraordinarily ascendible framework, as it accumulates and distributes extremely massive data sets through several of retailers that work in alignment. (2) Hadoop is a method of low-cost effective cache solution for businesses' exploding

- data. (3) Hadoop allows the businesses to easily reach to the fresh data sources and wander into various types of data to create attributes from that data (4).
- **Map Reduce:** It is a kind of computing architecture which is also a similar execution for handling and promoting massive data values along with collateral, distributed algorithm on a clutch [5, 6]. This framework consists of a Map() method to performs sieving and categorizing and a Reduce() method to performs a wrapping up operation. The “Map Reduce System” (also called “infrastructure” or “framework”) organizes the processing through collecting the distributed retainers, coordinating number of tasks in parallel, handling different communique and data transmission between the various parts of the framework, and providing for redundancy and fault tolerance [7].
 - **HDFS:** The Hadoop Distributed File System (HDFS) is a scattered file system modeled to run on commodity hardware [8, 9]. It is similar to extant distributed file systems. Still, the variation from other is consequential. It is very much fault-tolerant and is modeled to be redistributing on less-expensive hardware. It provides fast rate access to data and is relevant to large data sets applications [10]. It was initially designed as groundwork for the Apache engine project.
 - **ENERGY:** Energy basically deals with power derivation. It deals with derivation of power from chemical and physical resources, mainly to provide light and heat and to work on machines [11, 12]. It is a kind of conserved quantity. The basic fact about energy is that, “Energy can be converted in form, but it cannot be created or destroyed”.

1.1 Smart Meter Data Analysis

Electricity generation is dependent on unrenewable resources [13, 14]. Though India is wealthy in these resources, they are extinct at a very speed which means they will be almost gone soon whereas the renewable resources are not used to their best ability. Keeping these points in consideration the government has taken many measures towards the improvement of the electricity system [15]. The concept presented in this paper is an optimal procedure which if carried away properly will lead to an efficient and well-mannered electricity system in the future (Fig. 1).

1.1.1 Application of Big Data

Big data has a very vast application scenario be it in manufacturing, health care, education, media, government or it industries. It has increased the demand for management of the data sets [14]. In the government sector, the usage of big data provides efficiency in terms of cost, productivity, and usage [16]. Whereas in health care id helps in keeping record of various rage of data related to disease, patient, range of infection. Basically in every aspect of today’s world big data has been applied. One

Fig. 1 Application of big data

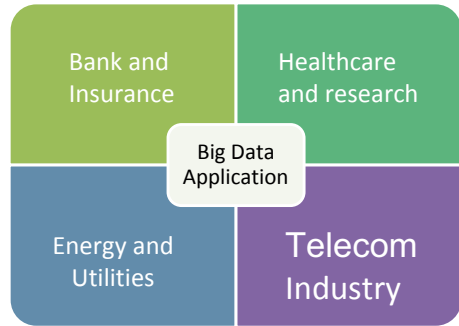
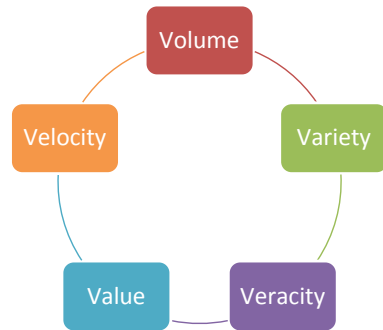


Fig. 2 Characteristics of big data



of its major applications is in energy consumption on which this paper has been worked upon (Fig. 2).

1.1.2 Components of Big Data

The basic five components of big data are: **Volume**; It basically deals with the amount of data generated. **Variety**; What kind of data are we dealing with and what is its nature comes under variety. **Velocity**; The speed of the generation of data basically comes under this category. **Variability**; [17–19] This deals with the consistency of data. **Varacity**; This deals with the quality of data (Fig. 3).

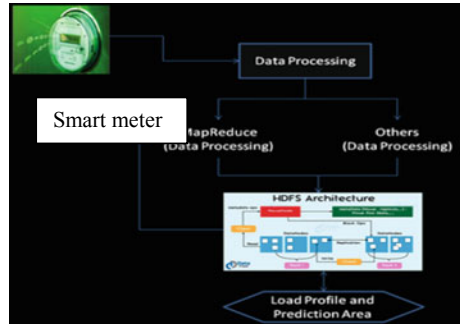
2 Related Work

See Table 1.

Table 1 Previous work done in smart meter analysis

Sl. No.	Year	Author	Paper title	Paper description	Tools and techniques
1	2015	Balaji K Bodkhe et al.	Analysis of smart meter data using hadoop	For efficient energy consumption, apache hadoop is used, which uses the data generated by smart meter for predicting energy consumption	<ol style="list-style-type: none"> 1. Smart meter 2. Apache hadoop
2	2014	Javier Conejero et al.	Analysis of hadoop power consumption	In this paper investigation and measurement is done on energy consumption [20]	<ol style="list-style-type: none"> 1. Media analysis 2. Instrumentation and monitoring
3	2017	Dr. Mohammed Abdul Waheed et al.	Analyzing the behavior of electricity consumption using hadoop	This paper proposes a novel approach for clustering of electricity consumption behavior dynamics	<ol style="list-style-type: none"> 1. SAX 2. Markov model 3. Distributed algorithm for large data sets
4	2017	Yimin Zhou et al.	A hierarchical system of energy consumption monitoring and information management system	In this paper an energy consumption monitoring and Information management system based on energy conservation model is developed [18]	<ol style="list-style-type: none"> 1. Energy consumption modeling 2. Information management

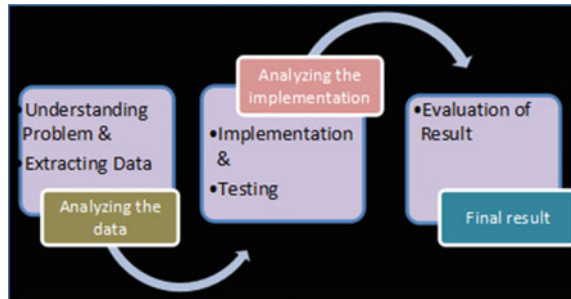
Fig. 3 Proposed framework



3 Proposed Methodology

The system that is implemented on smart meter data is providing a result for an electricity provider. Smart meter produces huge amount of data which is handled by Hadoop. The data has been processed through MapReduce and HDFS is used to store these data which includes various components like date, time, active, reactive, meter number, voltage, etc. MapReduce programming is done on these data sets that produces key/value pair. R language is used for this work. In this module, we have to create data set for electricity consumption which we usually get from smart meter data. It consists of various attributes such as customer details, billing details, payment details for the last few years. These data set has been loaded in HDFS from the local file system. We use HDFS to store massive amount of data which is later exported to R for load profile analysis. MapReduce is a preparing strategy and a program for appropriated processing. The MapReduce calculation contains two vital activities, in particular, Map and Reduce. In this module likewise utilized for dissecting the informal index using MapReduce (Fig. 4).

Fig. 4 Methodology



4 Implementation

4.1 Understanding the Problem and Extracting Data

This phase is quite essential in determining the success of the forecast. The main focus is made on extracting data and exploring those data from various industries and sub-industries. After analyzing the data simple forecast models are predicted. The data set is obtained from the site—<https://open-enernoc-data.s3.amazonaws.com/anon/index.html>. This data set consists of time series of various consumers and their corresponding metadata. Exploring and cleaning of data is done with help of a package named **data.table**. For visualization of relations **ggplot** package is used and for manipulation of date and time **lubricate** is used. Initially a frequency table for industry and sub-industry was made. We use the package **gmap** to map the location of our consumer on the map of use. After plotting the map we calculate the sq_m (square meter) of buildings for all the consumers. Further mean load for the sub-industry and median load for the industry was calculated (Figs. 5, 6 and 7; Table 2).

Fig. 5 Location of consumer



Fig. 6 Mean of sub-industry

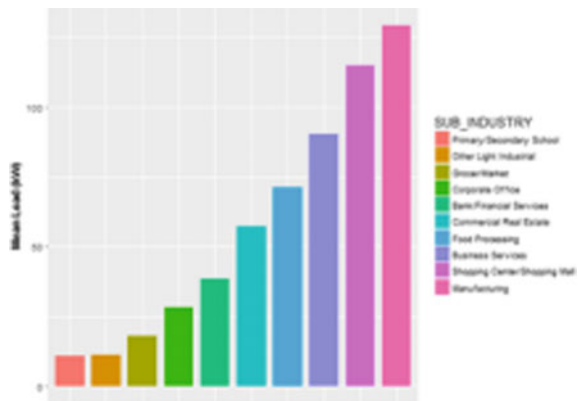


Fig. 7 Median of industry

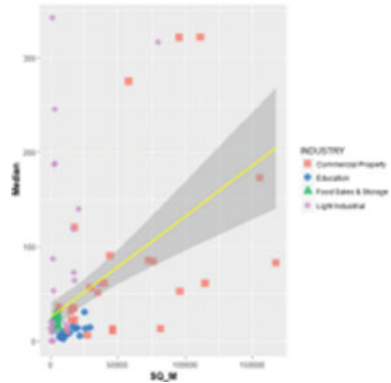


Table 2 Plot for industry and sub-industry

	Industry	Sub-industry	No.
1	Commercial property	Shopping center, shopping mal	14
2	Commercial property	Corporate office	2
3	Commercial property	Business services	3
4	Commercial property	Commercial real estate	4
5	Commercial property	Bank financial services	2
6	Education	Primary secondary school	25
7	Food sales and storage	Grocer market	25
8	Light industrial	Food processing	19
9	Light industrial	Manufacturing	5
10	Light industrial	Other light industrial	1

4.2 Implementation and Testing

In this phase data set is prepared to forecast and explore time series of load. Initially, we need to transform all the attributes of dates to classic date and time. The unwanted columns are being removed and we observe that the current structure of the present data set is full of time series. Then ids are being extracted (Figs. 8 and 9).

Now we need to fetch all the dates with all features taking the dimension, 288/days, now we will randomly check for any one id and its corresponding time series (Figs 10 and 11).

We observe that there is a strong dependency on time. So it would be better if the time series is aggregated at lower dimensions, around 48/day and once it is done then the plotting is done for four sub-industries providing them id’s respectively.

```
Console C:\Users\Sandeep Aganwal\Downloads\project\ jid
> value : num 106 105 102 103 102 ...
$ ID : chr "10" "10" "10" "10" ...
$ date_time: POSIXct, format: NA NA ...
- attr(*, "internal.selfref")=externalptr>
> count_ID <- DT[, .N, ID]
> full <- count_ID[N == max(N), .(ID)]
> DT <- DT[ID %in% full[, ID]]
> nrow(full) # number of extracted ids
[1] 43
> unique(DT[, ID])
 [1] "100" "101" "103" "110" "14" "144" "153" "189" "213" "218" "228" "236" "259"
[14] "275" "281" "285" "304" "31" "339" "341" "366" "384" "386" "391" "401" "455"
[27] "474" "475" "496" "512" "56" "65" "716" "718" "731" "761" "765" "766" "832"
[40] "88" "9" "92" "99"
>
```

Fig. 8 Screenshot of extraction of id

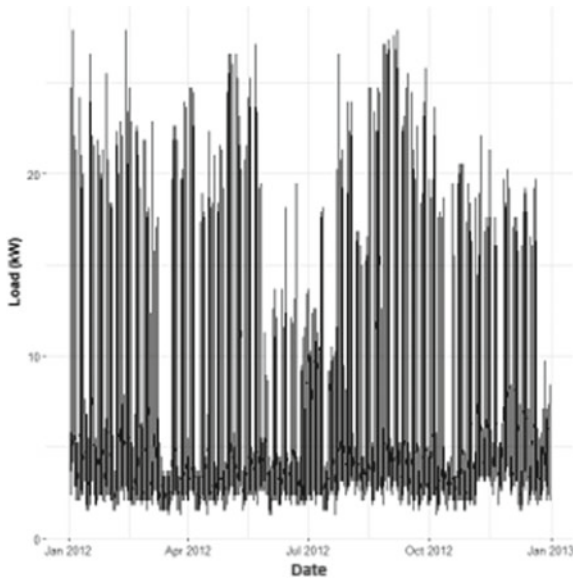


Fig. 9 Observation of consumption versus time series for 1 day

4.3 Evaluation of Result

During this phase, we consider all the data set from the previous implementation made and try to calculate the aggregate consumption by the user. For the forecast of electricity consumption, this aggregate value is used. Aggregate for all the customer is 45 and it has been plotted.

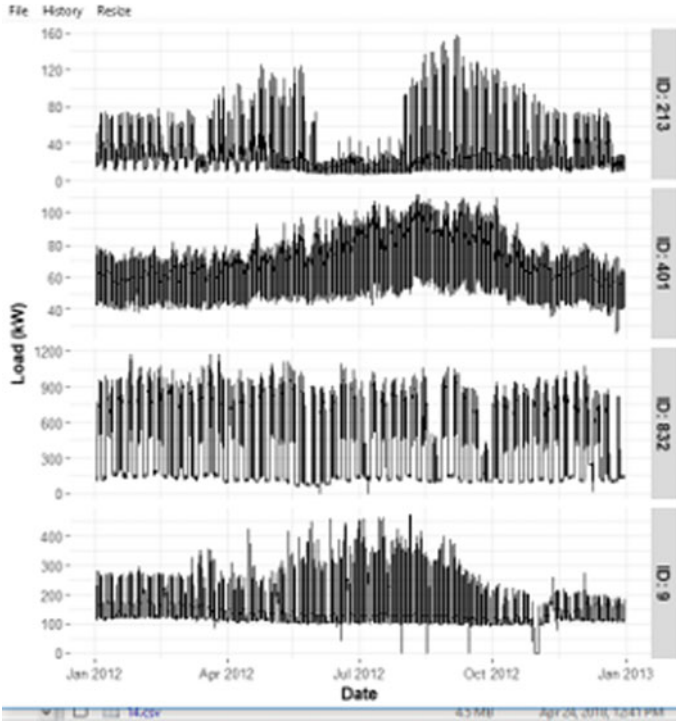


Fig. 10 Load versus date for 4 industries

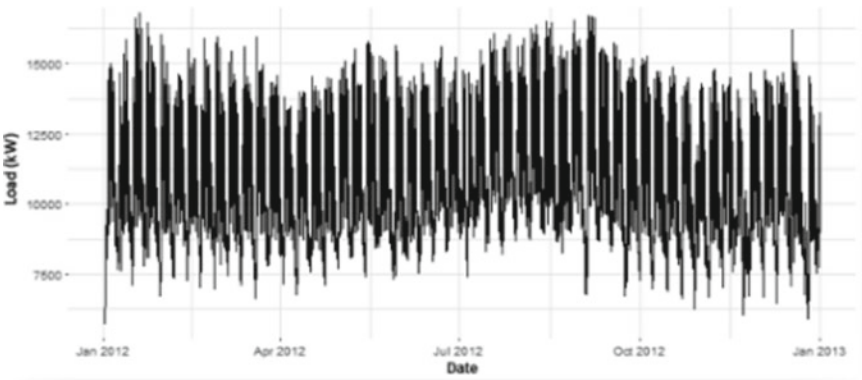


Fig. 11 Plot of aggregate consumption of load versus time series

5 Conclusion

In conclusion, we see that the project concentrates on the forecast model. This model helps us to make analysis about the daily consumption of various consumers of different industries and sub-industries with respect to time series. While we construct the histogram for plotting various consumers on map we observe that majority of the buildings are under 20,000 m². Through density plot we analyze that commercial buildings have viable size whereas food sales and storage buildings have smaller size. Through plotting the mean we get to know that the largest consumer of electricity is manufactures, shopping center, and business services whereas schools have the lowest consumption. The medial plot tells about the viable relation between the load and sq_m area. The forecast models help us to find the aggregate consumption to be 45%. The forecast model also helps to find out daily profile of consumer or word area.

6 Future Scope

In this paper, we have not made a forecast model for weekly or monthly consumption which can also be done further. These models can be further used to make comparisons between models of two different attributes. ARIMA can be used to predict more accurate forecast. By making the observation on consumption we can mark those areas where consumption is high and try to reduce it for conserving energy. In future methods like regression or neural network approach can be used for forecasting.

References

1. Herodotou, H., et al. (2011). Starfish: A self-tuning system for big data analytics. In *Cidr* (Vol. 11. No. 2011).
2. Quilumba, F. L., et al. (2015). Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities. *IEEE Transactions on Smart Grid*, 6(2), 911–918.
3. LaValle, S., et al. (2011). Big data, analytics and the path from insights to value. *MIT Sloan Management Review*, 52(2), 21.
4. Roy, C., Pandey, M., & Rautaray, S. S. (2018). A proposal for optimization of horizontal scaling in big data environment. In *Advances in data and information sciences* (pp. 223–230). Singapore: Springer.
5. Thusoo, A., et al. (2009). Hive: A warehousing solution over a map-reduce framework. *Proceedings of the VLDB Endowment*, 2(2), 1626–1629.
6. Chu, C.-T., et al. (2007). Map-reduce for machine learning on multicore. In *Advances in neural information processing systems*.
7. Patel, A. B., Birla, M., & Nair, U. (2012). Addressing big data problem using hadoop and map reduce. In *2012 Nirma University International Conference on Engineering (NUiCONE)*. IEEE.

8. Deshmukh, A. P., & Pamu, K. S. (2012). Introduction to hadoop distributed file system. *IJEIR*, 1(2), 230–236.
9. Ananthanarayanan, G., et al. (2010). Reining in the outliers in map-reduce clusters using Mantri. *OSDI*, 10(1).
10. Sitto, K., & Presser, M. (2015). *Field guide to hadoop: an introduction to hadoop, its ecosystem, and aligned technologies*. O'Reilly Media, Inc.
11. Feeney, L. M., & Nilsson, M. (2001). Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 2001* (Vol. 3). IEEE.
12. Efthymiou, C., & Kalogridis, G. (2010). Smart grid privacy via anonymization of smart metering data. In *2010 First IEEE International Conference on Smart Grid Communications (Smart-GridComm)*. IEEE.
13. Kavousian, A., Rajagopal, R., & Fischer, M. (2013). Determinants of residential electricity consumption: Using smart meter data to examine the effect of climate, building characteristics, appliance stock, and occupants' behavior. *Energy*, 55, 184–194.
14. Borthakur, D., et al. (2011). Apache hadoop goes realtime at facebook. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*. ACM.
15. Michael, Katina, & Miller, Keith W. (2013). Big data: New opportunities and new challenges [guest editors' introduction]. *Computer*, 46(6), 22–24.
16. Das, N., et al. (2018). Big data analytics for medical applications.
17. Jiang, L., Li, B., & Song, M. (2010). The optimization of HDFS based on small files. In *2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT)*. IEEE.
18. Szargut, J., Morris, D. R., Steward, F. R. (1987). Exergy analysis of thermal, chemical, and metallurgical processes.
19. Borthakur, D. (2008). The hadoop distributed file system: Architecture and design.
20. Liu, X., et al. (2009). Implementing WebGIS on hadoop: A case study of improving small file I/O performance on HDFS. In *IEEE International Conference on Cluster Computing and Workshops, 2009. CLUSTER'09*. IEEE.
21. Leverich, J., & Kozyrakis, C. (2010). On the energy (in) efficiency of hadoop clusters. *ACM SIGOPS Operating Systems Review*, 44(1), 61–65.

Movie Recommendation System



S. Rajarajeswari, Sharat Naik, Shagun Srikant, M. K. Sai Prakash
and Prarthana Uday

Abstract Recommender systems are called information filtering tools, which use big data to recommend likes of the user according to their preference and interest. Moreover, they also help in matching users with similar tastes and interests. Due to this, a central part of websites and e-commerce applications is taken up by the recommender systems. Systems using recommendation algorithms like Collaborative Filtering, Content-Based Filtering, etc., are called Recommendation systems. Recommendation systems are transforming the way quiescent websites corresponds with their users. Rather than providing an orthodox experience in which users search for the products which they want and potentially buy products, recommender systems increase communication to provide a better experience. The work would be to implement both collaborative as well as content-based recommenders available and try to extend the knowledge obtained to a more efficient hybrid model. Then benchmark these hybrid algorithms for accuracy and computation time as well.

Keywords Recommendation systems · Collaborative · Cosine similarity · SVD · Machine learning · Python · Surprise library

1 Introduction

It is expected that in future with the increased use of computers, the use of advanced and latest technologies by users and professionals and decision-makers will increase a lot. The basic idea of recommendation systems is that, if some users share the same interests, or maybe in the past, e.g., they liked the same book, they might also have similar tastes in the future. Most recommender systems take three basic approaches: simple recommender, collaborative filtering, and content-based filtering. A hybrid approach can combine these approaches to come up with a more optimal solution.

A recommendation system has become an indispensable component in various e-commerce applications. Recommender systems collect information about the user's

S. Rajarajeswari (✉) · S. Naik · S. Srikant · M. K. Sai Prakash · P. Uday
Department of Computer Science Engineering, RIT, Bangalore 560054, India
e-mail: raji@msrit.edu

© Springer Nature Singapore Pte Ltd. 2019

N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,

https://doi.org/10.1007/978-981-13-5953-8_28

329

preferences of different items (e.g., movies, tourism, TV, shopping,) by two ways, either implicitly or explicitly.

A movie recommendation system ideally analyzes factors like review, cast, plot, crew, genre, popularity and comes up with the best possible result. It takes into account a user preference and hence provides results based on the user's personal taste and choice. By increasing efficiency and accuracy of recommender systems, users expect lightning fast results and different types of recommender systems available today provide the same. There are four different kinds of recommendation systems, improving the accuracy and user personalization with each.

The four systems are the following:

Simple Recommendation System: A simple recommender is the most basic recommendation system available to us. It usually compares a metric/weight for all the available entities and provides the recommendation on the basis of this comparison.

Content-Based Recommendation System: This recommendation system involves the use of more attributes from the dataset to provide a well-filtered recommendation to the user. In this project, at first, the recommender just provides a recommendation using the movie overview and taglines. But there is a possibility that a user watches a certain movie for the director or the cast rather than the ratings. For this reason, the recommender takes into consideration the director, cast, genre, and keywords to provide a better search result. This algorithm makes use of k-means algorithm through cosine similarity to find how similar two movies are and hence can be recommended together.

Collaborative Recommendation System: Both the recommendation systems mentioned above are efficient but they are not personalized. They would show the same results to all the users. But that should not be the real case. Movies should be recommended based on the user's personal taste. That is where collaborative filtering comes into the picture.

Here, the movie ratings provided by a particular user and using the SVD surprise library functions, evaluate the RMSE and MAE values. Since these values are comparable, a predict function is used that provides an estimate for a movie for that particular user using his ratings. This way, the movie recommendations provided to each user would be different depending on their tastes.

Hybrid Recommendation System: Having demonstrated all the three available recommendation models, a hybrid system is developed which uses the concepts of both content base as well as collaborative recommendation systems. Here, the user's ID and movie title are taken as the inputs. Using the movie title, 30 similar movies are collected in a database based on the title, cast, crew, overview, etc., using the cosine similarity and linear kernel concept. And then using the user's ID, his ratings are procured which is then passed as an argument to the `svd.predict()`. Through the list of those 30 movies, the ones most suited to the user's taste are recommended.

2 Related Work

In a Movie Recommender System called “MOVREC” by Manoj Kumar, D. K. Yadav, Vijay Kr Gupta.

It uses collaborative and content-based filtering using the k-means algorithm. This asks a user to select his own choices from a given set of attributes and then recommends him a list of based on the cumulative weight of different attributes using k-means algorithm [1].

The disadvantage is that it does not use large datasets, hence there will not be meaningful results.

In a paper by Rahul Kataria and Om Prakash named An effective movie recommender system which uses cuckoo search, a bio-inspired algorithm such as cuckoo search has exclusive background sensing abilities and employs a special method to facilitate the evolution of continuing resolutions into novel and quality recommendations by generating clusters with reduced time [2].

It has a limitation where if the initial partition does not turn out well then the efficiency may decrease.

In 2007 Weng, Lin, and Chen performed an evaluation study which says using multidimensional analysis and additional customer’s profile increases the recommendation quality. Weng used MD recommendation model (multidimensional recommendation model) for this purpose. Multidimensional recommendation model was proposed by Tuzhilin and Adomavicius [3].

2.1 Sajal Halder: Movie Recommendation Using Movie Swarm

Movies swarm mining that mines a set of movies, which are suitable for producers and directors for planning new movie and for new item recommendation. Popular movie mining can be used to solve new user problems. It has the capability of handling both new users and new items. It is better than content-based and collaborative approaches when it comes to new users. This method, however, has a drawback of finding a group of users depending on the genre [4].

Yueshen Xu: Collaborative recommendation with User-Generated Content (UGC).

It proposes a UGC-based collaborative recommendation which integrates probabilistic matrix factorization and collaborative regression models. It reaches a higher prediction accuracy than most of the available models. It is efficient when it comes to cold start problem. The cost per iteration is more as compared to traditional CTR models [5] (Table 1).

Table 1 Comparative study related works and algorithms

S. No.	Technique	Algorithm	Drawback
1	Collaborative and content based	K-means	It is not suited for large datasets
2	Cuckoo search	Clustering	Efficiency decreases if the initial partition is not proper
3	Movie swarm	Mining	Drawback of finding a group of users based on the genre
4	User-generated content (UGC)	Regression analysis	Cost per iteration is more when compared to traditional models

3 Design

See Table 2.

General Architecture Diagrams

See Fig. 1.

Simple Recommender System

In Fig.2 Simple recommender gives a prediction based on the weighted rating (WR) calculated. It takes in vote_count and vote_average from Movies_metadata.csv.

The TMDB Ratings is used to come up with our Top Movies Chart. IMDB’s weighted rating formula is used

Mathematically, it is represented as follows:

$$\text{Weighted Rating (WR)} = (v/v + m.R) + (m/v + m.C)$$

Table 2 Description of attributes and dataset used in the project

Dataset name (.csv)	Description
Movies_metadata	Contains information about 45,000 movies featured in TMDB
Keywords	Contains movie plot keywords available in the form of a stringified JSON object
Credits	Contains cast and crew information, also saved as a stringified JSON object
Links	Contains TMDB and IMDB Ids of all the movies
Links_small	Contains a subset of links database with 9000 movie entries
Ratings_small	Contains a subset of 100,000 ratings from 700 users on 9000 movies

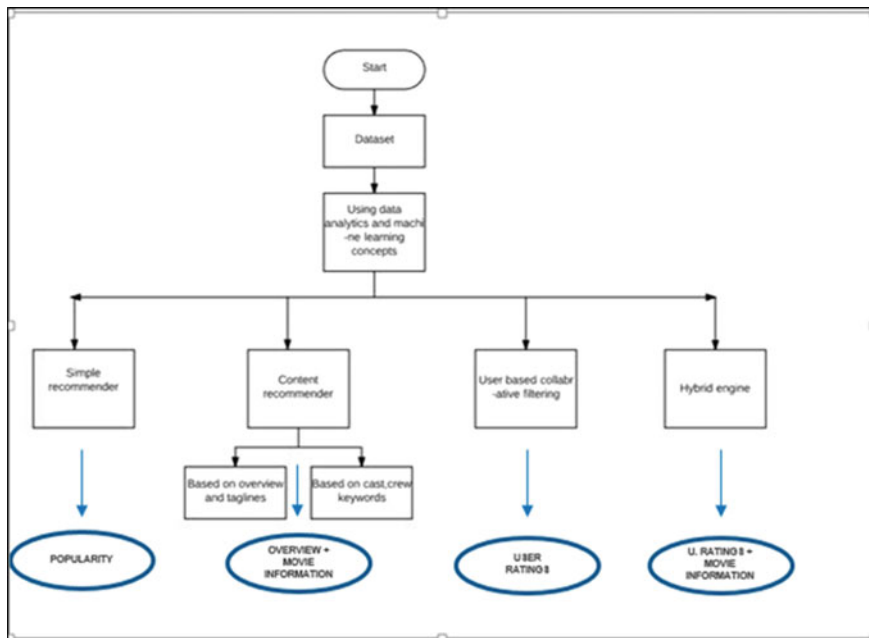


Fig. 1 General architecture of the recommendation system

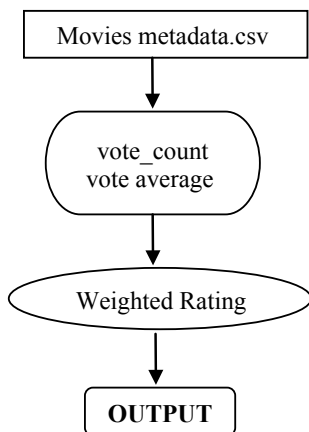


Fig. 2 Simple recommender

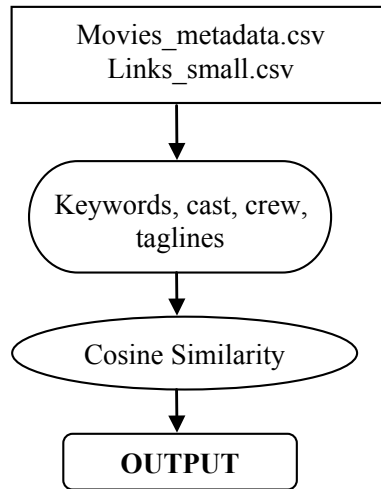


Fig. 3 Content-based recommender

Content Based Recommender System

In Fig.3, Content-based recommender gives prediction based on cosine similarity. It takes in keywords, taglines from `Movies_metadata.csv` & `links_small.csv`, it also takes director and cast as attributes.

To personalize our recommendations more, an engine is built that computes the similarity between movies based on certain metrics and suggests movies that are most similar to a particular movie that a user liked previously. Since movie metadata (or content) is used to build the engine, this is also known as Content-Based Filtering.

Cosine similarity is used to calculate a numeric value, which denotes the similarity between two movies.

Collaborative Recommender System

In Fig.4, Collaborative recommender gives prediction using singular value decomposition (svd). It takes in `userID` and `movieID` from `ratings_small.csv`. This is more personalized as it gives different suggestions to different users based on their taste.

A technique called Collaborative Filtering is used to make recommendations to People. Collaborative Filtering is based on the idea that users similar to me can be used to predict how much I will like a particular product or service those users have used/experienced but I have not.

Surprise library is being used that uses extremely powerful algorithms like Singular Value Decomposition (SVD) to minimize RMSE (Root Mean Square Error) and gives great recommendations to users.

Mean value is obtained for Root Mean Square Error, then we train our dataset and arrive at predictions.

Hybrid Recommender System

In Fig.5, Hybrid recommender gives a prediction based on the svd and cosine similarity. It uses Movies_metadata.csv as well as ratings.csv.

In this, content and collaborative filtering techniques are brought together.

Input given will be User ID and Title of a movie the user likes.

Output will be similar movies sorted on the basis of expected ratings by that particular user which is more personalized.

Pseudocode:

a. **Simple recommendation system**

build_chart(genre)

Input: genre to be searched for.

Output: Top 250 recommendation

Fig. 4 Collaborative recommender

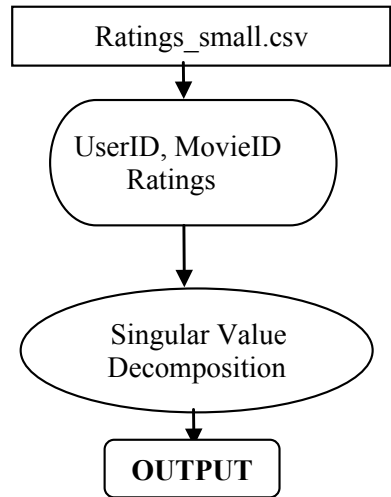
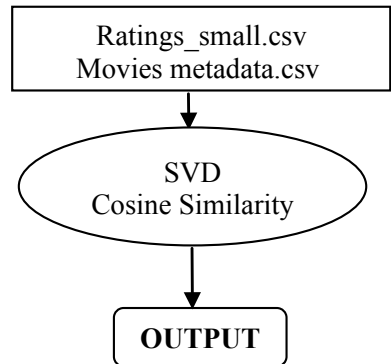


Fig. 5 Hybrid recommender



```

df= movies of 'genre'
vote_counts= number of votes for the genre
vote_average= average votes for the genre
C=vote_averages.mean()
m=vote_counts.quantile(percentile)
v=x['vote_count']
R=x['vote_average']
qualified['wr']=(v/(v+m) * R) + (m/(m+v) * C)
return qualified

```

b. Content-Based recommendation system

get_recommendations(title)

Input: Name of a movie

Output: Top 30 movie recommendation

```

Idx = id of the title
keywords= get keywords. Strip them of spaces.
Director= get_director(crew){if (job=='director') return name }
Cast= first three names in the cast
Count-_matrix=Get the count vectorizer matrix to calculate the cosine similarity
using keywords, director, cast.
Cosine_similarity=Get the cosine similarity.
Return result.

```

c. Collaborative Recommendation System

recommendations(userId, title)

Input: User's ID and title of the movie

Output: Movies which cater to user's preference

```

Data = Use user id and movie id to get this
svd = Calling the SVD function.
Evaluate RMSE and MAE
Train data set to arrive at function
svd.train(trainset)
Predict using SVD and return the movies.

```

Proposed System

The proposed hybrid system takes features from both collaborative as well as content-based recommendation to come up with a system, which is more personalized for users.

Technique

Cosine similarity technique is used to come up with movies similar to the one given in the search engine. The selected 30 movies are then compared using the user's ratings and the SVD. Hence, the search advances to predict the movies most suitable for the user.

Algorithm:

The algorithm used for the hybrid engine is as follows:

def hybrid(userId, title):

- Obtain the movieID (Idx) based on the title.
- Obtain the sim_scores using cosine_similarity on the procured Idx
- Use cosine similarity to obtain 30 movies similar to Idx using 'vote_count', 'year', 'vote_average' etc.
- Using svd.predict(), calculate the RMSE between the suggested movies as well as the movie passed as an argument.
- Sort the list obtained.
- Return the top 10 movies of the list.

4 Results

Content and Collaborative filtering techniques are brought together to build an engine that gives movie suggestions to a particular user based on the estimated ratings that it had internally calculated for that user. It makes use of cosine similarity concept to find similar movies and the SVD prediction to estimate the movie recommendation as per the user's taste.

Simple Recommendation System:

See Fig. 6.

Content-Based Recommendation System:

See Fig. 7.

Collaborative Filtering:

In Fig. 8, the RMSE and MAE values are evaluated by using SVD Algorithm available in Surprise library to minimize the RMSE.

Hybrid Recommender System:

See Fig. 9.

For our hybrid recommender, we get different recommendations for different users although the movie is the same. Hence, our recommendations are more personalised and tailored toward particular users.

```
In [8]: def weighted_rating(x):
v = x['vote_count']
R = x['vote_average']
return (v/(v+m) * R) + (m/(m+v) * C)

In [9]: qualified['wr'] = qualified.apply(weighted_rating, axis=1)

In [10]: qualified = qualified.sort_values('wr', ascending=False).head(250)

In [11]: qualified.head(15)
Out[11]:
```

	title	year	vote_count	vote_average	popularity	genres	wr
15480	Inception	2010	14075	8	29.1061	[Action, Thriller, Science Fiction, Mystery, A...	7.917588
12461	The Dark Knight	2008	12269	8	123.167	[Drama, Action, Crime, Thriller]	7.905871
22879	Interstellar	2014	11187	8	32.2135	[Adventure, Drama, Science Fiction]	7.897107
2843	Fight Club	1999	9678	8	63.8696	[Drama]	7.881753
4863	The Lord of the Rings: The Fellowship of the Ring	2001	8892	8	32.0707	[Adventure, Fantasy, Action]	7.871787
292	Pulp Fiction	1994	8670	8	140.95	[Thriller, Crime]	7.868660
314	The Shawshank Redemption	1994	8358	8	51.6454	[Drama, Crime]	7.864000
7000	The Lord of the Rings: The Return of the King	2003	8226	8	29.3244	[Adventure, Fantasy, Action]	7.861927
351	Forrest Gump	1994	8147	8	48.3072	[Comedy, Drama, Romance]	7.860656
5814	The Lord of the Rings: The Two Towers	2002	7641	8	29.4235	[Adventure, Fantasy, Action]	7.851924
256	Star Wars	1977	6778	8	42.1497	[Adventure, Action, Science Fiction]	7.834205
1225	Back to the Future	1985	6239	8	25.7785	[Adventure, Comedy, Science Fiction, Family]	7.820813
834	The Godfather	1972	6024	8	41.1093	[Drama, Crime]	7.814847
1154	The Empire Strikes Back	1980	5998	8	19.471	[Adventure, Action, Science Fiction]	7.814099
46	Se7en	1995	5915	8	18.4574	[Crime, Mystery, Thriller]	7.811669

Fig. 6 Top 250 movies based on the popularity/user ratings

```
In [32]: get_recommendations('The Godfather').head(10)
Out[32]:
```

973	The Godfather: Part II
8387	The Family
3509	Made
4196	Johnny Dangerously
29	Shanghai Triad
5667	Fury
2412	American Movie
1582	The Godfather: Part III
4221	8 Women
2159	Summer of Sam

Name: title, dtype: object

```
In [33]: get_recommendations('The Dark Knight').head(10)
Out[33]:
```

7931	The Dark Knight Rises
132	Batman Forever
1113	Batman Returns
8227	Batman: The Dark Knight Returns, Part 2
7565	Batman: Under the Red Hood
524	Batman
7901	Batman: Year One
2579	Batman: Mask of the Phantasm
2696	JFK
8165	Batman: The Dark Knight Returns, Part 1

Name: title, dtype: object

```
In [34]: credits = pd.read_csv('credits.csv')
keywords = pd.read_csv('keywords.csv')

In [35]: keywords['id'] = keywords['id'].astype('int')
credits['id'] = credits['id'].astype('int')
md['id'] = md['id'].astype('int')

In [36]: md.shape
Out[36]: (45463, 25)
```

Fig. 7 Similar 30 movies based on overview, taglines, director, cast


```
In [65]: svd = SVD()
evaluate(svd, data, measures=['RMSE', 'MAE'])
Evaluating RMSE, MAE of algorithm SVD.
-----
Fold 1
RMSE: 0.8923
MAE: 0.6891
-----
Fold 2
RMSE: 0.8982
MAE: 0.6883
-----
Fold 3
RMSE: 0.8995
MAE: 0.6927
-----
Fold 4
RMSE: 0.9060
MAE: 0.6967
-----
Fold 5
RMSE: 0.8935
MAE: 0.6890
-----
Mean RMSE: 0.8979
Mean MAE : 0.6912
-----

Out[65]: CaseInsensitiveDefaultDict(list,
{'mae': {0.68907711209711431,
0.68827862853515431,
0.69268970936514584,
0.69674717243046147,
0.68899872532741335},
'rmsse': {0.8922661712965325,
0.89823710302444226,
0.8994505602481602,
0.90599131930031884,
0.89346740974468719}})
```

Fig. 8 The RMSE and MAE Values

```
In [76]: hybrid(500, 'Avatar')
Out[76]:
```

	title	vote_count	vote_average	year	id	est
8401	Star Trek Into Darkness	4479.0	7.4	2013	54138	3.561917
8658	X-Men: Days of Future Past	6155.0	7.5	2014	127585	3.365317
974	Aliens	3282.0	7.7	1986	679	3.156818
4347	Piranha Part Two: The Spawning	41.0	3.9	1981	31646	3.056620
2014	Fantastic Planet	140.0	7.6	1973	16306	3.024476
8419	Man of Steel	6462.0	6.5	2013	49521	3.006235
1668	Return from Witch Mountain	38.0	5.6	1978	14822	3.004598
4017	Hawk the Slayer	13.0	4.5	1980	25628	2.999030
922	The Abyss	822.0	7.1	1989	2756	2.970150
1621	Darby O'Gill and the Little People	35.0	6.7	1959	18887	2.923382

Fig. 9 Movie prediction for User ID 500 with movie given as 'Avatar'

5 Conclusion

Upon surveying through different algorithms and models present for movie recommendation systems, we try to come up with a hybrid recommendation algorithm in order to provide the most accurate recommendation to a user based on his ratings and preference.

Recommendation systems of the future will work in e-commerce to offer a more visceral, immersive, and well-rounded experience for every step of a customer's journey. In addition, once a successful hybrid engine comes into the picture, the same algorithm can be extended to other types of recommendation systems. This would

result in an economic boom for the various e-commerce websites and applications with better user-specific experience.

References

1. Kumar, M., Yadav, D. K., Singh, A., & Gupta, V. K. A movie recommender system “MOVREC”. <https://pdfs.semanticscholar.org/e13e/b41de769f124b3c91771167fb7b01bc85559.pdf>.
2. Katarya, R., Verma, O. P. An effective collaborative movie recommender system with cuckoo search https://ac.els-cdn.com/S1110866516300470/1-s2.0-S1110866516300470-main.pdf?_tid=7a06b6fe-c95d-11e7-b816-00000aacb362&acdnat=1510679077_68b65f60dcc55e2aeac3588dab49e4.
3. Hande, R., Gutti, A., & Shah, K. Moviemender—A movie recommender system. <https://pdfs.semanticscholar.org/e13e/b41de769f124b3c91771167fb7b01bc85559.pdf>.
4. Halder, S. Movie recommendation system based on movie swarm. http://www.academia.edu/2396000/Movie_Recommendation_System_Based_on_Movie_Swarm.
5. Xu, Y. Collaborative recommendation with User Generated Content (UGC). https://www.researchgate.net/profile/Yueshen_Xu/publication/280733284_Collaborative_Recommendation_with_User_Generated_Content/links/55c3cf3408aeb97567401ddb/Collaborative-Recommendation-with-User-Generated-Content.pdf.

A Semiautomated Question Paper Builder Using Long Short-Term Memory Neural Networks



Rajarajeswari Subramanian, Akhilesh P. Patil, Karthik Ganesan and T. S. Akarsh

Abstract Through this research, we propose a model to generate different sets of question papers automatically using deep learning methodologies. We first develop LSTM classification models to classify questions into Bloom's level of taxonomy, chapter name and we have an LSTM prediction model to predict the marks to be allocated. These Deep Learning techniques can help in reducing human effort in deciding the marks, the section, and Bloom's level to be allocated to a question. Given a pool of hundreds of questions, our deep learning models develop a knowledge base consisting of questions and predicted attributes marks, Bloom's level, and chapter name. We then we have a randomization algorithm to pick the questions for different units of the question paper, keeping the standards to be maintained and even distribution of questions across all topics.

Keywords Bi-directional long short-term memory networks · Randomization technique · Classification

1 Introduction

In the current era of artificial intelligence, most of the processes are being automated. This has eliminated the human effort required and improved the accuracy with which the process can be carried out. Automation has greatly helped the industries and is yet to find its application in the educational background. Considering this, we have proposed a model which may help educational institutes automatize the process of generating a question paper. We can train the model in such a way that the marks, bloom's level, and chapter classification of a particular question can be predicted from the question as an input.

R. Subramanian · A. P. Patil (✉) · K. Ganesan · T. S. Akarsh
Department of Computer Science and Engineering, Ramaiah Institute of Technology, Bengaluru,
India
e-mail: apatil1997@gmail.com

© Springer Nature Singapore Pte Ltd. 2019
N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_29

341

1.1 Obtaining the Dataset

The data set consists of the set of questions from a textbook. The Bloom's level, chapter name, and the marks distribution is assigned according to each question and appended to the database. The entry to each of the categories is done in the form of a CSV file. This CSV file is read in the form of data-frames with the help of Pandas Library in Python to make the manipulation of data easier. The snapshot of the dataset is as shown in the below Figure.

1.2 Pattern of the Question Paper

The question paper to be set is of 200 marks, out of which the student is to attempt questions for 100 marks. The question paper consists of 5 individual units of 40 marks each. The student has to attempt either one of the questions from each unit which is 20 marks each. Each question also has a Bloom's Level associated with it. Our aim in building the question paper is to ensure that there are two questions in each unit of twenty marks each and the Bloom's Level is evenly distributed.

2 Related Work

In the paper cited in [1], a randomization technique was proposed to automatically generate a question based on the questions fed to the system according to a particular syllabus. The technique proposed here, which is a role-based model also takes care that the questions are not repeated. The paper cited in [2] also uses a randomization technique to generate sets of question papers, which are unique from each other. This method ensures that the questions generated are not the ones as in a particular question bank provided to students. A research cited in [3], used J2EE tools to design an automated question paper management system. Separate management modules were developed for the user, subject, and the classification of questions. The algorithm used was efficient enough to identify the subject, question type, and also the difficulty level. Some researches as cited in [4] also used fuzzy logic to generate non-repetitive question paper that distributed questions evenly throughout the chapters. Similarly, papers cited in [5] select questions based on the Blooms Levels of Taxonomy. This model was developed based on genetic algorithms. In the paper cited in [6], an adaptive technique was proposed for the generation of question paper, but the drawback of this model was that it assumes the entry of questions made in the database are error free. The question paper generator proposed in our research is different from all the other researches in the sense that separate classification model was built using the bidirectional Long Short-Term Memory Networks for each of

the attributes. The attributes taken into consideration for classification were chapter name, marks categorization and Bloom's Level of Taxonomy.

3 Techniques Used

The deep learning methodologies like Long Short-Term Memory Networks, Bidirectional Long Short-Term Memory Networks have been used to classify the text in the questions into slots such as the mark allotment to each question, unit-wise markings and Bloom's level of taxonomy. The following section gives a brief about the working of each of these models to serve our cause.

3.1 *Bidirectional Long Short-Term Memory Networks*

Bidirectional LSTMs can extract the most out of an input sequence by running over the input sequence both in the forward direction as well as in the backward direction. The architecture is developed by creating another copy of the first recurrent layer in the neural network so that there are two layers side-by-side. The next step is to provide the input sequence as it is to the first layer and feeding an inverted copy to the second duplicated layer. This approach is effectively used along side the Long Short-Term Memory Networks. The bidirectional LSTM is as shown in Fig. 1.

3.2 *Long Short-Term Memory Networks*

Humans do not start to think from scratch every time they read some new article. As a person reads an article, he or she reads every word based on the understandings of the previous words. Our thoughts can retain the things of the past. A simple neural network will not have the ability to do so. Recurrent neural networks can overcome this shortcoming, due to the presence of a looping methodology in the hidden layers.

Long Short-Term Memory networks are a special kind of Recurrent Neural Networks, which have the capability of learning from previously computed output results. In our problem where we predict the sequence of words that have to be a part of the question, the whole idea here is that the sentence in our data frame is a sequence of words which we may consider as a vector. The RNN that obtains the vectors as input and considers the order of vectors to generate predictions. From the embedding layer, the new representations will be passed to LSTM cells. These will add recurrent connections to the network, so we can include information about the sequence of words collected. Finally, the cells of the LSTM will go to the sigmoid output layer. We use a sigmoid because we are trying to predict the final understanding of the sentence. LSTMS have chain-like structure, the repeating module has a

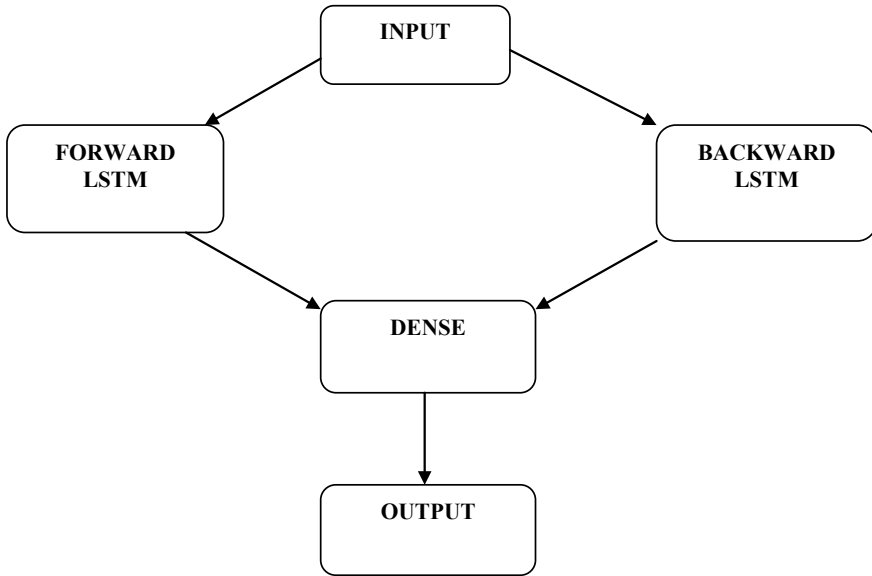


Fig. 1 The functioning of the bidirectional long short-term memory network

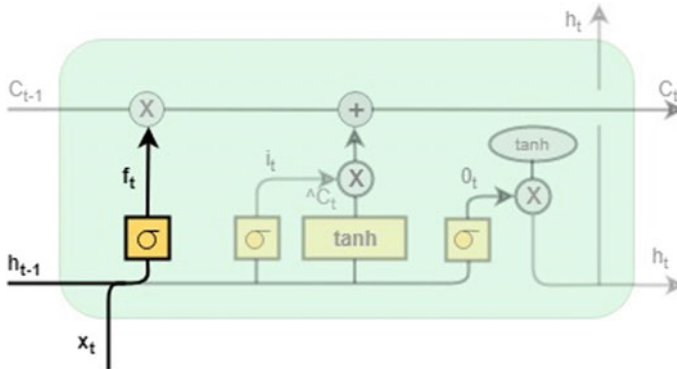


Fig. 2 First step in the LSTM

different structure. Instead of having a single neural network layer, there are four, interacting in a very special way. The basic structure of the LSTM architecture is as shown in Fig. 2.

In the first step, the LSTM model decides by a sigmoid layer, looks at a previously computed hidden layer- $h(t - 1)$ and the present output- $x(t)$ and outputs a number between 0 and 1 for each number in state $C(t - 1)$. The value of f_t can be found using the equation below (Fig. 3).

$$f_t = \sigma(W_f[h_{t-1}, x_t + b_f]) \tag{1}$$

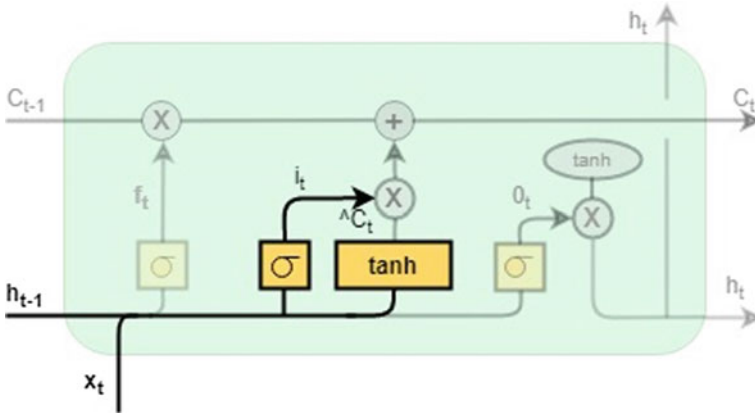


Fig. 3 Second step in LSTM

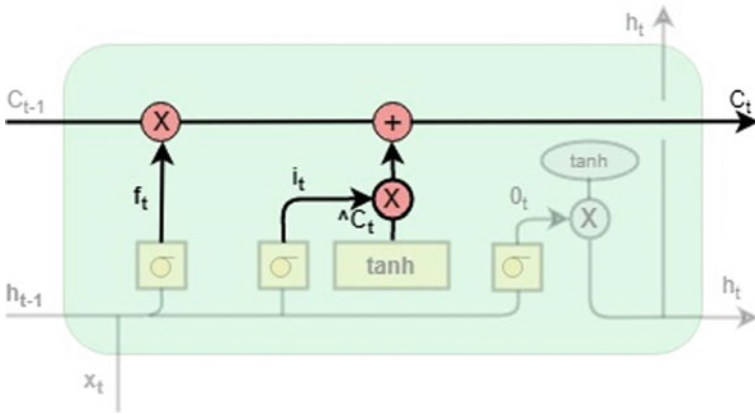


Fig. 4 Third step in LSTM

Second, we need to determine as to what is going to be stored in the successive cell. First, the sigmoid layer decides as to which values to update. Finally, the tanh layer vectorizes the values to form the C_{t1} values. The next step involves the combination of the two layers. i_t and C_{t1} can be found using the below equations (Figs. 4 and 5).

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{2}$$

$$C_{t1} = \tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{3}$$

In the next step, we update the old cell state $C(t - 1)$ to new state $C(t)$. In the case of our model, this is where we drop the information of the old state and add new information, as decided earlier. Here, C_{t-1} is the previous old cell state.

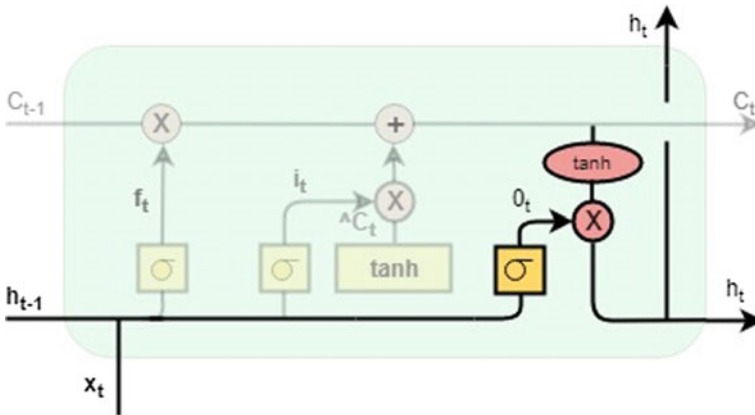


Fig. 5 Fourth step in LSTM

$$C_{t1} = f_t * C_{t-1} + i_t * C_t \tag{4}$$

Finally, we come up with the output that is the final information of a sentence. For the model, it might want to output information relevant the outcome of the result. The output o_t can be found using the below equations.

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * \tanh \tag{6}$$

3.3 Randomization Technique

3.3.1 Word Embeddings

Word Embeddings are used to represent two or more similar words with the same meaning. For the purpose of training, a pretrained genism model on Wikipedia data was used for converting words into vectors. Word Embeddings maps word to vectors in a high-dimensional space. If learnt the right way, these word embeddings can learn the semantic and the syntactic information of the words—i.e., similar words are close together and dissimilar words are further apart. This can be learnt from large size text files such as Wikipedia data. As an illustration, we have shown here the nearest neighbors from the word embeddings space for some words. This embedding space was learnt by the model that we propose through the following research. The example case of the word embeddings used is as shown in Table 1.

Table 1 Training example of word embeddings

Sunday	Training	Robot	Artificial intelligence	Time
Wednesday	Education	Machine	Knowledge engineering	Schedule
Monday	Building	Automaton	Neural net	Times
Sunday	Tuition	Artificial Intelligence	Machine learning	Schedules
Monday	Tune up	Automation	NLP	Dinnertime
Friday	Practice	Gadget	Information retrieval	Ord
Saturday	Schooling	Android	Tokenization	f28

3.3.2 Building the Classification Model

A many to one Long Short-Term memory network is used to classify a large number of questions into three different categories as chapter number, marks allocated, and Bloom’s Level of Taxonomy. However, the LSTM classifier can also be built for attributes such as difficulty level. In this section, we go through how the classifier is built for each of the attributes.

For the chapter classifier a many to one Long Short-Term Memory Network was built such a way that the input layer is the sequence of words in question and the target variable is the attribute we are trying to predict such as chapter number to which the question may belong to. The model is trained for a large corpus of questions by allocating the chapter numbers, respectively. A feedback to the input layer is provided when there is an error in the classification and the weights are updated accordingly.

Similarly, the classifier was built for the marks and Bloom’s Level attributes considering the range of marks for each question and the different levels of Bloom’s taxonomy as the target variables.

3.3.3 Randomization Algorithm

The classified questions were appended to the data set in the form of a CSV file. The CSV file was converted to a data frame for further processing to be carried out so that each of the classified questions can be written to a word file.

The pseudo-code for the randomization technique is as follows:

Step 1: Initializations:

- A dictionary with two sequences of marks that add up to a total of 20.
- A list with all the chapter numbers.
- A dictionary for matching Bloom’s Level of Taxonomy.
- A visited array to keep track of all the chapters taken into consideration for each unit.
- Two lists for the two-alternate question sets in a unit.

Step 2: Chose a random sequence number from the dictionary.

Step 3: For every value in the sequence, find the marks and Bloom's Level in the sequence. Also, select a random chapter number from the union set of chapter list and the visited array.

Step 4: Match the chapter selected and the marks selected with that in the data frame consisting of the classified chapters and the marks using a search technique.

Step 5: Append the chapter selected to the visited array and append the chapter name, marks and the Bloom's Level to the list initialized for the first alternate part of the unit.

Step 6: Initialize another list without the current selection of the random value chosen. Repeat steps 1 through 5 for choosing the second alternate question in the unit.

Step 7: In a similar way, repeat the procedure for all the five units for which the questions that must be set. Lastly, we append the lists to the word document to create the final copy of the question paper.

4 Results

Using the Bidirectional Long Short-term Memory Networks, classifiers were built for each of the attributes. The root mean square error method was used to find the accuracy of the marks prediction model and the misclassification error for Bloom's Level and chapter number classification models.

The root mean square error is found by taking the root of the variance of the predicted value of the classifier with that of the actual value of the classifier and the misclassification error is calculated from the confusion matrix. The accuracy for the models was found to be 82.4, 84.8, and 81.7%, respectively (Fig. 6).

The graph showing the comparison of the accuracies and errors of each model is shown in the figure below. The inverse relation graph is also shown in Fig. 7.

Fig. 6 Comparison of models

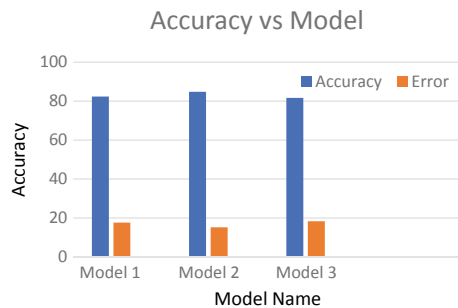
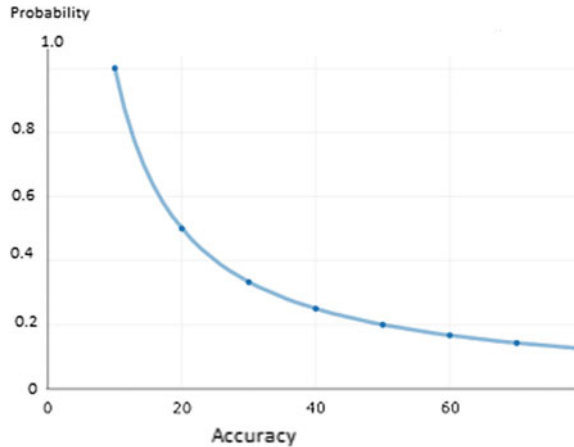


Fig. 7 Graph of accuracy versus probability



References

1. Bhirangi, R., & Bhoir, S. *Automated question paper generation system*. Computer Engineering Department, Ramrao Adik Institute of Technology, Navi Mumbai, Maharashtra, India.
2. Gangar, F. K., Gopal Gori, H., & Dalvi, A. *Automatic question paper generator system*. Department of Information Technology, K. J. Somaiya College of Engineering Vidhyavihar, Mumbai-77.
3. Cena, G., Dong, Y., Gao, W., Yu, L., See, S., Wang, Q., Yang, Y., & Jiang, H. A implementation of an automatic examination paper generation system.
4. Mohandas, M., Chavan, A., Manjarekar, R., Karekar, D., Qing, L., & Byeong Man, K. (2003). Automated question paper generator system. In *IEEE/WIC International Conference on Clustering Approach for Hybrid Recommender System, in Web Intelligence, 2003. WI 2003. Proceedings*, (pp. 33–38).
5. Leekha, A., Barot, T., & Salunke, P. Bharati Vidyapeeth College of Engineering Sector-7, C.B.D. Belpada, Navi Mumbai-400614, India.
6. Naik, K., Sule, S., Jadhav, S., & Pandey, S. *Automatic question paper generation system using randomization algorithm*.
7. Divate, M., & Salgaonkar, A. *Automatic question generation approaches and evaluation techniques*.

Time-Critical Transmission Protocols in Wireless Sensor Networks: A Survey



Archana R. Raut, S. P. Khandait and Urmila Shrawankar

Abstract Wireless Sensor Network (WSN) is an extremely important tool for closely monitoring, understanding and controlling application processes to the end users. The main purpose behind installing the wireless sensor network is to make real-time decisions based on data received from the sensor nodes. This data transmission from sensor nodes to the base station is considered to be very complicated because of the resources and communication capability constraints of various sensor nodes and enormous amount of data is generated by WSNs. Real-time applications in WSN like mission-critical monitoring, surveillance systems, etc., demands well-timed and reliable delivery of data. For such applications, besides energy, Quality of Services (QoS) routing, i.e., requirement of message delivery timeliness is also one of the significant issues. Based on the type of application, it is essential to grant different levels of QoS in WSNs. In this paper, QoS requirements for mission-critical WSNs applications are highlighted and existing QoS-aware protocols to support such applications are discussed with their boundaries in that domain.

Keywords Wireless sensor network · Quality of service · Mission-critical · Delay · Reliability

A. R. Raut (✉)

Research Scholar, Department of Information Technology, G. H. Rasoni College of Engineering, Nagpur 440016, India

e-mail: archana.kakade5@gmail.com

S. P. Khandait

Research Supervisor, Department of Information Technology, G. H. Rasoni College of Engineering, Nagpur 440016, India

U. Shrawankar

Department of Information Technology, G. H. Rasoni College of Engineering, Nagpur 440016, India

© Springer Nature Singapore Pte Ltd. 2019

N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,

https://doi.org/10.1007/978-981-13-5953-8_30

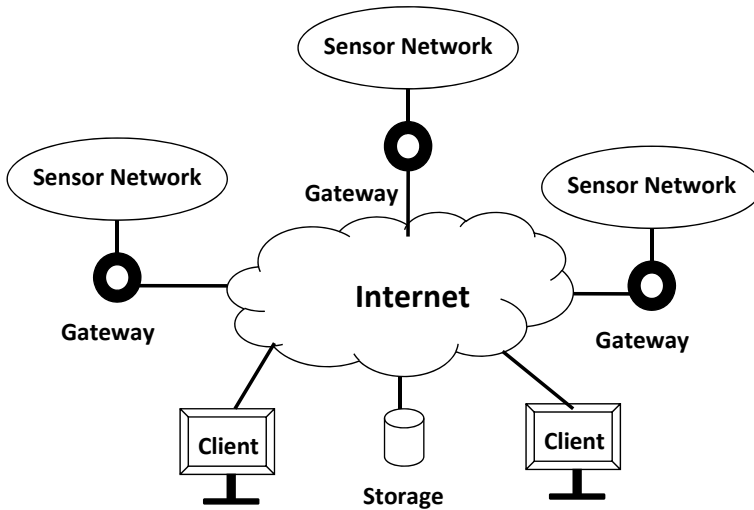


Fig. 1 Wireless sensor network

1 Introduction

The most important capability of WSN is to sense the environmental parameters and communicate it between sensors nodes deployed in the broad area of attention. Grouping and installing huge number of sensor nodes in the interested area in an unplanned or planned manner can do this. These nodes correspond mutually, wirelessly in self-organized way after installation and send collected data to base station as shown in Fig. 1. WSN have numerous applications areas [1–4]. Some major applications are in agriculture, in wildlife monitoring, in military surveillance, for industry control applications [5, 6], environmental sensing, and personal health concern as well as for home security and automation, etc. In health care applications, wireless devices are used for patient monitoring and personal health care. In remote monitoring, the necessary requirement is that wireless systems can be used in place of wired systems to reduce cost of wiring and to permit various types of measurements. Remote monitoring also includes Environmental monitoring of water, soil and air monitoring, Industrial machine monitoring, Structural monitoring for various constructions, object tracking and in military. In all these applications, reliable and real-time monitoring is the necessary requirement.

Because of constraints like limited power, bandwidth, memory, etc., in sensor networks, energy efficiency of nodes turns out to be an important parameter for designing an efficient data collection schemes for sensor networks. Major part of the energy of the sensors is required for the data transmission in sensor network. As sensor nodes life in WSN is based on energy remained in the node so, reducing the energy utilization enlarge the network lifetime. With the emergence of critical, multimedia and real-time (RT) applications, QoS is becoming a significant parameter in

WSNs. So, in addition to conserving the overall network energy, QoS is also a key measure to compute the overall network performance in RT delays sensitive WSN applications [4–6]. As there is tradeoff between various application necessities such as energy efficiency in addition to delaying performance, systems that grant real-time services with guaranteed data delivery time, assured reliability and additional required QoS-related parameters are essential in WSNs. Data transmission in such critical applications demands both energy and QoS support for effective accessibility of the collected measurements and efficient utilization of sensor resources. Therefore, time-critical and reliable transmissions of collected data are two most important parameters QoS provisioning in WSNs. In this paper, research is performed in the direction of concentrating on the issues in existing data transmission schemes considering the reliability and delay in time-critical WSN applications.

The entire paper is organized as given below. Section 2 discusses different challenges in providing QoS in sensor networks. Some QoS-aware MAC and routing protocols are reviewed in Sect. 3. A comparative revision on some QoS protocols based on few important parameters in context of WSNs is also done. Section 4 discusses some issues that need to attend to enhance the performance of WSNs. Last, in Sect. 5, concluding observations in the literature are presented with some research directions in critical WSN applications.

2 Quality of Service

2.1 QoS Challenges in WSNs

In WSNs, different layers demand different QoS requirements to get better system throughput [2–4]. Different application areas demand different kinds of QoS related to various layers. To conserve RT applications the following are the main challenges related to QoS in WSNs (Fig. 2). Amongst lots of WSNs applications, with a range of QoS requirements, the most challenging application can be found in real-time WSNs [4–6].

- **Resource constraints:** It mainly involves energy constraints, bandwidth constraints, memory constraints, processing capability, and transmission power constraints.
- **Node Deployment:** Sensor nodes can be deployed randomly or in a pre-planned manner.
- **Topology changes:** In WSN, topology rapidly changes in view of sensor node mobility, link failures between nodes, faulty behavior of nodes or due to energy reduction of nodes.
- **Data Redundancy:** Redundant data transmission in the network leads to network congestion. To decrease the redundancy, we can make use of data aggregation mechanisms [7].

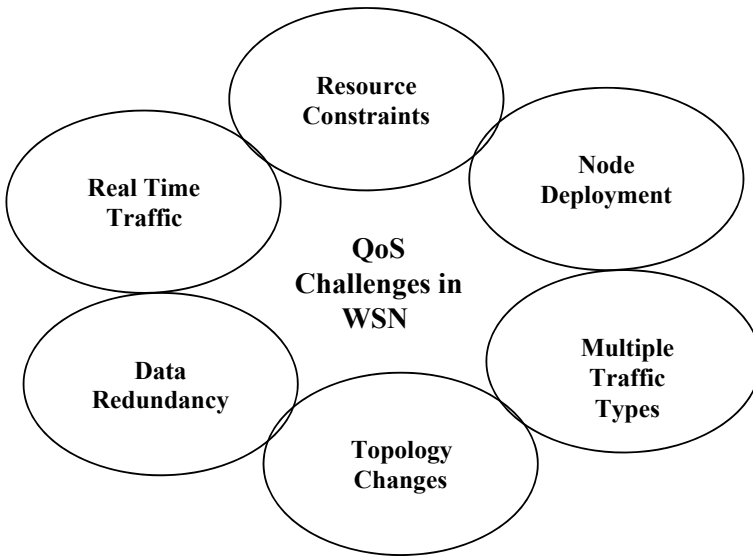


Fig. 2 QoS challenges in WSNs

Real-time traffic: In some critical applications like security inspection or natural disaster monitoring, collected/sensed data should reach the target before its time limit. So, such type of significant data must be handled by sufficient QoS mechanisms.

Critical applications like fire-monitoring, intruder-tracking, and medical care are mostly used in real-time WSNs. All these applications include high needs of bandwidth, delay guarantees and delivery time. When an application forwards data, TCP/IP stack is used to transmit it from the source to the destination nodes. Every layer of TCP/IP stack is associated with some QoS parameters. Figure 3 point up QoS requirements for basic five layers of TCP/IP model such as application layer, transport layer, network layer, medium access control (MAC) layer and physical layer. In delay-protected RT applications in WSN, QoS guarantees can be considered as either providing a deterministic time delay barrier or a probabilistic delay guarantee should be provided. In deterministic approach, data arrived later than its deadline is of no use or it can be considered as collapse of the system whereas in probabilistic approach, some delay is acceptable. Hence, for supporting QoS in such applications WSNs, either required probabilistic or deterministic time delay assurance. Also, based on the research issues, QoS protocols are again classified as soft RT and hard RT protocols. So, choosing the best QoS protocol to obtain better throughput based on the purpose requirements is the most important concern mission-critical applications in WSN.

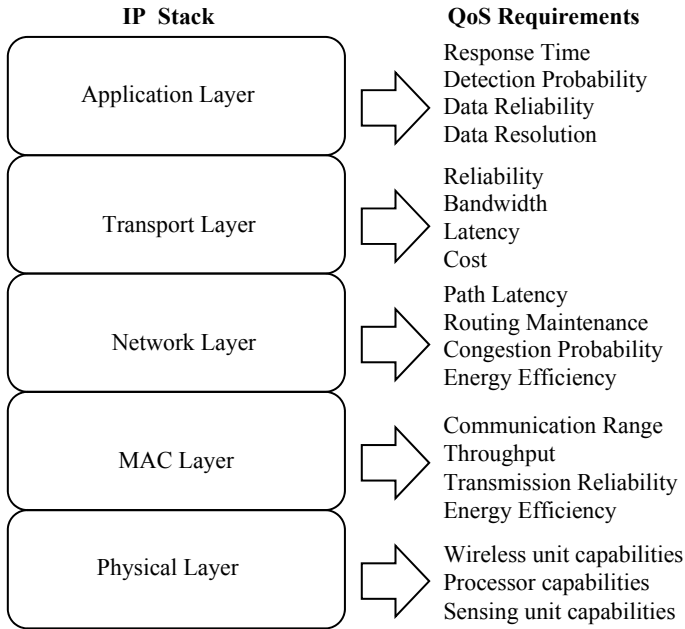


Fig. 3 QoS associated with different layers

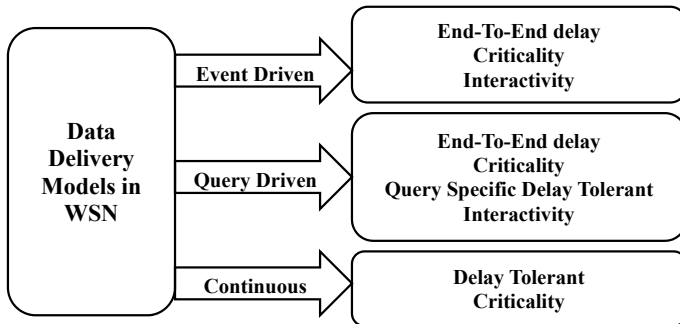


Fig. 4 Data delivery models requirement for various application classes

2.2 Data Delivery Models in WSNs

Various data delivery models used in WSN that are utilized in special applications are given in Fig. 4. Depending on QoS needs in different applications, three data delivery model found in the literature are event driven, query-driven in addition to continuous model [8, 9].

- Event-driven model: For applications like interactive, delay-intolerant, mission-critical, event-driven model is used as these applications require detecting the

events and as a result make the right decisions at the earliest. The accomplishment of these applications based on the efficient event detection and delivery of notification of that event to the end user.

- Query-driven model: This includes applications like query-specific delay lenient, interactive and time-critical applications. Here queries can be forwarded as required so as to minimize the energy usage. This model is alike to that of first model. Only the difference is that in query driven the data is fetched by base station whenever it is required whereas in the event-driven model, data is pressed to the base station when desired event occurs. This data delivery model involves WSN applications like environmental control or habitat monitoring.
- Continuous model: Sensor node sends the gathered data to the base station continuously at a particular time in the continuous model.

Applications requirements in all above three models are different. By making use of these models we can include desired application-specific functionalities in WSNs. The characteristics of the application requirements are depend on the following factors shown in Fig. 4:

- Interactivity: Application can be either interactive or not.
- End-to-end: Application may need end-to-end management or not.
- Delay tolerance: Application may delay accepting or not.
- Criticality: Application may be time-critical or not.

3 Related Works

While expanding RT applications in WSN, resource constraints must be considered in addition to reliability constraints to support desired performance at all time. In recent years, many QoS-aware MAC and routing protocols were proposed for WSNs [8–24]. Many of these protocol works for improving energy efficiency, considering energy of sensor nodes as a significant resource. Besides, some additional challenges also caused by time-critical applications where both energy proficient and QoS supported routing is required in order to utilize the sensors efficiently and accessing the collected data successfully within time bounds. In QoS provisioning, MAC layer should provide delay guarantee for getting access to channel whereas network layer should provide the transmission time barrier. Figure 5 shows the classification of various protocols in view of probabilistic or deterministic approaches.

In WSNs, main role of the MAC layer is to determine delay required for getting access to the channel, channel utilization, and energy requirements. Data transport, reliability, and delay are fundamental performance objectives in mission-critical application scenarios [8]. Delay-aware MAC protocols like Sensor-MAC (S-MAC), Timeout-MAC (T-MAC) Berkeley-MAC (B-MAC) [9–11], etc., only reduce delay to offer best-effort services but RT assurance is not granted. An energy efficient and low latency MAC protocol (D-MAC) [12] and a Delay-Bounded MAC Protocol (DB-MAC) [13] are proposed for application-specific data gathering tree but are bounded

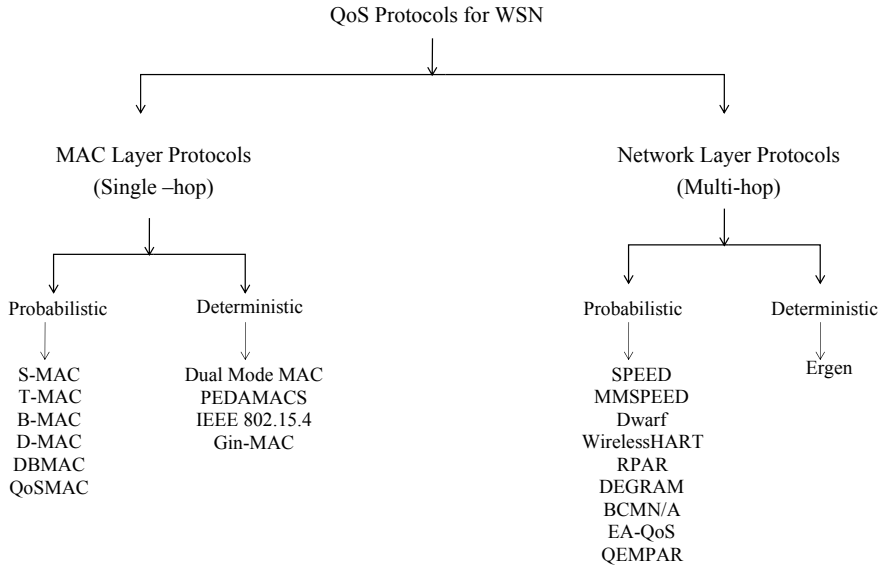


Fig. 5 Classification of QoS protocols for WSN

to use in general topology and even if latency is reduced but precise RT guarantee is not offered by these protocols. In 2006, TDMA-based MAC protocol PEDAMACS [14] is proposed by Ergen, Varaiya to accomplish energy and delay effectiveness. It provides hard RT guarantee but due to limitation of requirement of powerful AP it cannot be used in variety of applications. IEEE 802.15.4 [15] protocol is developed with guaranteed time slot (GTS) mechanism for managing time significant data to provide explicit QoS guarantees and is still developing to support hard RT guarantee.

Even though, MAC layer provides RT guarantee, still without incorporating transmission delay bound in the network layer packet deadline is hard to achieve. Therefore, in real-time applications, routing protocols capable of providing delay assurance are preferential. A Stateless Protocol for Real-Time Communication (SPEED) [16] is a RT routing protocol proposed in 2003 to achieve flexible end-to-end time limit guarantee by supporting a preferred speed in the network. A Multi-Path and Multi-SPEED (MMSPEED) [17] protocol is an improved SPEED, which provides service discrimination and data transmission delay guarantee in addition to data reliability. By providing high link reliability MMSPEED meets both timeliness and reliability requirements over SPEED. However, both in SPEED and MMSPEED energy expenditure of the network was not considered. Some other protocols like Quality of Service MAC (QoSMAC) [18] ensure reliability and transmission delay from source to destination node. Priority-MAC protocol [25] was designed for handling time-critical data transmission for industrialized wireless sensors and actuators network by categorizing the traffic in four different classes depending on priorities considering 1 for highest and 4 for lowest priority and traffic in each of the class utilizes its

committed scheme for medium access. Furthermore, In Dwarf [19] message delay is reduced with increased reliability, whereas in WirelessHART [20], a wireless sensor networking technology based on the Highway Addressable Remote Transducer Protocol only delay from source to destination is promised without guaranteed reliability. In further literature, a novel TDMA-based MAC protocol called Gin-MAC [21] guarantees time delay as well as data delivery reliability in addition to efficient energy utilization.

In 2009, author proposed a routing protocol [22] for WSNs, which enhances the quality of service in real-time data delivery. This protocol improves throughput of the system by reducing the packet deadline miss ratio and efficiency of energy consumption by using energy balancing approach. In [23], author proposes a gateway control protocol (H-NAME) which uses grouping approach to avoid interference between overlapping clusters and to eliminate collisions of hidden nodes in a single or several clusters WSNs by using contention-based MAC protocols. Data aggregation cross-layered design protocol Lump is proposed in 2010, which maintain QoS in various applications by giving priorities to packets for distinguished services and helps aggregation decisions [24]. It reduces overheads in network processing, and thus suitable in many applications.

In 2011, author proposed energy efficient QoS routing protocol (EEQAR) [26] to manage both the energy efficiency and QoS the requirement. EEQAR protocol uses cellular topology scheme for clustering with reasonable energy consumption to gain the enhanced performance. Also, QoS and network lifetime is highly improved using EEQAR for multimedia sensor networks. Similar kind of protocol QoS and Energy-Aware Multi-Path Routing (QEMPAR) [27] is proposed in the same era for providing QoS requirements in RT applications in WSNs. As the protocol utilizes four main parameters of QoS, end-to-end delay is optimized. Here author also highlighted some architectural and other operational challenges for managing the QoS data transmission in sensor networks in addition to enhance energy efficiency of the sensors. Single-hop hybrid cluster architecture is proposed in 2013, which includes two kinds of nodes transmit-only nodes as well as standard nodes [28]. Here author also proposed a framework for MAC layer protocol called RARE with the purpose of managing the one hop hybrid cluster efficiently and reliably in a self-structured fashion in densely deployed area. This work brings reliable scheduling scheme using transmit-only nodes as well as standard nodes. In direction to reduce the sensor nodes required to perform various tasks, energy unconscious protocols may effect in unequal expenditure of sensor energy by passing on irregular workloads to the sensors. In this, heavily loaded sensors may create energy holes, which may result in partitioning the network into incoherent parts. To overcome this trouble and to enhance the network lifetime, in [29] author proposed two energy aware task management protocols: wherein combination centralized and distributed protocols is used for improving network lifetime using load balancing technique amongst sensor nodes. Two stages are used in distributed approach to guess the maximum energy altogether between them. These protocols improve the network lifetime by utilizing the sensor nodes energy on event basis as well as controlling energy gaps between the sensor nodes exist in identical region. Balancing the sensor nodes energy utilization with no

change in effective density of the sensor network is the key benefit of this approach. In [30], Joint Optimization of Lifetime and Transport Delay under Reliability constraint (BCMNA) protocol also improves network lifetime and reduces network delay by using energy as well as delay efficient data aggregation process both in intra- as well as inter-cluster communication. A TDMA-based protocol (DGRAM) that gives significant delay guarantee with desired energy efficiency is proposed in [31]. This protocol is completely self-configuring protocol in which assignment of slots is done with no control messages exchanging. As it does not provide significant delay guarantee, it is not useful in a wide range of applications.

Thus, here some existing MAC and Network layer protocols in Wireless Sensor Networks designed with intension of providing reliable and timeliness data transmission are discussed. In addition, some cross-layer solutions invented for offering QoS in WSNs are also discussed. We elaborated on some of the existing approaches with special focus on the techniques used to boost the network performance of delay-bounded applications considering reliability, delay, energy efficiency, type of service as specific QoS parameter in order to find the technical issues for providing energy efficient, QoS guarantee in WSN. Table 1 summarizes some QoS features provided by various existing data transmission techniques in WSNs.

4 Research Directions

Efficient collection of data from various sensors installed in the field and transmission of the accumulated data to the respective base station is the important responsibilities of the sensor nodes. The major challenge here is to conserve the sensor node energy to maximize their existence, which leads to enhance the network lifetime. It is a most important and considerable factor in efficient data collecting schemes in favor of wireless sensor networks. In addition, QoS is the most important issue in time-critical applications in WSNs. Reliability along with timeliness is the significant parameter to be concern in providing QoS in such applications [32–34]. Data redundancy concept can be used to accomplish Reliability. But, it will affect timeliness factor as it introduces delay in transmission. So, optimum techniques should be launched which will take care of both parameters reliability and timeliness in QoS provisioning. Presently, only some protocols deal with the dual purpose of attaining delay and reliability bounds metrics, but only some of the proposed protocols can preserve time-critical applications. Hence, it is essential to develop event based reliable data transmission methods to support time-critical applications within sensor network domain which gives optimal performance considering QoS guarantee in terms of delay bound and reliability. So, while designing a QoS scheme, researchers must focus on event-based reliable transmission techniques to guarantee the delivery of crucial data in timely as well as reliable manner which gives optimal performance considering delay and reliability bearing in mind the growing needs of various time-critical application domains.

Table 1 Comparison of various QoS protocols in WSN

Protocol name	Type	Reliability	Delay	Energy	RT type	Issues
S-MAC, T-MAC, B-MAC	CSMA/CA- based MAC	Not assured	Decrease node-to- node delay	Duty cycling	Best effort	No real-time data delivery guarantee
D-MAC, DB-MAC	CSMA/CA- based MAC	Not assured	Decrease overall end-to- end delay	Duty cycling	Best effort	No delay guarantee
QoSMAC	TDMA- based MAC + Routing	Yes (Node level)	Yes (Node level)	Duty cycling	Soft RT	No end- to-end delay guarantee
Dual Mode MAC	TDMA- based MAC	Yes (End level)	Yes (End level)	Not con- sidered	Hard RT	Only applicable when load is low
PEDAMACS	TDMA- based MAC	Not assured	Yes (End level)	Duty cycling	Hard RT	Requires high power AP, low scalability
IEEE802.15.4	Slotted CSMA/CA	Yes (End level)	Yes (End level)	Moderate	Best effort/Hard RT	Need to improve reliability
Gin-MAC	TDMA- based MAC	Yes (End level)	Yes (End level)	Duty cycling	Hard RT	Low scalability
Priority-MAC	MAC	Increases (End level)	Decreases (End level)	Yes	Soft RT	No collision avoidance
SPEED	MAC + Routing	Soft (End level)	Soft (End level)	No	Soft RT	Energy metric not consid- ered
MMSPEED	MAC + Routing	Probabilistic (End level)	Probabilistic (End level)	No	Soft RT	Energy metric not consid- ered
Dwarf	MAC + Routing	Increase (End level)	Decrease (End level)	Duty cycling	Soft RT	More focus on reliability than delay parameter

(continued)

Table 1 (continued)

Protocol name	Type	Reliability	Delay	Energy	RT type	Issues
WirelessHART	TDMA/ FDMA MAC + Routing	Increase (End level)	Guarantees (End level)	Duty cycling	Soft RT	More communi- cation overhead
SchedEx	TDMA- based MAC	Guarantees (End level)	Decreases (End level)	Duty cycling	Soft RT	Guaranteeing reliability bounds effects latency bounds
RPAR	Routing	Increase (End level)	Decreases (End level)	Yes	Soft RT	No end- to-end delay & reliability guarantee
DEGRAM	TDMA- based MAC + Routing	No	Guarantees (End level)	Yes	Soft RT	No end to end reliability guarantee
BCMNA	Routing	Overall Network increase	Overall Network delay decreases	Yes	Soft RT	Channel access delay not consid- ered
EA-QoS	Routing	Moderate increase	Guarantees (End level)	Yes	Soft RT	Channel access delay not consid- ered
QEMPAR	Routing	No	Increase in latency	Yes	Soft RT	Throughput affect due to increased latency
Ergen	Routing	No	Guarantees (End level)	Yes	Hard RT	Channel access delay not consid- ered

5 Conclusion

The time-critical application in WSN poses several challenges in concern with timely and reliable delivery of data collected at sensor nodes to the base station. This paper discusses the existing QoS provisioning data transmission techniques in WSNs with different requirements and necessity for different approaches. This paper has discussed the requirements; challenges for sustaining QoS in WSNs. Current developments discussed in the paper indicate that there are still many issues and challenges that need to attend. Considering the various application domains, in addition to energy QoS is also the main concern for successful implementation of WSNs. So, for assured and reliable event awareness in the network is the major requirement of many delay constraint and mission-critical WSN applications. To address this problem, it is necessary to discover most efficient solution in WSNs scheme, which gives reliability, and timeliness in delay sensitive applications, in WSNs. Thus, it looked for research overcome both end-to-end level reliability and delay limitations in real-time systems in WSNs.

References

1. Al-Karaki, J. N., & Kamal, A. E. (2004). Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, 11, 6–28.
2. Li, Y., Chen, C. S., Song, Y.-Q., & Wang, Z. (2007). Real-time QoS support in wireless sensor networks: A survey. In *7th IFAC International Conference on Fieldbuses & Networks in Industrial & Embedded Systems - FeT'2007*, Nov 2007, Toulouse, France.
3. Arampatzis, T., Lygeros, J., & Manesis, S. (2005). A survey of applications of wireless sensors and wireless sensor networks. In *Proceedings of the 20th IEEE International Symposium on Intelligent Control (ISIC 05)* (pp. 719–724), June 2005.
4. Chen, D., & Varshney, P. K. (2004). QoS support in wireless sensor network: A survey. In *Proceedings of International Conference on Wireless Networks (ICWN2004)*, Las Vegas, Nevada, USA, June 2004.
5. Balen, J., Zagar, D., & Martinovic, G. Quality of service in wireless sensor networks: a survey and related patents.
6. Raut, A. R., & Malik, L. G. (2011, May). ZigBee based industrial automation profile for power monitoring systems. *International Journal on Computer Science and Engineering (IJCSE)*, 3(5), 2028–2033 ISSN: 0975-3397.
7. Raut, A. R., & Malik, L. (2011). ZigBee: The emerging technology in building automation. *International Journal on Computer Science and Engineering*, 3(4), 1479–1484.
8. Suriyachai, P., Roedig, U., & Scott, A. (2012, Second Quarter). A survey of MAC protocols for mission-critical applications in wireless sensor networks. *IEEE Communications Surveys & Tutorials*, 14(2).
9. Ye, W., Heidemann, J., & Estrin, D. (2002). An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the 21st Annual Joint Conference IEEE Computer and Communications Societies*, New York, NY, USA (Vol. 3, pp. 1567–1576).
10. Van Dam, T., & Langendoen, K. (2003). An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the 1st ACM Conference on Embedded Networked Sensor Systems*, Los Angeles, CA, USA, 2003 (pp. 171–180).
11. Polastre, J., Hill, J., & Culler, D. (2004). Versatile low power media access for wireless sensor networks. In: *Proceedings of ACM Sensys* (pp. 95–107).

12. Lu, G., Krishnamachari, B., & Raghavendra, C. S. (2004). An adaptive energy efficient and low-latency MAC for data gathering in wireless sensor networks. In *Proceedings of the 18th International Parallel and Distributed Processing Symposium*, Santa Fe, NM, USA, 2004 (pp. 224–231).
13. Bacco, G. D., Melodia, T., & Cuomo, F. (2004). A MAC protocol for delay-bounded applications in wireless sensor networks. In *Proceedings of the MedHoc-Networks 2004* (pp. 208–220).
14. Ergen, S. C., & Varaiya, P. (2006). PEDAMACS: power efficient and delay aware medium access protocol for sensor networks. *IEEE Transactions on Mobile Computing*, 5, 920–930.
15. Yoo, S., Chong, P. K., Doh, Y., Pham, M.-L., Kim, D., Choi, E., & Huh, J. (2010, November). Guaranteeing real-time services for industrial wireless sensor networks with IEEE 802.15.4. *IEEE Transactions On Industrial Electronics*, 57(11), 3868–3876.
16. Hea, T., Stankovica, J. A., Lub, C., & Abdelzahera, T. SPEED: A stateless protocol for real-time communication in sensor networks. In: *Proceedings of the ICDCS* (pp. 46–55).
17. Felemban, E., Lee, C., & Ekici, E. (2006). MMSPEED: Multipath multi SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 5, 738–754.
18. Suriyachai, P., Roedig, U., & Scott, A. (2009). Implementation of a MAC protocol for QoS support in wireless sensor networks. In *Proceedings of the 1st International Workshop Information Quality and Quality of Service for Pervasive Computing, in conjunction with 7th Annual IEEE International Conference on Pervasive Computing and Communications*, Galveston, TX, USA, 2009 (pp. 1–6).
19. Strasser, M., Meier, A., Langendoen, K., & Blum, P. (2007). Dwarf: Delay-aware robust forwarding for energy-constrained wireless sensor networks. In *Proceedings of the 3rd IEEE International Conference on Distributed Computing in Sensor Systems*, Santa Fe, NM, USA, 2007 (pp. 64–81).
20. HART Communication Foundation. WirelessHART technology. [Online]. Available: http://www.hartcomm.org/protocol/wihart/wireless_technology.html, December 2009.
21. Suriyachai, P., Brown, J., & Roedig, U. (2010). Time-critical data delivery in wireless sensor networks. In *Proceedings of the 6th IEEE International Conference on Distributed Computing in Sensor Systems*, Santa Barbara, CA, USA, 2010 (pp. 216–229).
22. Li, Y., & Chen, C. S. (2009, May). Enhancing real-time delivery in wireless sensor networks with two-hop information. *IEEE Transactions On Industrial Informatics*, 5(2), 113–122.
23. Koubâa, A., Severino, R., Alves, M., & Tovar, E. (2009, August). Improving quality-of-service in wireless sensor networks by mitigating “hidden-node collisions”. *IEEE Transactions on Industrial Informatics*, 5(3), 299–313.
24. Jeong, J., & Kim, J. (2010). A QoS-aware data aggregation in wireless sensor networks. In *12th International Conference on Advanced Communication Technology (ICACT)*, February 7–10, 2010.
25. Subramanian, A. K., & Paramasivam, I. (2016). PRIN: A priority-based energy efficient MAC protocol for wireless sensor networks varying the sample inter-arrival time. <https://doi.org/10.1007/s11277-016-3581-5> (Springer Science Business Media, New York).
26. Lin, K., & Rodrigues, J. J. P. C. (2011, December). Energy efficiency QoS assurance routing in wireless multimedia sensor networks. *IEEE Systems Journal*, 5(4).
27. Heikalabad, S., Rasouli, H., Nematy, F., & Rahmani, N. (2011, January). QEMPAR: QoS and energy aware multi-path routing algorithm for real-time applications in wireless sensor networks. *IJCSI International Journal of Computer Science Issues*, 8(1).
28. Zhao, J., Qiao, C., & Sudhaakar, R. S., & Yoon, S. (2013, March). Improve efficiency and reliability in single-hop WSNs with transmit-only nodes. *IEEE Transactions on Parallel and Distributed Systems*, 24(3).
29. AbdelSalam, H. S., & Olariu, S. (2011, November). Toward efficient task management in wireless sensor networks. *IEEE Transactions on Computers*, 60(11).
30. Dong, M., & Ota, K. Joint optimization of lifetime and transport delay under reliability constraint wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, TPDS-2013-12-1250.

31. Shanti, C., & Sahoo, A. (2010, October). DGRAM: A delay guaranteed routing and AC protocol for wireless sensor networks. *IEEE Transactions on Mobile Computing*, 9(10).
32. Pöttner, W.-B., & Seidel, H. (2016, January). Constructing schedules for time-critical data delivery in wireless sensor networks. *ACM Transactions on Sensor Networks*, V(N), Article A.
33. Mohammad, B. M., Abd Kader, S., & Konber, H. A. Designing a channel access mechanism for wireless sensor network. *Wireless Communications and Mobile Computing*, 2017, Article ID 7493269, 31 p. <https://doi.org/10.1155/2017/7493269>.
34. Richert, V., & NaumanIsrar, B. Implementation of a modified wireless sensor network MAC protocol for critical environments. *Wireless Communications and Mobile Computing*, 2017, Article ID 2801204, 23 p. <https://doi.org/10.1155/2017/2801204>.

Impact of Shuffler Design Pattern on Software Quality



G. Priyalakshmi, R. Nadarajan, Joseph W. Yoder, S. Arthi and G. Jayashree

Abstract Modern software has become intricate and versatile due to the world-wide growth of new software technologies. In this regard, the evolution of software quality metrics to support software maintainability is studied. The impact of the Shuffler design pattern on software quality metrics is evaluated in this paper. The Shuffler design pattern provides an efficient design approach for shuffling. The pattern helps to choose generic shuffling alternatives that make the client program loosely coupled, and thus attaining high reusability. A few software quality metrics, which has a higher influence on software reusability and maintainability, are experimented on three gaming applications like Jigsaw, Poker, and Scramble. These gaming applications are redesigned using Shuffler design pattern and a combination of other patterns. The three software quality metrics, which show improvement on the redesigned applications, are McCabe Cyclomatic Complexity, Lack of Cohesion of Methods and Specialization Index. The authors also have tested the pattern with a reusability metrics suite, which measures the reusability of the black-box components of the aforementioned applications without any source code. The results with high cohesive and low coupling values would help software designers in the industry to be more confident in using the Shuffler pattern along with other design patterns. The interdependence of the three software metrics on the software quality attributes is finally tabulated to show their impact on software quality.

1 Introduction

Design patterns represent solutions that have emerged and matured over time. Patterns solve particular design problems and make object-oriented designs more flexible, refined, and ultimately reusable. They assist designers to reuse successful designs

G. Priyalakshmi (✉) · R. Nadarajan · S. Arthi · G. Jayashree
Department of Applied Mathematics and Computational Sciences, PSG College of Technology,
Coimbatore, India
e-mail: priya.venky2001@gmail.com

J. W. Yoder
The Refactory, Inc, Urbana, USA

© Springer Nature Singapore Pte Ltd. 2019
N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_31

by depending upon new designs on prior experience. A software designer who is acquainted with such patterns can put them into effect to design problems directly without ascertaining them [1]. According to ISO 9126, software must be portable, completely functional, reliable, usable, efficient, and maintainable. Each quality sub-attribute (low-level quality attributes, such as complexity, cohesion, etc.), can be assessed by a set of metrics that can be used as indicators for the score of a system with respect to the corresponding high-level quality attribute.

Apostolos, Sofia, and Ioannis have presented an extensive survey of 120 papers to impart an overview of GOF pattern research [2]. They have accounted that, the most prominent research field is the impact of patterns on software quality attributes. Finally, they have formally stated that the reported research on the effect of patterns on software quality attributes is controversial, since some studies identify one pattern's effect as beneficial whereas others report the same pattern's effect as harmful. Hence, this work attempts to narrow these gaps by investigating the effect of applying patterns on software quality attributes and providing an empirical study.

Shuffler design pattern creates a design technique from existing solutions that make them available to software developers [3]. Shuffle is a method that refers to jumbling or interchanging the positions of objects. Online games like Jigsaw puzzle, Sudoku, Poker, Jumbled word games, etc., more commonly use the shuffling design solution. In music applications, shuffling refers to the capability to randomize the order of the songs in a playlist. Thus, shuffling has an immense number of well-known uses in the software industry. The other real-time practical applications are shuffling questions and choices in quiz apps or online tests, image shuffling in games, shuffling wallpapers in desktops or mobiles, shuffling characters in password generators, etc.

The research work proposed by the authors is to measure quantitatively the impact of Shuffler design pattern on open source gaming applications. An Eclipse plug-in called *Metrics* calculates 22 metrics for these applications. Out of 22 metrics, three of them showed a significant improvement, when the Shuffler design pattern was applied. Table 3 provides a brief description of a few of the metrics. In addition, few coupling metrics listed in Table 4, were used to study the impact of Shuffler pattern on the software quality of the above mentioned open source systems. A comparative analysis of the open source applications for each of the significant metrics yields better results.

To emphasize the proposed work, the authors also consider reusability metrics for software components. The metrics studied were EMI (Existence of Meta-Information), RCO (Rate of Component Observability), RCC (Rate of Component Customizability), SCCr (Self-Completeness of Component's Return Value), SCCp (Self-Completeness of Component's Parameter), for computing the existence of meta-information, observability, customizability, and external dependency of software components without source code [4]. These metrics were applied to the applications Poker, Jigsaw, and Scramble. The Poker gaming application after redesigning using Shuffler design pattern had greater flexibility in the metrics EMI and SCCp. When applied to Scramble game, it showed similar values for all the metrics except EMI, indicating minimal impact of Shuffler design pattern on this application. When the Jigsaw system was measured with the same metrics, except for the RCC and

SCCp, all other metrics showed improved values when compared to the same module designed without Shuffler design pattern.

A decision table, showing the interdependence between software quality measures and object-oriented metrics, coupling metrics, and reusability metrics for components, was reconstructed. This could be very effective for software practitioners to interpret the direct impact of design patterns on software quality. Finally, the study was extended with the impact of Shuffler design pattern coupled with Singleton [5] and a similar analysis was done on Poker and Scramble projects. The effect of Shuffler and Prototype coupled was measured on the Scramble project, and the results were tabulated.

2 Related Work

There has been a booming involvement in the adoption of design patterns, since their initiation in 1995 by Gamma et al. [1]. In this section, some lines of work which performs a study of the impact of design patterns on quality attributes are presented. Lange and Nakamura proved [6] that patterns help in inspection of a software program thus increasing its understandability. However, this study was limited to a single quality attribute and to a few subset of patterns. Foutse and Yann hypothesized [7] that though object-oriented best practices help producing systems with good quality; it is surprising that their adoption seems to decrease quality. The negative evaluations may be just an a priori on the pattern because their respondents considered the pattern not suitable. Also, their evaluation may not reflect the real impact of the implementation of the pattern on quality. In addition, they inferred that the consequences for some design patterns cannot be justified, thus calling for further studies on the impact of these principles on quality.

Bansiya and Davis [8] evaluated high-level quality features in designs of object-oriented software systems. The hierarchical QMOOD model, which the authors proposed, relates various characteristics like cohesion, coupling, encapsulation, modularity, etc. to high-level quality attributes using experimental and informal information. The authors have examined qualities like understandability, reusability, flexibility, etc. The model also provides a unique feature to include different relationships and weights. Hence, it delivers a practical quality assessment tool compliant to a variety of needs. Wendorff [9] warned developers of commercial products that the magnitude of problems in maintenance phase is proportional to the unconstrained use of patterns. The intention of this paper is not to condemn the idea of patterns; instead, it is motivated by the remark that the improper application of patterns can possibly go wrong. The paper identified two categories of inappropriately applied patterns: first, patterns that were simply misused by software developers who had not understood the rationale behind the patterns; second, patterns that do not match the project's requirements. There are many more studies which include deep learning and theoretical evaluation of the design patterns. This led to the hypothesis of vari-

Table 1 Survey on existing work with their limitations

References	Authors	Content	Limitations
[1]	Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides	Introduction to object-oriented design. Applications, advantages, disadvantages of design patterns	
[7]	Foutse Khomh, Yann-Gael Gueheneuc	Object-oriented practices help producing systems with good quality	The outcome of their research could not be justified for some design patterns
[6]	D. B. Lange, Y. Nakamura	Demonstration of design patterns that serve as a guide in program exploration	Limited to a single quality attribute. Limited to certain patterns
[8]	Jagdish Bansiya, Carl G. Davis	Proposed a hierarchical QMOOD model to assess design with abstract quality characteristics	
[9]	Peter Wendorff	In many commercial softwares, the increase in maintenance issues is a result of uncontrolled use of patterns	
[14]	Ronald Jabangwe, Jürgen Börstler, Darja Šmite, Claes Wohlin	Analysis of the association between design measures and quality attributes not related to source code was done and the stability of the relationship was evaluated across other studies	Can expand types of external attributes that were explored
[2]	Apostolos Ampatzoglou, Sofia Charalampidou, Ioannis Stamelos	A survey of existing work in the field of GoF patterns is done by a mapping review of 120 basic researches	Investigation of the impact of patterns on design quality using other empirical methods or theoretical methods

ous models and theories for assessing the design patterns and analyzing their impact. Table 1 shows a consolidated list of few studies and their limitations.

Object-oriented environments are evaluated using various metrics. These metrics are developed as user-friendly tools in various programming languages like Object

Pascal, Java and C++. A survey of few of these tools and the metrics evaluated by them are provided in Table 2.

3 Object-Oriented Metrics

The object-oriented metrics are used to measure, foresee and enhance the quality of software products. There are various metrics that are used in different phases of project management, but this paper focuses on object-oriented design metrics. Object-oriented design metrics are classified by Archer into coupling level, inheritance level, class level, and method level. This study focuses on the object-oriented metrics tabulated in Table 3.

McCabe Cyclomatic Complexity counts the number of flows through a piece of code. Each time a branch occurs (if, for, while, do, case, catch, ?: ternary operator, as well as the && and || conditional logic operators in expressions) this metric is incremented by one. It is calculated for methods only. If a method has more than 10 different loops, break the method to reduce McCabe Cyclomatic Complexity. For a project, McCabe Cyclomatic Complexity must be less. The complexity M is defined as

$$M = E - N + 2P \quad (1)$$

where

E the number of edges in the graph.

N the number of nodes in the graph.

P the number of connected components

Lack of Cohesion of methods (LCOM) is a measure for the cohesiveness of a class. It is calculated using Henderson-Sellers method. If $m(A)$ is the number of functions retrieving an attribute A , compute the average of $m(A)$ for all attributes. From the mean, subtract the number of functions, m and divide the resultant expression by $(1 - m)$. A small value signifies a cohesive class and a value almost nearer to 1 implies a lack of cohesion. A high LCOM value means low cohesion [10]. A class with high cohesion might show a better design if split into a number of subclasses [11]. Take each pair of methods in the class. If they access disjoint sets of instance variables, increase P by one. If they share at least one variable access, increase Q by one. LCOM is computed as

$$\text{LCOM} = P - Q \quad (\text{if } P > Q) \quad (2)$$

$$\text{LCOM} = 0 \quad (\text{otherwise}) \quad (3)$$

Table 2 Software tools with the metrics they measure

S. No.	Tools	Metrics measured
1	Eclipse metrics plug-in [15]	Cyclomatic complexity
		Coupling
		Specialization index
		Depth of inheritance tree
		Abstractness
		Instability
2	Analyst4j [16]	Normalized distance
		Cyclomatic complexity
3	CCCC [17]	Halstead Effort
		McCabe’s Complexity
		Chidamber & Kemerer metrics
4	Chidamber & Kemerer Java metrics [18]	Henry & Kafura metrics
		Afferent coupling
		Number of public methods
		Depth of inheritance tree
5	Dependency Finder [19]	Response for a class
		Static classes per group
		Single lines of code
6	Testwell CMT Java [20]	Inner classes per group
		McCabe’s Cyclomatic number
		Depth of control structure nesting
		Maintainability index
7	Resource standard metrics [21]	Halstead metrics
		Cyclomatic complexity
		Overall complexity
8	CodePro AnalytiX [22]	LOC/SLOC
		Comments ratio
		Efferent coupling
		Block depth
9	Java Source Code Metrics [23]	Cyclomatic complexity
		Depth of Inheritance tree
10	JDepend [24]	Coupling
		Abstractness
		Instability

(continued)

Table 2 (continued)

S. No.	Tools	Metrics measured
		Package dependency cycle
11	JHawk [25]	Average cyclomatic complexity
		Maintainability index
		Cumulative Halstead effort
		Cumulative Halstead Bugs
12	jMetra [26]	Lines of code
		Method tallies
13	JMetric [27]	OO metrics
14	Krakatau Metrics [28]	Line counting metrics
		Complexity metrics
		Halstead metrics
15	Refactorit [29]	Weighted method per class
		Lines of code
		Depth of Inheritance tree
16	SonarJ [30]	Cyclic dependencies
17	OOMeter [31]	Ratio of cohesive inheritance
		Coupling between objects
		Tight class cohesion
		Loose class cohesion
18	SemmlCode [32]	Number of lines of code
19	Understand [33]	Average complexity
		Average line code
		Average line blank
20	VizzAnalyzer [34]	Class complexity

LCOM = 0 indicates a cohesive class. LCOM > 0 indicates that the class needs or can be split into two or more classes since its variables belong to disjoint sets. Classes with a high LCOM are fault-prone.

The specialization index is defined as in Eq. 4. This is a class level metric.

$$\text{Specialization Index} = (\text{NORM} * \text{DIT}) / \text{NOM} \tag{4}$$

To maintain high-quality software, developers need to aim for a low-coupled and highly cohesive design. The coupling and cohesion metrics are computed by dealing with a number of associations proposed by various authors doing research. Coupling metrics are influenced by acyclic dependencies, stable dependencies, and stable abstractions. The authors have also considered the impact of Shuffler pattern

Table 3 Object-oriented metrics with their interpretations

Object-oriented metrics	Interpretation
Number of children	Total number of direct subclasses of a class
Number of interfaces	Total number of interfaces in the selected scope
Depth of Inheritance Tree (DIT)	Distance from class Object in the inheritance hierarchy
Number of Overridden Methods (NORM)	Total number of methods in the selected scope that are overridden from an ancestor class
Number of Methods (NOM)	Total number of methods defined in the selected scope
Number of fields	Total number of fields defined in the selected scope
Lines of code	TLOC: Total lines of code that will count non-blank and non-comment lines in a compilation unit
	MLOC: Method lines of code will count and sum non-blank and non-comment lines inside method bodies
Specialization index	It is defined as $NORM * DIT / NOM$. This is a class level metric
McCabe Cyclomatic Complexity	Counts the number of flows through a piece of code. Calculated for methods only
Weighted Methods per Class (WMC)	Sum of the McCabe Cyclomatic Complexity for all methods in a class
Lack of Cohesion of Methods (LCOM)	A measure for the cohesiveness of a class

Table 4 Coupling metrics with their meaning

Coupling metrics	Interpretation
Afferent coupling (C_a)	The number of classes outside a package that depends on classes inside the package
Efferent coupling (C_e)	The number of classes inside a package that depends on classes outside the package
Instability (I)	$C_e / (C_a + C_e)$
Abstractness (A)	The number of abstract classes and interfaces, divided by the total number of types in a package
Normalized distance from main sequence (D_n)	$ A + I - 1 $, this number should be small, close to zero for good packaging design

on coupling metrics. The coupling metrics which are used in this work are elaborated in Table 4.

4 Reusability Metrics for Components

Reusability is one of the key features of well-designed object-oriented software. In order to evaluate the software reusability, Component-based Development (CBD) is carried out. CBD is a software development method to state, execute and constitute loosely coupled individualistic components into software. CBD is characterized by two activities: development of components for reuse, and development of software systems with reuse of components [12]. As discussed, the various reusability metrics include EMI, RCO, RCC, SCCr, and SCCp. For each of the gaming applications redesigned using the Shuffler design pattern, the components are evaluated for reusability based on the above-mentioned metrics as shown in Figs. 4, 5 and 6. It can be inferred from the analysis that there is a positive impact on reusability of the redesigned software components.

Reusability metrics such as EMI, RCO, RCC, SCCr, and SCCp [4] are applied on the above mentioned gaming software Poker, Scramble, and Jigsaw. EMI measures how much the users of the target component can understand its usage. RCO indicates the component's degree of observability and understandability. RCC indicates the component's degree of customizability and adaptability. Thus, RCO is the ratio of readable fields to all the fields declared and defined within the target class in the component. This metric signifies that a large value of readability would assist the user to comprehend the working of a component from an external environment. Similarly, RCC is the ratio of writable fields to all the fields declared and defined within the target class in the component. High value of RCC metric stipulates higher customizability of the component as per the demand of the user, thus causing inflated adaptability. SCCr is the ratio of business functions with no return to all the business functions defined within a component. SCCp is the ratio of business functions with no parameters to all the business functions defined within a component.

5 Results

This section of the paper presents the results of the assessment, according to two perspectives. The first section presents the analysis of metrics on applications with only Shuffler pattern. The second part provides the research outcomes with applications redesigned with both Shuffler pattern and the related patterns, Singleton and Prototype.

5.1 Analysis of Metrics—Shuffler Pattern

The games such as Poker, Jigsaw puzzle, and Scramble were redesigned and implemented with Shuffler design pattern. The quality of these software projects is assessed

using an Eclipse Plug-in for Object-Oriented Metrics, named Metrics. The impact of Shuffler Design Pattern on the above-mentioned projects are recorded below.

Poker is a gambling card game in which the dealer distributes the shuffled cards to the players. This game is designed using object-oriented paradigm and implemented in Java. The same game is rejuvenated using Shuffler Design Pattern and analyzed using Metrics. The assessment of various object-oriented metrics for the project before and after Shuffler Design Pattern are shown in Table 5. The improved metrics are highlighted.

Jigsaw Puzzle is a tiling puzzle game that displays shuffled pieces of an image. Scramble is a word game that jumbles the letters of a word. The results of similar metric analysis on Jigsaw Puzzle and Scramble are tabulated in Tables 6 and 7, respectively.

Table 5 Object-oriented metrics for Poker

Metrics	Original Design			Redesign		
	Total	Mean	Std. Dev.	Total	Mean	Std. Dev.
Number of overridden methods	8	1	0.707	9	0.9	0.7
Number of attributes	26	3.25	4.918	26	2.6	4.587
Number of children	0	0	0	1	0.1	0.4
Number of classes	8	8	0	10	10	0
Method lines of code	326	5.094	8.148	331	4.868	7.959
Number of methods	64	8	3.969	68	6.8	4.285
Nested block depth		1.484	0.77		1.456	0.756
Depth of inheritance tree		1.75	0.968		1.7	0.9
Afferent coupling		0	0		0	0
Number of interfaces	0	0	0	0	0	0
McCabe Cyclomatic Complexity		2.125	2.684		2.059	2.617
Total lines of code	506			524		
Instability		1	0		1	0
Number of parameters		0.156	0.404		0.206	0.439
Lack of cohesion of methods		0.292	0.321		0.234	0.31
Efferent coupling		3	0		3	0
Number of static methods	0	0	0	0	0	0
Normalized distance		0	0		0	0
Abstractness		0	0		0	0
Specialization index		0.258	0.324		0.306	0.378
Weighted methods per class	136	17	17.183	140	14	16.498
Number of static attributes	0	0	0	0	0	0

Table 6 Object-oriented metrics for Jigsaw puzzle

Metrics	Original Design			Redesign		
	Total	Mean	Std. Dev.	Total	Mean	Std. Dev.
Number of overridden methods	0	0	0	2	0.4	0.8
Number of attributes	11	3.667	4.497	11	2.2	3.919
Number of children	0	0	0	1	0.2	0.4
Number of classes	3	3	0	5	5	0
Method lines of code	125	7.812	13.749	130	6.5	12.808
Number of methods	14	4.667	1.247	18	3.6	1.625
Nested block depth		1.438	0.788		1.35	0.726
Depth of inheritance tree		4.667	1.886		3.4	2.154
Afferent coupling		0	0		0	0
Number of interfaces	0	0	0	0	0	0
McCabe Cyclomatic Complexity		1.688	1.685		1.55	1.532
Total lines of code	232			251		
Instability		1	0		1	0
Number of parameters		0.75	1.09		0.8	0.98
Lack of cohesion of methods		0.267	0.377		0.16	0.32
Efferent coupling		0	0		0	0
Number of static methods	2	0.667	0.943	2	0.4	0.8
Normalized distance		0	0		0	0
Abstractness		0	0		0	0
Specialization index		0	0		0.4	0.8
Weighted methods per class	27	9	4.32	31	6.2	4.792
Number of static attributes	0	0	0	0	0	0

The NOM metric in each class of redesigned Scramble project is depicted in Table 8 for clarification. The total number of methods in that project is 6 since there are 2 methods in WordShuffler.java file, 2 methods in Word.java file and 2 in Scramble Shuffler.java. The number of files and the number of methods in that project are 4 and 6, respectively. The average number of methods in the package ScrambleWithPattern is 6/4, which is 1.5. The standard deviation is 0.886 from the mean.

The three highlighted metrics from the above tables has been plotted in the bar graphs. Figures 1, 2 and 3 represents the changes in McCabe Cyclomatic Complexity, Lack of Cohesion of methods and Specialization Index, respectively, for all the above-mentioned projects.

According to the CBD model [4], the measurement values of all five metrics are always normalized to a number between 0 and 1. The confidence co-efficient of the confidence interval of each metric is 95%. The metric quality is believed to be

Table 7 Object-oriented metrics for Scramble

Metrics	Original Design			Redesign		
	Total	Mean	Std. Dev.	Total	Mean	Std. Dev.
Number of overridden methods	0	0	0	2	0.5	0.866
Number of attributes	1	0.5	0.5	1	0.25	0.433
Number of children	0	0	0	1	0.25	0.433
Number of classes	2	2	0	4	4	0
Method lines of code	12	4	2.16	17	2.429	2.129
Number of methods	2	1	1	6	1.5	0.866
Nested block depth		1.333	0.471		1.143	0.35
Depth of inheritance tree		1	0		1.25	0.433
Afferent coupling		0	0		0	0
Number of interfaces	0	0	0	0	0	0
McCabe Cyclomatic Complexity		1.333	0.471		1.143	0.35
Total lines of code	35			53		
Instability		1	0		1	0
Number of parameters		0.667	0.471		0.857	0.35
Lack of cohesion of methods		0	0		0	0
Efferent coupling		0	0		0	0
Number of static methods	1	0.5	0.5	1	0.25	0.433
Normalized distance		0	0		0	0
Abstractness		0	0		0	0
Specialization index		0	0		0.5	0.866
Weighted methods per class	4	2	0	8	2	0
Number of static attributes	0	0	0	0	0	0

enormous, if the metric value M is in a confidence interval $[LLM, ULM]$, where LLM is the lower confidence limit and ULM is the upper confidence limit.

Figure 4 shows the measurement of component-based metrics for Poker module. Out of the five metrics, EMI and SCCp are within the 95% confidence interval, thus the quality factor corresponding to these metrics is suitably high. The remaining metrics RCO, RCC and SCCr illustrate lesser measures in their corresponding quality factors.

The five metrics applied to the Scramble module redesigned using Shuffler design pattern are shown in Fig. 5. The Scramble software shows higher values only in the metric EMI. The quality factors corresponding to RCO, RCC, SCCr and SCCp metrics show lowered measures.

The measurement of metrics for Jigsaw module is shown in Fig. 6. Out of the five metrics, EMI, RCO and SCCr are within the 95% confidence interval, thus the

Table 8 NOM metric for scramble

Number of Methods (avg/max per type)	6	1.5	0.866	2	/ScrambleWithPattern/src/ScrambleWithPattern/WordShuffler.java
src	6	1.5	0.866	2	/ScrambleWithPattern/src/ScrambleWithPattern/WordShuffler.java
ScrambleWithPattern	6	1.5	0.866	2	/ScrambleWithPatternAro/ScrambleWithPattern/WordShuffler.java
WordShuffler.java	2	2	0	2	/ScrambleWithPattern/src/ScrambleWithPatternMordShuffler.java
Word Shuffler	2				
Word.java	2	2	0	2	/ScrambleWithPattern/src/ScrambleWithPattern/Word.java
Word	2				
SorambleShuffler.java	2	2	0	2	/ScrambleWithPattern/5ro/ScrambleWithPattern/ScrambleShuffler.java
ScrambleShuffler	2				
WordPrintjava	0	0	0	0	/ScrambleWithPattern/src/ScrambleWithPattern/WordPrint.java
WordPrint	0				

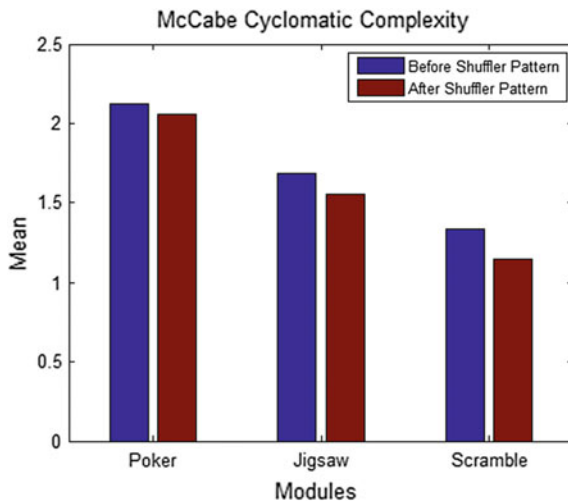


Fig. 1 Comparison of McCabe Cyclomatic complexity metric before and after the Shuffler pattern

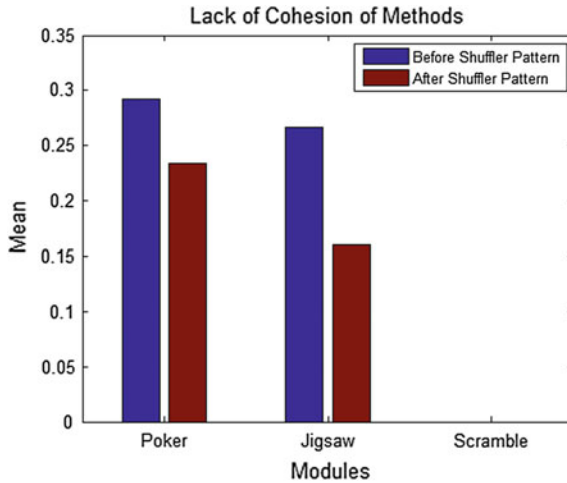


Fig. 2 Comparison of Lack of Cohesion of Methods Metric before and after the Shuffler pattern

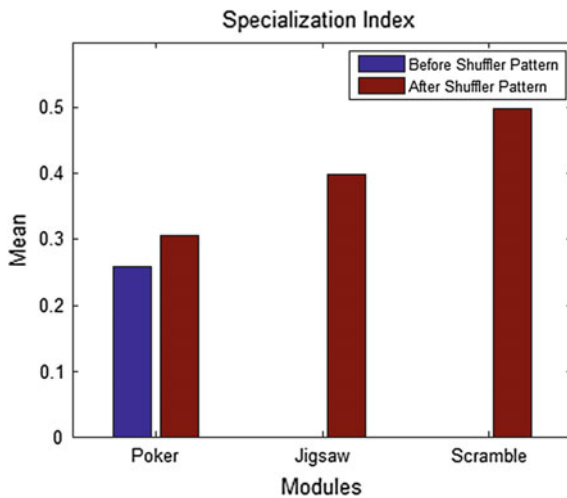


Fig. 3 Comparison of specialization index metric before and after the Shuffler pattern

quality factor corresponding to these metrics is suitably high. The remaining metrics RCC and SCCp illustrate lesser measures in their corresponding quality factors.

In order to obtain a clear understanding of how various object-oriented metrics impact the software quality, various software quality measures are assessed against the object-oriented metrics and are tabulated in Table 9. The impact of McCabe Cyclomatic Complexity, Lack of Cohesion of Methods, Specialization Index on various software quality attributes are shown in the table. All the metrics have an impact on maintainability, understandability, and reliability of the software. Lack of Cohe-

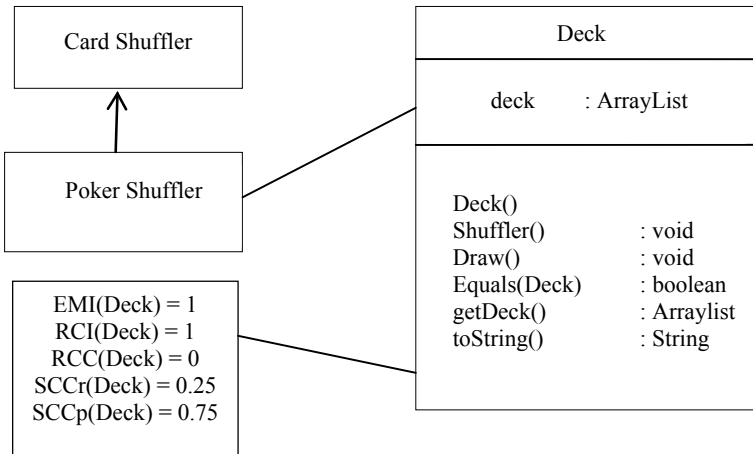


Fig. 4 Component based metrics for Poker

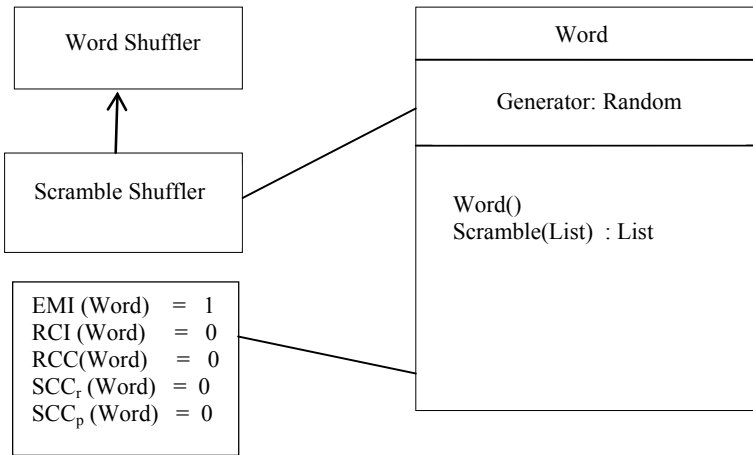


Fig. 5 Component based metrics for scramble

sion of Methods impacts most attributes. The table depicting the fourteen software quality attributes and three design metrics is tabulated to infer the dependency of high-level quality attributes and low-level design metrics. The decision table entries for Cyclomatic Complexity are completed from the observations in the hierarchical model [8]. Similarly, from the comparisons and analyses in various research studies [4, 6, 8, 13], the other table entries are filled. It is evident from the table that the assessed object-oriented metrics do have a positive impact on the reusability of the software and other software quality attributes.

The paper interprets that McCabe Complexity influences software quality 42% and Specialization index has 50% impact on quality. The Lack of Cohesion of Meth-

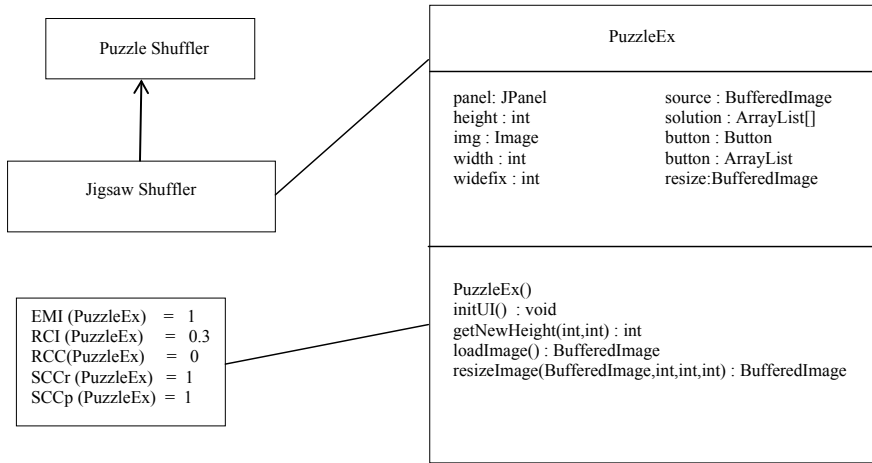


Fig. 6 Component based metrics for Jigsaw

Table 9 Decision table showing the interdependence between software quality measures and object-oriented metrics

	McCabe Cyclomatic Complexity	Lack of cohesion of methods	Specialization index
Expandability	No	Yes	No
Reusability	No	Yes	Yes
Flexibility	Yes	Yes	No
Modularity	Yes	Yes	No
Generality	No	No	No
Scalability	No	No	No
Robustness	No	No	No
Stability	No	Yes	Yes
Adaptability	No	No	Yes
Maintainability	Yes	Yes	Yes
Reliability	Yes	Yes	Yes
Simplicity	Yes	No	No
Learnability	No	Yes	Yes
Understandability	Yes	Yes	Yes

ods has the maximum hit of 64%. On the other hand, the study concludes that Maintainability, Reliability, and Understandability are the qualities, which are highly influenced by the Shuffler design pattern. The other quality features may or may not show improvement with Shuffler pattern.

5.2 Analysis of Metrics—Shuffler Combined with Related Patterns

A combination of various related design patterns were applied to the existing projects. Along with the Shuffler design pattern, Singleton pattern was applied to Jigsaw project and the impact was measured by the Eclipse plug-in *Metrics* and the results are provided in Table 10. There was no significant impact of the combined patterns when compared to that of the project implementation with only Shuffler pattern. The Shuffler pattern was combined with Singleton pattern and the results are tabulated in Table 11. The results were similar to the results in Table 7 with no significant improvement in any of the metrics. It was also observed that it had negative impact when the Scramble project was implemented with Prototype design pattern. This impact is evident from the highlighted metrics in Table 12.

6 Conclusion

The most vital state of the art research on design patterns is; the impact of design patterns on quality features of a software. This paper aims at emphasizing the effect of pattern application on quality attributes. The first research effort made by the authors in the direction of design patterns was the identification of Shuffler design pattern and more importantly its known uses. After a thorough literature survey on papers related to GOF design patterns, the impact of Shuffler pattern on three open source projects was studied. With this study on the impact of Shuffler design pattern on the software quality metrics tested for gaming applications, it has been analyzed that the software quality metrics such as McCabe Cyclomatic Complexity, Lack of Cohesion of Methods and Specialization Index are improved. The software quality attributes that got affected by the pattern were listed in the Results section.

In addition to the conventional metrics, this study also focuses on reusability metric to measure the reuse of the black-box components of the three gaming applications. The advantage of this analysis is, it would be helpful to designers when the source code of components is not available. The CBD analysis carried out for the projects supported to prove that the use of Shuffler pattern increases software quality.

The paper also relates high-level software quality measures to the three low-level object-oriented metrics, McCabe Cyclomatic Complexity, Lack of Cohesion of Methods and Specialization Index. The study finally measures the software quality

Table 10 Object-oriented metrics for Jigsaw with Singleton

Metrics	With Shuffler pattern only			With Shuffler and Singleton pattern		
	Total	Mean	Std. Dev.	Total	Mean	Std. Dev.
Number of overridden methods	2	0.4	0.8	2	0.4	0.8
Number of attributes	11	2.2	3.919	11	2.2	3.919
Number of children	1	0.2	0.4	1	0.2	0.4
Number of classes	5	5	0	5		
Method lines of code	130	6.5	12.808	130	6.5	12.808
Number of methods	18	3.6	1.625	18	3.6	1.625
Nested block depth		1.35	0.726		1.35	0.726
Depth of inheritance tree		3.4	2.154		3.4	2.154
Afferent coupling		0	0		0	0
Number of interfaces	0	0	0	0	0	0
McCabe cyclomatic complexity		1.55	1.532		1.55	1.532
Total lines of code	251			251		
Instability		1	0		1	0
Number of parameters		0.8	0.98		0.8	0.98
Lack of cohesion of methods		0.16	0.32		0.16	0.32
Efferent coupling		0	0		0	0
Number of static methods	2	0.4	0.8	2	0.4	0.8
Normalized distance		0	0		0	0
Abstractness		0	0		0	0
Specialization index		0.4	0.8		0.4	0.8
Weighted methods per class	31	6.2	4.792	31	6.2	4.792
Number of static attributes	0	0	0	0	0	0

of the gaming applications with Shuffler and related patterns like Singleton and Prototype.

Table 11 Object-oriented metrics for Scramble with Singleton

Metrics	With Shuffler pattern only			With Shuffler and Singleton pattern		
	Total	Mean	Std. Dev.	Total	Mean	Std. Dev.
Number of overridden methods	2	0.5	0.866	2	0.5	0.866
Number of attributes	1	0.25	0.433	1	0.25	0.433
Number of children	1	0.25	0.433	1	0.25	0.433
Number of classes	4	4	0	4		0
Method lines of code	17	2.429	2.129	17	2.429	2.129
Number of methods	6	1.5	0.866	6	1.5	0.866
Nested block depth		1.143	0.35		1.143	0.35
Depth of inheritance tree		1.25	0.433		1.25	0.433
Afferent coupling		0	0		0	0
Number of interfaces	0	0	0	0	0	0
McCabe Cyclomatic Complexity		1.143	0.35		1.143	0.35
Total lines of code	53			53		
Instability		1	0		1	0
Number of parameters		0.857	0.35		0.857	0.35
Lack of cohesion of methods		0	0		0	0
Efferent coupling		0	0		0	0
Number of static methods	1	0.25	0.433	1	0.25	0.433
Normalized distance		0	0		0	0
Abstractness		0	0		0	0
Specialization index		0.5	0.866		0.5	0.866
Weighted methods per class	8	2	0	8	2	0
Number of static attributes	0	0	0	0	0	0

Table 12 Object-oriented metrics for Scramble with Prototype

Metrics	With Shuffler pattern only			With Shuffler and Prototype pattern		
	Total	Mean	Std. Dev.	Total	Mean	Std. Dev.
Number of overridden methods	2	0.5	0.866	3	0.5	0.764
Number of attributes	1	0.25	0.433	2	0.333	0.471
Number of children	1	0.25	0.433	2	0.333	0.471
Number of classes	4	4	0	6	6	0
Method lines of code	17	2.429	2.129	40	4	7.085
Number of methods	6	1.5	0.866	9	1.5	0.764
Nested block depth		1.143	0.35		1.3	0.9
Depth of inheritance tree		1.25	0.433		1.333	0.471
Afferent coupling		0	0		0	0
Number of interfaces	0	0	0	0	0	0
McCabe Cyclomatic Complexity		1.143	0.35		1.5	1.5
Total lines of code	53			85		
Instability		1	0		1	0
Number of parameters		0.857	0.35		0.6	0.49
Lack of cohesion of methods		0	0		0	0
Efferent coupling		0	0		0	0
Number of static methods	1	0.25	0.433	1	0.167	0.373
Normalized distance		0	0		0	0
Abstractness		0	0		0	0
Specialization index		0.5	0.866		0.5	0.764
Weighted methods per class	8	2	0	15	2.5	2.062
Number of static attributes	0	0	0	0	0	0

References

1. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns: Elements of reusable object-oriented software*. ACM Digital Library.
2. Ampatzoglou, A., Charalampidou, S., & Stamelos, I. (2013). Research state of the art on GoF design patterns a mapping study. *Journal of Systems and Software*.
3. Priyalakshmi, G., Nadarajan, R., & Anandhi, S. *Software reuse with shuffler design pattern*. SEAT.
4. Washizaki, H., Yamamoto, H., & Fukazawa, Y. (2003). A metrics suite for measuring reusability of software components. In *Proceedings of 9th International Software Metrics Symposium, METRICS' 03*.
5. Bieman, J. M., & Wang, H. (2006). *Design pattern coupling, change proneness, and change coupling: A pilot study*. Computer Science Technical Report.
6. Lange, D. B., & Nakamura, Y. (1995). *Interactive visualization of design patterns can help in framework understanding*. New York: ACM Press.
7. Khomh, F., & Gueheneuc, Y.-G. (2008). *Do design patterns impact software quality positively?*. New York: IEEE.
8. Bansiya, J., & Davis, C. G. (2002, January). A hierarchical model for object-oriented design quality assessment. *IEEE Transactions on Software Engineering*.
9. Wendorff, P. (2001, March). Assessment of design patterns during software reengineering: Lessons learned from a large commercial project. In *Proceedings of 5th Conference on Software Maintenance and Reengineering* (pp. 77–84).
10. Cohesion metrics: Retrieved March 21, 2018. <http://www.aivosto.com/project/help/pm-oo-cohesion.html>.
11. Software metrics in eclipse: September 2, 2005, Retrieved March 21, 2018. <http://reasearchgroup.org/SEMaterials/tutorials/metrics>.
12. Wand, A. J. A. (2002). Reuse metrics and assessment in component-based development. In *Proceedings of 6th IASTED International Conference on Software Engineering and Applications, IASTED* (pp. 583–588).
13. Henderson-Sellers, B. (1996). *Object-oriented metrics, measures of complexity*. Englewood Cliffs: Prentice Hall.
14. Jabangwe, R., Orstler, J. B., Smite, D., & Wohlin, C. Empirical evidence on the link between object-oriented measures and external quality attributes: A systematic literature review.
15. Eclipse metrics plug-in: December 27, 2002, Retrieved March 21, 2018. <https://sourceforge.net/projects/metrics>.
16. Analyst4j: Provides Java code search using software metrics, Retrieved March 21, 2018. <https://www.theserverside.com/discussions/thread/44693.html>.
17. Alternative tools LOC metrics, Retrieved March 21, 2018. <http://www.locmetrics.com/alternatives.html>.
18. Chidamber and Kemerer Java metrics: May 22, 2010, Retrieved March 21, 2018. <http://www.spinellis.gr/sw/ckjm>.
19. Dependency Finder: Tool to analyse Java codes, Retrieved March 21, 2018. <http://depfind.sourceforge.net>.
20. Testwell: Complexity measures tool for Java, September 20, 2012, Retrieved March 21, 2018. <http://www.testwell.fi/cmtjdesc.html>.
21. Resource Standard Metrics: Quality analysis tool, Retrieved March 21, 2018. <http://www.msquaredtechnologies.com/index.html>.
22. CodePro AnalytiX: Java software testing tool for Eclipse developers, Retrieved March 21, 2018. <https://dzone.com/articles/codepro-integration-eclipse>.
23. Automated tools for software engineering: Retrieved March 21, 2018. <http://www.semdesigns.com>.
24. JDepend: Generate design quality metrics for given Java package, Retrieved March 21, 2018. <http://clarkware.com/software/JDepend.html>.

25. Object oriented software metrics—A short guide: Retrieved March 21, 2018. <http://www.virtualmachinery.com/jhawkmetrics.htm>.
26. jMetra: Output project metrics to XML files, May 12, 2006, Retrieved March 21, 2018. <http://www.loribel.com/java/tools/jmetra.html>.
27. JMetric: OO-metrics tool, June 19, 2001, Retrieved March 21, 2018. <https://sourceforge.net/projects/jmetric>.
28. Essential metrics: Command line metrics for C, C++, and Java projects, February 01, 2015, Retrieved March 21, 2018. <http://www.powersoftware.com/em>.
29. Refactorit: Automated refactorings, April 21, 2016, Retrieved March 21, 2018. <https://sourceforge.net/projects/refactorit>.
30. SonarJ: Visualize and analyze structure of Java code, March 21, 2015, Retrieved March 21, 2018. <http://sonarj.software.informer.com>.
31. Alghamdi, J. S., Rufai, R. A., & Khan, S. M. (2005). OOMeter: A software quality assurance tool. In *Proceedings of 9th European Conference on Software Maintenance and Reengineering*.
32. SemmleCode: Eclipse plugin to query Java code, June 01, 2007, Retrieved March 21, 2018. <http://www.developerfusion.com/thread/47560/semmlcode-free-eclipse-plugin-to-query-java-code>.
33. Understand: Static code analysis tool, Retrieved March 21, 2018. <https://scitools.com>.
34. Löwe, W., Ericsson, M., Lundberg, J., Panas, T., & Pettersson, N. (2003, October). Vizzanalyzer—a software comprehension framework. In *Third Conference on Software Engineering Research and Practise in Sweden*. Sweden: Lund University.

Efficient Algorithms for Text Lines and Words Segmentation for Recognition of Arabic Handwritten Script



Amani Ali Ahmed Ali and M. Suresha

Abstract A new methodology for Arabic handwritten document images segmentation is done in this paper to segment the documents into distinct entities as words and text lines. Based on features of Arabic scripts, the document images are divided into three main subsets of connected components where the Hough transform method is applied to them to achieve text lines segmentation. To enhance the result by avoiding the Hough transform text line detection failure, the authors used a method in postprocessing stage based on skeletonization that covers the possible false correction alarms to create proficiency vertical connected characters' segmentation. The segmentation of the Arabic words is pointed as a two-class problem. The authors used fusion of convex and Euclidean distance metrics to calculate the distance between neighboring overlapped components, which in the Gaussian mixture modeling framework is classified as a distance of an intra-word or as an inter-word. The proposed method performance is depended on a constant and particular evaluation method that appropriate measures of the performance used to compare the segmentation of our result against the other strong researcher result. The proposed method showed higher efficiency and accuracy in the experimentation, which was conducted on two various Arabic handwriting datasets that are IFN/ENIT and AHDB.

Keywords Gaussian mixture modeling · Handwritten document images · Hough transform · Word segmentation · Text line segmentation

1 Introduction

In the area of recognition of handwritten document to segment, segmenting the document images into their fundamental entities such as words and text lines is a very complex problem in Arabic handwritten document images because various kinds

A. A. A. Ali (✉) · M. Suresha

Department of Computer Science, Kuvempu University, Shimoga 577451, Karnataka, India
e-mail: t_amani_ali2@yahoo.com

A. A. A. Ali

Department of Computer Science, Taiz University, Taiz, Yemen

© Springer Nature Singapore Pte Ltd. 2019

N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,

https://doi.org/10.1007/978-981-13-5953-8_32

of difficulties are faced in both process of words and text line segmentation. These complexes make them a challenging task. The main difficulties in the segmentation process of Arabic text line are overlapping words, neighboring text lines touching, and over same text line or on the page between lines the variances angle of the skew. The main difficulties in segmentation procedure of Arabic words are nonuniform spacing which is popular in handwritten words, the punctuation marks within the text line, and the slant and skew in the text line.

A new methodology is introduced in this paper for Arabic handwritten document images segmentation, segmenting the documents into their distinct entities such as text lines and words. Basically, an improved method is included in the proposed method for the vertically connected separation for the text lines, and the new method for word segmentation depends on an efficient recognition of intra-word and inter-word gaps using combination of two various distance metrics, which are convex hull-based and Euclidean distance. The recognition of two classes treated as clustering problem of an unsupervised, Gaussian mixture theory used for model these two classes. The efficiency of the proposed method on two various datasets with various collections of gap metrics is proved in the experimentation. The Arabic handwriting datasets used are IFN/ENIT and AHDB. A fixed benchmarking enables the comparison between existing researchers' methods and the proposed methods. This paper is organized as follows: the related work is presented in Sect. 2 for the segmentation of text line and words with text line of Arabic handwritten document images. The proposed words and text lines segmentation methods are detailed in Sect. 3. The experimental results are presented in Sect. 4, and Sect. 5 describes the conclusions with the future work.

2 Related Work

In the Arabic handwritten document images literature, broad variety methods of segmentation have been reported. The authors classified these methods based on whether they refer to the segmentation of word or text line or both word and text line.

2.1 *Text Line Segmentation*

In [10], the authors used both morphological dilatation and projection profile. To rate skew of the line, the horizontal projection profile method is used. In every zone for smearing, the slope is used, using dilation with adaptive structuring element to do the changes according to the zone, the slope, and the size. The big blobs are detected in the second phase with a recursive function that search for the cut point because segment the components and match them with their lines following repulsion and attraction criterion. Using erosion recursively to do the thinness portion touching components detection which matches the cut point these are detected and segmented.

In [2], the authors used projection profile method after joining broken characters and removing small components to determine the point of separation within the horizontal projection profile; the curve of Fourier fitting is used. To segment the baseline of the connected component which allows determining the cut point between various neighbor lines, the contour is used. Regarding the nearest line, the components are determined, which are approximated by the curve of polynomial that suits in the baselines the pixels. According to the components of the closest connected nearest neighbor in four directions, the lifted of small size components are reassigned to their line.

In [11], the authors begin the algorithm by deleting diacritics, and then, the sparse similarity graph is built based on the local orientation of the component. By breadth first search (BFS) via set of disjoint of text line are performed. To assign to the lines of text the blobs, the affinity propagation clustering method is used.

In [9], the algorithm in this technique begins by eliminating outlier components by using the value of threshold; after that, the letters related to two lines at the half distance are detected and segmented horizontally. For line detection, a rectangular neighborhood on a current component is centered and increases to contain which satisfy specific conditions. At the beginning from their bounding box with respect to distance, the filtered components to the corresponding lines are allocated.

In [8], the algorithm in this technique shows a new algorithm for automatic line of the text segmentation from Arabic handwritten documents, presenting the problems of a multi-touching and an overlapping characters. Using unsupervised method, their method depended on analysis of a block covering. The algorithm in this method in the first step performs the analysis of a statistical block which calculates the typical number into vertical strips of document decomposition. Their method in the next point used fuzzy C means technique which achieves line detection based on fuzzy base. In the final phase to its corresponding lines, blocks are assigned. Experiment results of this approach showed high accuracy, around 95% for detecting lines in Arabic handwritten document images which were written with various document images.

2.2 *Word Segmentation*

In [3], the authors realized a static threshold for gaps classification as “within-a word” and “between-words”, by analyzing more than 250 words that were taken from the IFN/ENIT database through 200 document images. For the errors of the minimum classification, the Bayesian criterion was employed. This method achieved an accuracy rate of around 85%.

In [22], the authors have proposed a new technique for words segmentation on off-line Arabic handwritten. It segmented the connected characters into smaller components where every one of them does not contain more than three characters. Every character may be at most segmented into five parts. Additionally, developing of Arabic words recognition, another possible application of the segmentation of the

proposed method is to build of small size lexicon, including no combinations more than three characters. Generally because lot number of word in Arabic language that make produce lexicon for the unconstraint handwritten Arabic document images recognition too complex.

2.3 Text Line and Words Segmentation

Al-Dmour and Fraij [4] realized dynamic gaps as “within-a word” and “between-words”, which were taken from the input document image itself. Experiments using various clustering methods were performed with an accuracy rate of 84.8%.

3 Methodology

The following challenges are faced by the proposed method for Arabic handwritten segmentation from document images treated with:

- overlapping and touching,
- skewed or slanting, and
- short lines.

3.1 Text Line Segmentation

The authors used the method of connected components to segment the document images into a group of CCs which help us to join probably connecting to each other with the same line which is connected. To join all the components that belong to the same line:

Step 1: In the beginning, the authors used a rating threshold to remove the diacritical components from the documents; those will add them to their corresponding lines in the final phase.

Step 2: Based on the average height $M(H)$ of every connected component for the whole document image, the average of the character height is calculated. The authors suppose the average of the character width $M(W)$ and the average of character height is equal.

Step 3: For every component meet the majority of characters size “Subset 1” is predicted to comprise them where the following constraint is satisfied:

$$(0.5 * M(H) \leq H < 3 * M(H)) \text{ and } (0.5 * M(W) \leq M(W)) \quad (1)$$

where W and H , respectively, are the component's width and height, and $M(W)$ and $M(H)$, respectively, the average width and the average height of the character.

Step 4: For all large connected components, "Subset 2" is expected to comprise them where the characters from neighbor text lines which touch each other are the large components. The following equation defined the height of these connected components:

$$H \geq 3 * M(H) \quad (2)$$

Step 5: Punctuation marks, accents, diacritic marks, Tanween, Hamza, Sha'ada, and the dots of characters, all these characters should include in "Subset 3". The following equation defined this:

$$\begin{aligned} & ((H < 3 * M(H)) \text{ and } (0.5 * M(W) > W)) \\ & \text{or} \\ & ((H < 0.5 * M(H)) \text{ and } (0.5 * M(W) < W)) \end{aligned} \quad (3)$$

Step 6: The gravity center is calculated in this step which contained in every block of connected component after creating the blocks.

Step 7: A line to transform the Cartesian space into the Polar coordinate is Hough transform. The following equation defined the line in the space of the Cartesian coordinate:

$$x \cos(\theta) + y \sin(\theta) = p \quad (4)$$

The line in the space of Cartesian is defined via designating in the space of coordinate of the Polar where their coordinates are p and θ . In the subset, there are matches between the accumulator array and every gravity center to a group of cells of the domain (p, θ) . The resolution along the direction of θ was set to 1° , where the value of θ was in the domain of $85-95^\circ$ and over direction (p) their solution set to $0.2 * M(H)$ to build the domain of Hough as shown in Eq. (4).

Step 8: The cell (p_i, θ_i) is detected by the authors with maximum contribution and assigned every point which vote to the line of the text (p_i, θ_i) in the area $(p_i - 5, \theta_i) \dots (p_i + 5, \theta_i)$. To specify if a connected component belongs to a text line, at least half of the points which form the corresponding blocks must be reallocated to this area. All votes from the accumulator array of the Hough transform that matches this connected component are deleted after the assignment to a connected component text line. This process is repeated in order to avoid false alarms till cell (p_i, θ_i) having the maximum contribution comprises less than n_1 votes. Through the evolution of the process, the angle of the skew of actually detected lines is calculated. The authors applied more constraint in the case when the cell (p_i, θ_i) with less than $n_2 (n_2 > n_1)$ contribution has as a maximum contribution; in that case, the text line is only correct if the matching of the line skew angle deflects from the skew angle less than 2° .

Step 9: To improve the result through Hough transform via making correction for some false alarms, the authors applied a merging method, y_i mentioned to the average y detected line intersection i values are calculated and the bounding box determined by $(i = 1, \dots, n)$ of connected components are calculated.

Step 10: If the condition at Eq. (5) is not satisfied, the authors exclude the last line n from the process; all the components of this subset belong to n detected text lines ($n > 1$).

$$\frac{\sum_{y=y_n-(y_n-y_{n-1})/10}^{x_e} I(x, y)}{\sum_{y=y_{n-1}}^{x_e} I(x, y)} > 0.08 \quad (5)$$

Here the image of the component is I which value 1 mentioned for foreground pixel representation and 0 background pixel representation and are the component coordinates bounding box. The component location near line n is not because a long character descender from text line $n - 1$ and because a vertical character merging which verified in Eq. (5).

Step 11: The authors define zones Z_i for every line $i(i = 1, \dots, n - 1)$, with consideration of the following constraint:

$$y_i + \frac{y_{i+1} - y_i}{2} < y < y_{i+1} \quad (6)$$

The connected component skeleton is computed. We remove all points of the junction and remove them if they are inside zone Z_i from the skeleton. In the Z_i zone of the segmentation, the authors remove all points of the skeleton on the zone center if there does not exist any junction point.

Step 12: With Flag id "1" for all zone Z_i , the skeleton portions with line i are intersected. With flag id "2", all other portions are flagged. In every zone Z_i , the connected component segmentation in the initial stage into various portions is accomplished in the last by assigning the nearest skeleton pixel id to a pixel.

3.2 Word Segmentation

The segmentation process of the Arabic words is as follows:

- The first phase treats the text line with neighbor components computation distances of the images of the document, and
- The second phase treats the gaps of the classification as inter-word or inter-character of previously computed distances.

For the first step twos, the authors proposed different average metrics which are convex hull-based [15] combined with the Euclidean distance metric [19], utilizing a well-known technique which is the Gaussian mixtures where the computed distances classification from the part of unsupervised clustering methods is performed.

3.2.1 Distance Computation

A preprocessing is applied in the text line image to calculate the neighbor components distance. The preprocessing treats the text line of document image with a dominant slant angle and the skew angle correction [21]. On only overlapped components (OCs), the gap metric computation is considered; the OC is a group of connected components where their projection profiles overlap within the vertical direction and not considered on (CCs) the connected components. The authors determine the average value distance based on the convex hull and the Euclidean distance as the distance of two neighbor OCs. The minimum between of each pairs Euclidean distances of the two neighbor OCs points is determined the Euclidean distance among two adjacent OCs. For the Euclidean distance calculation, the authors apply a quick planned that takes into consideration the pixels' subset of the right and left OCs only rather than the whole number of black pixels. This subset comprises the black pixel of the right-most of every scan line instead of determining the subset with the left pixels of OC. Including the leftmost black pixel of all scan line is defined the subset of pixels for the right OC. The minimum Euclidean distances of every pixels pairs are defined finally the distance of the Euclidean of two OCs. As the following, the authors calculate the metric based on the convex hull: C_i and C_{i+1} , which are a pair of neighbor OCs, and H_i and H_{i+1} , which are their convex hulls are given. The line which linking the gravity centers (or centroid) of H_i and H_{i+1} is denoted by L . The intersection points of L with the hulls H_i and H_{i+1} are denoted by P_i and P_{i+1} , respectively. Between the points P_i and P_{i+1} , the Euclidean distance has defined the gap between the two convex hulls.

3.2.2 Gap Classification

A novel method is used for the gap classification problem. This method is depended on the classification of the unsupervised the already computed distances into two distinguishable classes forming, respectively, the word interclass and the word intra-class. The authors adopt the Gaussian mixtures method to this end. A clustering with mixture model based on every cluster is mathematically introduced via a parametric distribution. The authors have a two-cluster problem; hence, every cluster is modeled with a Gaussian distribution. The EM algorithm is the algorithm that is used to calculate the parameters for the Gaussians. Because the Gaussian mixture is a well-known approach for unsupervised clustering, the authors used this method with many advantages which include the following:

- i. A soft classification is available.
- ii. For each cluster, the density rating can be obtained.
- iii. The mixture model covers well the data. For the Gaussian mixtures, the detailed description is given in [16].

4 Evaluation and Experimental Result

4.1 Performance Evaluation Methodology

Observation manually of the result of segmentation is not in every case unbiased procedure, a high time-consuming, and boring. To avoid user interference, the authors used an automatic evaluation of the performance method that compared the result of detected segmentation with a previously annotated ground truth. The performance evaluation is depended on counting the matches number between the areas inside the areas which detected by the method and the areas inside the ground truth. There is a table the authors used it where its values are calculated according to the labeled pixel sets overlap as the ground and either text lines or words.

Where the set of every image points is I , all points set inside the i word or text line ground truth region is G_i ; all points set inside the j word or text line result region is R_j , a function that counts the set elements s is $T(s)$. As follows, table $MS(i, j)$ forms the corresponding results of the region j of result and the region i of ground truth:

$$MS(i, j) = \frac{T(G_i \cap R_j \cap I)}{T((G_i \cup R_j) \cap I)} \quad (7)$$

where MS refer to MatchScore, along the MatchScore table the performance evaluator seeks for one-to-one, many-to-one, or one-to-many corresponding pairs. o2o denotes a pair of one-to-one correspond if the pair corresponding score is above or equal to the acceptance threshold θ evaluator. go2m denotes g_one-to-many correspond which is a word or text line ground truth, partially that corresponds with the detected result of more than one of word or text line. gm2o denotes g_many-to-one correspond, which corresponds to more than one of the ground truth word or line of text, partially that corresponds with one word or text line in the detected result. do2m denotes d_one-to-many correspond which is detected word or text line, partially that corresponds with two or more words or line in the text in the ground truth. dm2o finally denotes d_many-to-one correspond, which corresponds with more than one words or detected text lines, partially that match one word or line of the text in the ground truth. In the following, the equations of detection rate (DR) and recognition accuracy (RA) are as follows:

$$DR = w_1 \frac{o2o}{N} + w_2 \frac{g_o2m}{N} + w_3 \frac{g_m2o}{N} \quad (8)$$

$$RA = w_4 \frac{o2o}{M} + w_5 \frac{d_o2m}{M} + w_6 \frac{d_m2o}{M} \quad (9)$$

where the ground truth segments words or lines of the text count are denoted by N , the result segments words or text lines count are denoted by M , and predetermined weights are $w_1, w_2, w_3, w_4, w_5, w_6$. From Eq. (7), the following $o2o, g_m2o, d_o2m, g_o2m$, and d_m2o entities which correspond to the one-to-one, g_many -to-one, d_one -to-many, g_one -to-many, and d_many -to-one numbers are calculated following the steps of [18]. If the authors merge the values of RA and DR the metric of a global performance can be defined. The F -measure (FM) equation is as follows:

$$FM = \frac{2 * DR * RA}{DR + RA} \quad (10)$$

4.2 Experimental Results

The proposed algorithm is tested on various Arabic handwritten databases, which are IFN/ENIT and AHDB databases. The authors conducted various experiments and explained them in detail in the following parts. For the segmentation of the text line, the authors implemented the Hough-based method [12], the fuzzy RLSA [20], and the projection profiles [7]. For skew correction, the authors implemented the method based on [1].

In the proposed method of segmentation of the text line, the parameters n_1 and n_2 in (Sect. 3.1) are set with $n_1 = 5, n_2 = 9$ experimentally. Furthermore in (Sect. 4.1) for the segmentation of word and text line, the acceptance threshold θ evaluator's is defined, respectively, as 0.99 and 0.94.

4.2.1 AHDB

The AHDB which denotes Arabic handwriting database [5, 6] is an Arabic handwriting database processed with numerous preprocessing. It consists of words; paragraphs and the words are used to form digits conducted by 100 various writers on checks of Arabic handwritten. The AHDB contained unconstrained pages' text. The proposed methodologies' efficiency was tested by proceeding experiments on Arabic handwritten document images.

The authors have the corresponding ground truth for all the required images which comprise words and text lines. The number of words is 13,311, and the corresponding number of lines is 1773. As shown in detail in Table 1, the segmentation results of the text line in the cases of FM, RA, and DR and the number of corresponds are given. The authors combined the distance metrics defined with two gap classification techniques in Sect. 3.2.1 for concerning word segmentation. These include the following:

- i. The classification method of Gaussian mixture.

Table 1 Experimental results for text line segmentation over AHDB handwritten

<i>N</i>	<i>M</i>	o2o	dm2o	go2m	gm2o	do2m	RA	DR	FM
1095	1093	1082	4	2	4	2	99.1	98.9	99.0

Fig. 1 Words segmentation of free Arabic handwritten document images from AHDB database

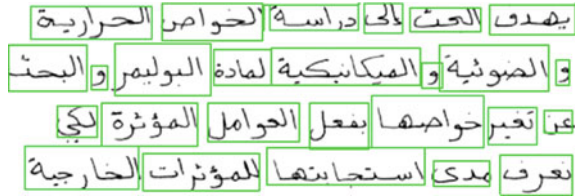
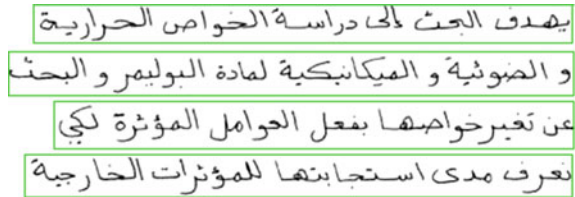


Fig. 2 Text lines segmentation of free Arabic handwritten document images from AHDB database



- ii. The method in [13] which used the lengths median on the scan line of white pixels that have to white transitions the maximum number of black.

Table 2 shows the experimental segmentation results of words in the cases of FM, RA, and DR and gives the matches number. EDM, CDM, GMCM, and LTCM in Table 2 represent Euclidean distance metric, convex distance metric, LT classification methods, and GM classification methods. The proposed result of segmentation method of the text line took it as input to the segmentation module of the words. The proposed method is parameter free.

Table 3 shows the results of experimental after combination of every segmentation methodologies of the words with each segmentation methodologies of the text line. The segmentation results of word process and text line process visualize, respectively, are shown in Figs. 1 and 2.

4.2.2 IFN/ENIT

The IFN/ENIT includes words of towns' names and villages' names in Tunisia and postal code, which are all represented in the Arabic handwritten database IFN/ENIT [17]. By filling specified form, the dataset of 411 volunteers has been developed. In the database, the total names words city and town reached to 26,400 including 210,000 characters. The data of ground truth with the database contains information on the writer sequence, baseline, and character shapes. All filled forms are stored as binary and digital images at 300 dpi.

Table 2 Word segmentation experimental results with fusion of gap classification methodologies and distance metrics over AHDB handwritten

EDM	CDM	LTCM	GMCM	N	M	o2o	gm2o	go2m	dm2o	do2m	DR	RA	FM
0	0	1	0	510	1546	235	8080	6655	697	8576	81.1	87.8	84.3
0	1	0	1	1190	956	554	8810	6895	445	8576	85.1	82.5	83.8
1	0	0	1	733	1419	342	8282	6674	643	8576	96.8	92.9	94.8
0	0	0	1	630	1280	292	8305	6854	586	8576	83.4	87.2	85.3
1	0	1	0	1050	843	485	8809	7087	397	8576	86.8	84.2	85.5

Table 3 Comparison between well-known methods and proposed method

Method	Measures	Accuracy (%)
[3]	Word segmentation	85
[4]	Text and word segmentation	84.8
[8]	Text line segmentation	95
Proposed work	Text line segmentation	98.9
Proposed work	Word segmentation	96.8

Table 4 Experimental results for text line segmentation over IFN/ENIT handwritten

<i>N</i>	<i>M</i>	o2o	dm2o	go2m	gm2o	do2m	RA	DR	FM
1773	1770	1717	34	6	13	17	97.4	97.4	97.4



Fig. 3 Words segmentation of free Arabic handwritten document images from IFN/ENIT database

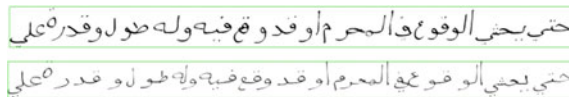


Fig. 4 Text lines segmentation of free Arabic handwritten document images from IFN/ENIT database

The authors have the corresponding ground truth for all the required images which comprise words and text lines. The number of words is 8576, and the corresponding number of lines is 1095. As shown in detail in Table 4, the segmentation results of the text line in the cases of FM, RA, and DR and the number of corresponds are given. For word segmentation problem, the experiment that is applied to these sets is the same which is applied to the AHDB set.

Table 5 shows the experimental segmentation results of words in the cases of FM, RA, and DR and gives the matches number. EDM, CDM, GMCM, and LTCM in Table 2 represent Euclidean distance metric, convex distance metric, LT classification methods, and GM classification methods. The proposed result of segmentation method of the text line took it as input to the segmentation module of the words. The proposed method is parameter free. Table 6 shows the results of experimental after combination of every segmentation methodologies of the words with each segmentation methodologies of the text line. The segmentation results of word process and text line process visualize, respectively, are shown in Figs. 3 and 4.

Table 5 Word segmentation experimental results with fusion of gap classification methodologies and distance metrics over FN/ENIT handwritten

EDM	CDM	LTCM	GMCM	N	M	o2o	gm2o	go2m	dm2o	do2m	DR	RA	FM
0	0	1	0	13,311	13,249	11,953	779	334	764	367	91.8	92.3	92.0
0	1	0	1	13,311	13,622	12,093	1082	206	454	514	93.2	90.5	91.8
1	0	0	1	13,311	13,655	12,190	1062	175	371	503	94.9	90.8	92.8
0	0	0	1	13,311	13,322	11,933	869	326	732	410	91.8	91.7	91.7
1	0	1	0	13,311	13,334	12,018	828	302	673	390	92.4	92.1	92.2

Table 6 Comparison between well-known methods and proposed method

Method	Measures	Accuracy (%)
[3]	Word segmentation	85
[4]	Text and word segmentation	84.8
[8]	Text line segmentation	95
Proposed work	Text line segmentation	97.4
Proposed work	Word segmentation	94.9

5 Conclusion

A methodology of Arabic handwritten documents segmentation which segments the document into their distinct entities as lines and words of the text is presented. The proposed approach is composed of (i) an improved method for the vertically connected separation for the lines of text, and (ii) new method for word segmentation which depends on an efficient recognition of intra-word and inter-word gaps using combination of two various distance metrics which are convex and Euclidean distance metrics. The authors utilized the theory of Gaussian mixture for shaping the two classes where distinguish between these two classes is treated as clustering problem of an unsupervised. The authors show in the experimental results that the proposed algorithms are more accurate than the existing state-of-the-art algorithms for segmenting of the line and word of the text in Arabic handwritten script. Basically in the future work, the attention needs to be paid to improve the segmentation of word using dots, touching lines, and punctuation detection method, and also to improve a feedback from the modules of the character segmentation and character recognition.

References

1. Amani, A. A. A., & Suresha, M. (2017). A novel approach to correction of a skew at document level using an Arabic script. *International Journal of Computer Science and Information Technologies*, 8(5), 569–573.
2. Adiguzel, H., Sahin, E., & Duygulu, P. (2012). A hybrid for line segmentation in handwritten documents. In *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, (pp. 503–508), September 18–20, 2012.
3. AlKhateeb, J. H., Jiang, J., Ren, J., & Ipson, S. (2009). Interactive knowledge discovery for baseline estimation and word segmentation in handwritten Arabic text. In M. A. Strangio (Ed.), *Recent Advances in Technologies*.
4. Al-Dmour, A., & Fraij, F. (2014). Segmenting Arabic handwritten documents into text lines and words. *International Journal of Advancements in Computing Technology (IJACT)*, 6(3), 109–119.
5. Al-Maadeed, S., Elliman, D., & Higgins, C. A. (2004). A data base for Arabic handwritten text recognition research. *International Arab Journal of Information Technology*, 1, 117–121.

6. Al-Maadeed, S., Elliman, D., & Higgins, C. A. (2002). A database for Arabic handwritten text recognition research. In *Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition* (pp. 485–489).
7. Ataer, E., & Duygulu, P. (2006). Retrieval of Ottoman documents. In *Proceedings of the Eighth ACM SIGMM International Workshop on Multimedia Information Retrieval*, 26–27, Santa Barbara, CA, USA, October 2006.
8. Boussellaa, W., & Zahour, A., et al. (2010). Unsupervised block covering analysis for text-line segmentation of Arabic ancient handwritten document images. In *20th International Conference on Pattern Recognition (ICPR)*. New York: IEEE.
9. Khandelwal, A., Choudhury, P., Sarkar, R., Basu, S., Nasipuri, M., & Das, N. (2009). Text line segmentation for unconstrained handwritten document images using neighborhood connected component analysis. In: *Pattern Recognition and Machine Intelligence* (pp. 369–374). Berlin: Springer.
10. Khayyat, M., Lam, L., Suen, C. Y., Yin, F., & Liu, C. L. (2012, March). Arabic handwritten text line extraction by applying an adaptive mask to morphological dilation. In *10th IAPR International Workshop on Document Analysis Systems (DAS)* (pp. 100–104). New York: IEEE.
11. Kumar, J., Abd-Almageed, W., Kang, L., & Doermann, D. (2010, June). Handwritten Arabic text line segmentation using affinity propagation. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems* (pp. 135–142). New York: ACM.
12. Likforman-Sulem, L., Hanimyan, A., & Faure, C. (1995). A Hough based algorithm for extracting text lines in handwritten documents. In *Proceedings of the Third International Conference on Document Analysis and Recognition* (pp. 774–777), Montreal, Canada.
13. Louloudis, G., Gatos, B., & Pratikakis, I. (2008, August). Line and word segmentation of handwritten documents. In *International Conference on Frontiers in Handwriting Recognition (ICFHR'08)* (pp. 247–252), Montreal, Canada.
14. Louloudis, G., Halatsis, K., Gatos, B., & Pratikakis, I. (2006, October). A block-based Hough transform mapping for text line detection in handwritten documents. In *The 10th International Workshop on Frontiers in Handwriting Recognition (IWFHR)* (pp. 515–520), La Baule.
15. Mahadevan, U., & Nagabushnam, R. C. (1995). Gap metrics for word separation in handwritten lines. In *The Third International Conference on Document Analysis and Recognition* (pp. 124–127), Montreal, Canada.
16. Marin, J. M., Mengersen, K., & Robert, C. P. (2005). *Bayesian modelling and inference on mixtures of distributions, handbook of statistics* (Vol. 25). Amsterdam: Elsevier-Sciences.
17. Pechwitz, M., Maddouri, S. S., Maergner, V., Ellouze, N., & Amiri, H. (2002). IFN/ENIT-database of handwritten Arabic words. In *Proceedings of CIFED* (Vol. 2, pp. 127–136).
18. Phillips, I., & Chhabra, A. (1999). Empirical performance evaluation of graphics recognition systems. *IEEE Transaction of Pattern Analysis and Machine Intelligence*, 21(9), 849–870.
19. Seni, G., & Cohen, E. (1994). External word segmentation of off-line handwritten text lines. *Pattern Recognition*, 27(1), 41–52.
20. Shi, Z., & Govindaraju, V. (2004). Line separation for complex document images using fuzzy run length. In *First International Workshop on Document Image Analysis for Libraries* (p. 306).
21. Vinciarelli, A., & Luettin, J. (2001). A new normalization technique for cursive handwritten words. *Pattern Recognition Letters*, 2(9), 1043–1050.
22. Wshah, S., & Shi, Z., et al. (2009). Segmentation of Arabic handwriting based on both contour and skeleton segmentation. In *10th International Conference on Document Analysis and Recognition, ICDAR'09*. New York: IEEE.

Privacy-Preserving Lightweight Image Encryption in Mobile Cloud



M. Sankari and P. Ranjana

Abstract With the rapid development of mobile cloud, multimedia data like image is a major challenge to maintain security and privacy for mobile users. Encryption is one of the best solutions to protect data. However, traditional encryption algorithms are not suitable for images which they were proposed for text data. In this paper, we propose an image encryption technique called privacy-preserving lightweight image encryption (PPLiIE) to make an encryption in a simple way, suitable for images and to maintain privacy. PPLiIE algorithm proceeds with a three-step process to secure the image data in mobile before storing to the cloud. We implement in Python language and analyze our results with various file images to conclude that the encryption time of the PPLiIE is reduced to 50% approximately than the encryption time of AES. In addition, the measurement of key sensitivity and file with variation of chunk size have expressed superior performance of PPLiIE. Finally, we review various security attacks against PPLiIE to express the security level.

Keywords Mobile cloud · Data privacy · Image · AES · Mobile computing · Cloud computing

1 Introduction

Due to the increasing development of the mobile cloud, users have a lot of personal information like photos, images, videos, audio that are stored in cloud [1]. They can do all computation in cloud than mobile and they believe in third-party cloud vendors to keep our data secure [2]. It reduces the usage of resources, memory, battery life in mobile. However, untrusted third party in cloud may be possible to lose our data. Therefore, maintaining security is one of the greatest challenges for moving mobile

M. Sankari (✉) · P. Ranjana
Department of CSE, Hindustan Institute of Technology and Science, Chennai 600025, India
e-mail: vpsankarim@gmail.com

© Springer Nature Singapore Pte Ltd. 2019
N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_33

403

data to the cloud. Encryption is the best technique, which is converting to cipher text from plain text to secure data from unauthorized user. There are various encryption algorithms developed to secure data in mobile such as AES [3], RSA [4], DES, 3-DES, Blowfish. It is mainly suitable for text data than image data. In early stages, mobile data are moved to cloud [5], where computation and encryption take over in cloud. Later, due to the lack of privacy issues, partial data are encrypted and passed to cloud [2]. Further, all data are encrypted in mobile before storing it in the cloud. However, computational overhead is increased. So we can move onto the lightweight image encryption [6, 7] to reduce the computational overhead and maintain privacy. The major drawback of encryption is it takes longer time to execute the code. In addition, various drawbacks are outlined and the following encryption techniques are used to secure data.

The security mechanism uses RSA algorithm [4] to secure the data by the process of encryption and decryption. Encryption is used to provide data security during transmission from mobile to cloud. Users can believe their data are safe and secure because of storing encrypted file to the cloud. The encrypted file is transferred over the channel, where data leakage is reduced. Being unknown of data owner's key, third person/unauthorized user cannot access the encrypted file. Attribute-based encryption mechanisms [8] encrypts/decrypts the data based on the user's attribute. It defines access policies for other users to utilize data and protect the private data. Encrypted file is accessed by only those authorized users that satisfy the access policies. A (M-HABE) modified hierarchical Attribute-Based Encryption [9] method develops the three-layered structure to provide better security. It is developed to ensure the users with legal authorities. The user can satisfy the requirements of the algorithm to access the ciphertext. The three major roles of the methods are data process, storage, and access.

Biometric authentication [10] is another authentication technique, which is used to protect the access of unauthorized users and database. It examines with the Samsung Galaxy S3, BlackBerry Z to define the process time for an image taken (Maxi. Time and Avg. Time are calculated). In [3], the author explains to protect the user data and the retrieval keywords by existing ciphertext mechanism and ORAM (oblivious) is introduced to protect the privacy security of the cipher access patterns. Also, the performance of ciphertext retrieval scheme is improved than other security-level schemes. In [11], the author has explained the solution of privacy-preserving data storage by integrating PRF-based key management for unlinkability, a secure indexing method for privacy-preserving keyword search and access pattern hiding scheme based on redundancy. It is implemented in private cloud to build privacy for mobile health systems. It provides auditability for misusing health data. In [12], the author proposed a secure and efficient index to locate the users with the huge amount of encrypted image and applied two encryption approaches for reproducing the secure image in the cloud. This approach produces 90% bandwidth consumption in the mobile. Table 1 represents the traditional data privacy techniques available in mobile cloud.

Table 1 Existing cryptography data privacy techniques in cloud and mobile cloud

Works	Technique/s	Encryption used	Security enforcement	Issues
An efficient and secure data storage in mobile cloud computing through RSA and Hash function [4]	RSA algorithm and Hash function	Yes	Confidentiality	Utilizes more CPU resources, high memory for encryption
			Data privacy	
			Authentication	Unauthorized user penetrates when key is known
			Data integrity	
Securing mobile cloud data with personalized attribute-based meta information [8]	ABE-attribute-based encryption	Yes	Authentication	Encryption based only on the attribute
			Data confidentiality	
			Availability	Randomized keys only can find the decryption
			Data privacy	
A modified hierarchical attribute-based encryption access control	M-HABE Model	Yes	Data privacy	Attack of any layers affect the users data through encryption and decryption
			Authentication	
			Confidentiality	
Method for mobile cloud computing [9]				
Securing Mobile Cloud Computing Using Biometric Authentication (SMCBA) [10]	Biometric authentication	N/A	Authentication	High response time longer time needed with database comparison
			Data privacy	
The cloud storage ciphertext retrieval scheme based on ORAM [3]	AES, ORAM	Yes	Data privacy	Hierarchical ORAM takes extra time for execution
			Data availability	
Cloud-assisted mobile-access of health data with privacy and auditability [11]	ABE, search index method, PRP	Yes	Auditability	High memory power required for key management and decryption
			Data storage privacy	
Harnessing encrypted data in cloud for secure and efficient mobile image sharing [12]	Homomorphic Encryption (HE), symmetric encryption and image reproduction	Yes	Data privacy	Increasing process time for image storage in cloud

Here, most of the papers focused on the security and privacy plays a major role for the mobile user. It referred many techniques such as RSA, ABE, Biometric, HE, AES, PRP, and so on. RSA is used for double encryption, ABE is used for encryption based on attribute/s, Biometric is used for user's authentication, HE is used for image sharing with assuring privacy, AES for maintaining data privacy by encryption, PRP is used for permutation-based encryption [13] especially for lightweight encryption. Mobile complexity acts as the great issue for all of these methods and encryption takes longer time to execute in mobile. Therefore, we propose an image encryption technique (PPLiIE) [14, 15] to secure data and reduce complexity, improve the throughput to maintain privacy before storing into the cloud by overcoming the traditional techniques. It is implemented in Python where Python is a scripting, high-level language, and easily understandable code. It is an open source and image library is included such as PyCrypto encryption and decryption library for execution. It is very suitable for image processing.

The remaining sections are as follows: In Sect. 2, we present the proposed PPLiIE method deeply with processes and its schema. In Sect. 3, we implement the PPLiIE method with various metrics in Python language. In Sect. 4, we define the decryption process of the PPLiIE. In Sect. 5, we describe the security analysis against the PPLiIE method. Finally, we conclude the implemented method.

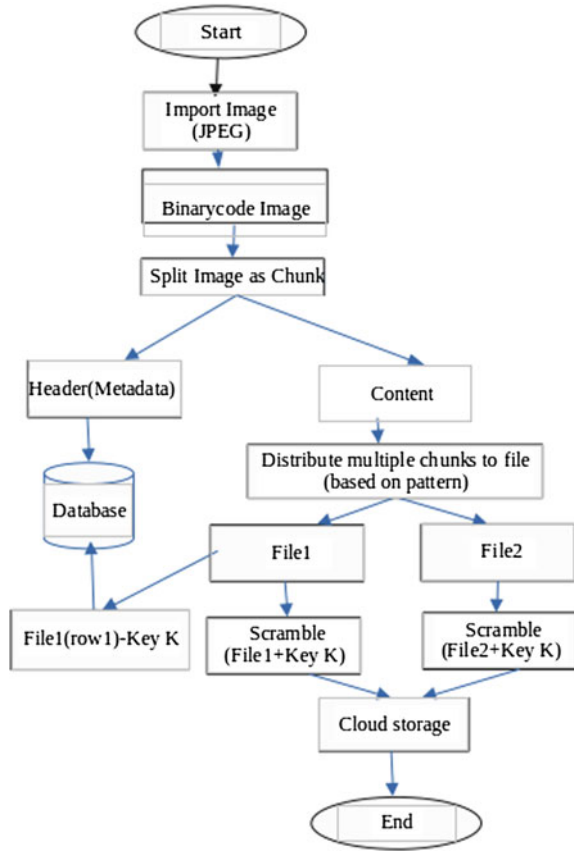
2 Proposed PPLiIE Method

The PPLiIE method defines the image split into chunks, distribute to different files based on pattern and scramble based on file key to maintain the user's data privacy rather than traditional technique such as AES. Consider the JPEG image probably used in mobile and is split as chunks (like 2 bytes as chunk). Chunks is grouped by pattern and make as files. Key K formed by distributed file and stored in Database. Finally, we scrambled the files and store it in cloud.

2.1 Layout of the PPLiIE Schema

See Fig. 1.

Fig. 1 Layout of the PPLiE schema



2.2 Basic Three Steps Required for the PPLiE Method

2.2.1 Split

The original image file is split into two divisions such as the header and the content of the image file. Header file contains file type, size, chunk size, date created, width, height, and resolution. The equation used for a split file is

$$\text{Image_file}_i = H_i + \sum_{k=1}^{k=\max} C_{i,k} \tag{1}$$

where H_i as a header of the image file, $C_{i,k}$ as the number of chunks formed.

$$\text{Max} = [(\text{File_Size}_i / \text{Chunk_size}_i) - H_Size] \tag{2}$$

where File_Size_i denotes the size of the file, Chunk_size_i denotes the size of chunks, and H_size denotes the header's size of the original file. All are represented in bytes.

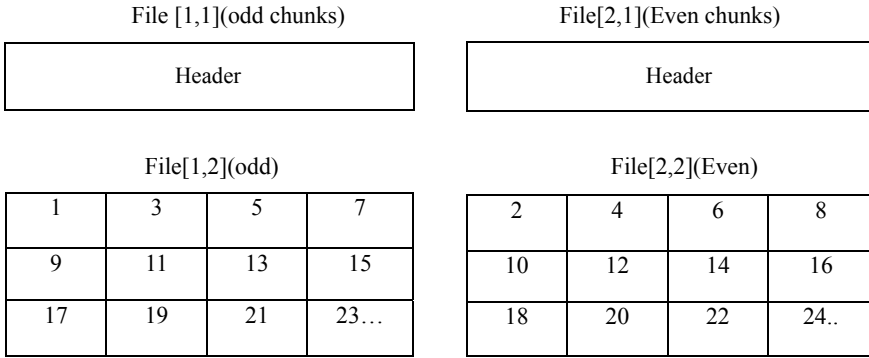


Fig. 2 Example of pattern for the PPLiE method

2.2.2 Distribute (Pattern)

After splitting the image, chunks are grouped into the different files based on the pattern. A user can act a pattern as a key or it can be selected randomly by a predefined method. Consider as an example, the proposed PPLiE method takes a pattern as odd chunks act as file1, even chunks act as a file2 and continues (Fig. 2).

2.2.3 Scramble

After completion of the distribution of chunks to files, scramble the file within it by adding Key K_1 (first row of the file1) with all rows of the files. Key K_i stored in the database.

$$K_i = \text{File}_{i,2}(\text{Row}_1) \tag{3}$$

where $\text{File}_{i,2}$ represents the distribute file split by pattern, Row_1 represents the first row (first two bytes) of the $\text{File}_{i,2}$.

$$\sum_{i=1}^2 \text{File}_{i,2} = K_i + \sum_{j=1}^m \text{File}_{i,2}(\text{Row}_j) \tag{4}$$

where n denotes the number of files split by pattern, $\text{File}_{i,2}$ denotes the i th file split by pattern, Row_j denotes the j th row of the $\text{File}_{i,2}$, m denotes the number of rows in the $\text{File}_{i,2}$.

2.3 Algorithm

Start

```

Input img ← ".jpeg"
Image filei ← bin (Img)
for i ← 0 to n-1 do
for k ← 0 to max do
Image_filei = Hi + ∑k=1max Ci,k (Split)
[Header Hi is stored in Database]
where Max = [(File_Sizei/Chunk_sizei)-H_Size]
End for
End for

File1,2 ← Collection of even chunks (Ci,k[even])
File2,2 ← Collection of odd chunks (Ci,k[odd])

for i ← 1 to 2
Ki = Filei,2 (Row1)
Ki → Database
for j ← 0 to m-1
Filei,2 = Ki + Filei,2(Row1)
End for
End for
File → cloud storage (Encry_Image)
End
    
```

3 Implementation

We have implemented the proposed PPLiIE method in Python language and used hardware as Windows 7 OS. Consider the size of the chunk as 64, key size of AES as 16 bytes, key size of proposed PPLiIE method as the first row of the file1 and

Table 2 Various file size images with execution time (ms)

Image name	Pixel range	File size	Encryption algorithm [Execution Time (ms)]	
			AES	PPLiIE
Cat	244*206	4.52	4000	1000
Penguin	225*225	5.669	4000	1000
Flower	259*194	7.85	4000	2000
Lena	225*225	8.1	4000	2000
Bear	318*159	10.7	5000	2000

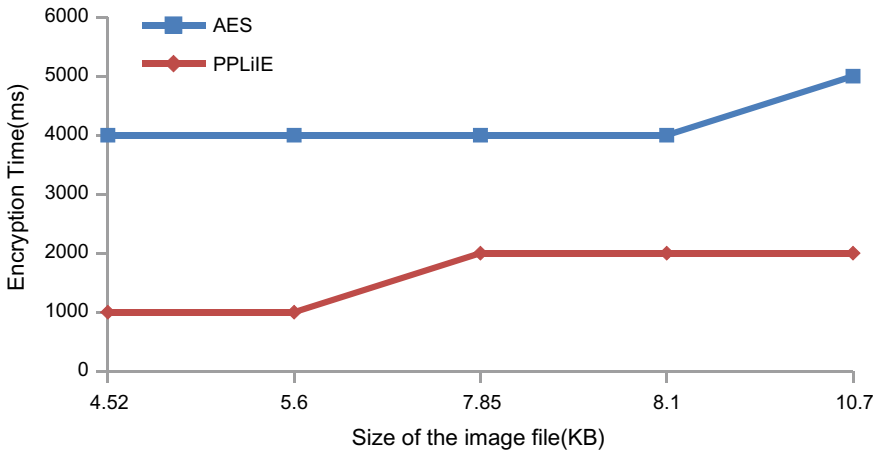


Fig. 3 A comparison of the proposed PPLiIE method against AES encryption with various file image sizes

pattern taken as even and odd chunks in Fig. 2. Table 2 represents the various file size images with execution time by using AES and PPLiIE.

We analyzed with the sample jpeg images and concluded that the encryption time of the proposed PPLiIE method is reduced approximately to 50% which is represented in a line chart compared with AES encryption in Fig. 3.

3.1 Key Sensitivity

Key sensitivity is an important factor to execute an algorithm. Large key maintains more security for image and small key reduces the resource used in mobile. AES have three long keys such as 16, 24, 32 bytes to encrypt the data. PPLiIE method has taken the key automatically from the first row of the first file (K) in an image. There is no manual requirement to assign the key. To ensure privacy, storing key in user's mobile and never pass it to cloud.

3.2 File with Chunk Size Measurement

In the proposed method, chunk size is used for splitting the image file. The size of the chunk in AES sets in which the function uses to read and encrypt the image file. When the size of the chunk is increased, encryption time is also increased in AES. However, in our proposed method, the encryption time is relatively constant even with the change of chunk size. It proves to have the better performance such as

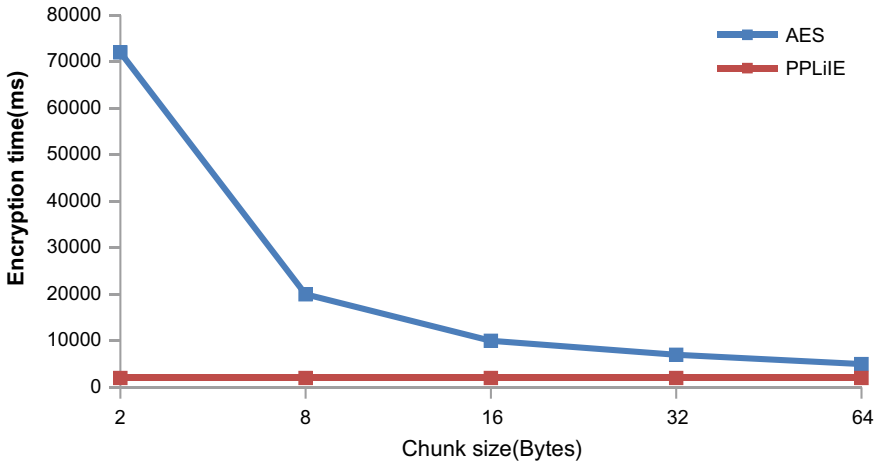


Fig. 4 Encryption time varied with chunk size

speed, the limited resource used, lightweight process for all chunk sizes. Consider the image .jpeg file as bear with file size as 10.7 MB and variation of chunk size are taken for the line chart. Figure 4 represents the execution time of the encryption in AES and PPLiE method with various chunk sizes.

Our result of the conclusion decided that the lower encryption time in PPLiE with the variation of chunk size than AES. Therefore, it have superior performance and reduced the overall execution time.

4 Decryption Process

We can retrieve the encrypted image data from cloud to mobile. Using the key stored in mobile, subtract them with the file1 and file2. We can merge the files based on pattern. Chunks of image data are connected based on chunk size with reference of the header file. Binary code images are converted to jpeg image. Now, image is assembled from the disassembled image. Mobile user retrieved the image from the cloud.

5 Security Analysis

We have stored our image-related data such as key, metadata of the header file, pattern, and chunk size in mobile to maintain privacy. Assume that the unauthorized

user is gathering the image encryption data in the cloud. In addition, it is impossible to retrieve the original image due to the lack of key, pattern and the chunk size.

We present various scenarios of security attacks in the proposed method. An attacker mainly requires key k_1 , scramble method, pattern, header information, and the size of the chunks to retrieve the original image data. Consider an image as 10×10 bytes, chunk size as 2 bytes.

Scenario 1:

Assume the key K_i is known, but the attacker does not know the pattern and the size of the chunks.

If the attacker knows the key K_i , he retrieves the file1 and file2. But he cannot know the scrambled method. So, he would not move to the distribute process (pattern). Assume he may apply brute force attack to know the scrambled method. The minimum probability of checking with known key are $12! = 479,001,600$ possibilities needed to retrieve the scrambled data.

Scenario 2:

Assume that the key K_i and the scrambled method are known, he does not know the pattern of file.

An attacker retrieves the file1 and file2. But file1 has approximately 13 parts and file2 has 12 parts. Assume that the attacker checks with consequent pattern, odd/even pattern, and random pattern. Consequent pattern such as 3 or 4 or 5 are taken. For example, first 3 chunks move to file1 next 3 moves to file2 and so on. There are 9 possibilities for 3 chunks taken, 7 possibilities for 4 chunks, 5 possibilities for 3 chunks, and so on. The total probabilities are $9 + 7 + 3 = 19$. The same procedure for odd/even pattern that is first odd in first1 and second odd in file2 and so on. Randomly selected pattern are impossible to retrieve it.

Scenario 3:

Assume the key k_1 , the scramble method and the pattern are known, unknown the size of the chunks.

An attacker can check the chunks with different sizes such as 2, 3, 4, 5, 6 bytes. 25 chunks formed by 2 bytes 20 by 3, 4 by 15, 5 by 10, 6 by 9. The minimum total possibilities for identifying chunk size are $25 + 20 + 15 + 10 + 9 = 79$. Even the chunk size may have 7, 8, 9 and so on. So the attacker is difficult to retrieve the chunk size.

6 Conclusion

The proposed PPLiIE method secured the image data by simple lightweight encryption and to maintain user's privacy in mobile. It also provides a clear, effective performance, data privacy solution for mobile devices which have limited resources. We have taken the sample images for performance analysis, which is implemented in Python language. It is proved to reduce more than half of the encryption time compared to existing method like AES. Finally, we have introduced security attack scenarios to explain the difficulty to access the proposed method from attackers.

References

1. Zhang, X., Schiffman, J., Gibbs, S., Kunjithapatham, A., & Jeong, S. (2009, November). Securing elastic applications on mobile devices for cloud computing. In *Proceedings of ACM Workshop on Cloud Computing Security, CCSW '09*, Chicago, IL, USA.
2. Mollah, M. B., Azad, M. A. K., & Vasilakos, A. (2017). Security and privacy challenges in mobile cloud computing: Survey and way ahead. *Journal of Network and Computer Applications*, pp. 38–54.
3. Song, N., & Sun, Y. (2014). The cloud storage ciphertext retrieval scheme based on ORAM. *China Communications*, 11(14), 156–165. (IEEE Journals & Magazines).
4. Garg, P., & Sharma, V. (2014). An efficient and secure data storage in mobile cloud computing through RSA and Hash function. In *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*. New York: IEEE Society.
5. Qin, Z., Weng, J., Cui, Y., & Ren, K. (2018). Privacy-preserving image processing in the cloud. *IEEE Cloud Computing*, 99 (IEEE Early Access Articles).
6. Bahrami, M., Li, D., Singhal, M., & Kundu, A. (2016). An efficient parallel implementation of a light-weight data privacy method for mobile cloud users. In *2016 Seventh International Workshop on Data-Intensive Computing in the Clouds (DataCloud)* (pp. 51–58).
7. Bahrami, M., & Singhal, M. (2016). CloudPDB: A light-weight data privacy schema for cloud-based databases. In *2016 International Conference on Computing, Networking and Communications, Cloud Computing and Big Data*.
8. Zickau, S., Beierle, F., & Denisow, I. (2015). Securing mobile cloud data with personalized attribute-based meta information. In *2015 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*.
9. Xie, Y., Wen, H., Wu, B., Jiang, Y., & Meng, J. (2014). A modified hierarchical attribute-based encryption access control method for mobile cloud computing. *IEEE Transaction on cloud Computing*.
10. Rassan, A., & AlShaher, H. (2014). Securing mobile cloud computing using biometric authentication (SMCBA). In *2014 International Conference on Computational Science and Computational Intelligence (CSCI)*.
11. Tong, Y., Sun, J., Chow, S. S. M., & Li, P. (2014, March). Cloud-assisted mobile-access of health data with privacy and auditability. *IEEE Journal of Biomedical and Health Informatics*, 18(2).
12. Cui, H., Yuan, X., & Wang, C. (2017). Harnessing encrypted data in cloud for secure and efficient mobile image sharing. *IEEE Transactions on Mobile Computing*, 16(5).
13. Bahrami, M., & Singhal, M. (2015). A light weight permutation based method for data privacy in mobile cloud computing. In *2015 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*.

14. Bahrami, M., Khan, A., & Singhal, M. (2016). An energy efficient data privacy scheme for IoT devices in mobile cloud computing. In *2016 IEEE International Conference on Mobile Services*.
15. Balouch, Z. A., Aslam, M. I., & Ahmed, I. (2017). Energy efficient image encryption algorithm. In *2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT)*.

Performance Evaluation of Ensemble-Based Machine Learning Techniques for Prediction of Chronic Kidney Disease



K. M. Zubair Hasan and Md. Zahid Hasan

Abstract Chronic kidney disease (CKD) is widespread and related with enhanced risk of cardiovascular disease and end-stage renal disease, which are possibly escapable through early detection and treatment of individuals at risk. Machine learning algorithm helps medical experts to diagnose the disease correctly in the earlier stage. Therefore, machine-predicted analysis has become very popular in recent decades that can efficiently recognize whether a patient has certain kidney disease or not. In this regard, we propose an ensemble method based classifier to improve the decision of the classifiers for kidney disease diagnosis efficiently. Ensemble methods combine multiple learning algorithms to achieve better predictive performance than could be obtained from any of the constituent learning algorithms. In addition, Data is evaluated by using tenfold cross-validation and performance of the system is assessed on receiver operative characteristic curve. Extensive experiments on CKD datasets from the UCI machine learning repository show that our ensemble-based model achieves the state-of-the-art performance.

Keywords Machine learning · Classification · Chronic kidney disease (CKD) · Ensemble method · Data mining · Healthcare informatics

1 Introduction

According to the report of the Centers for Disease Control and Prevention (CDC), kidney disease causes millions to die each year. A good number of people in the world with kidney damage and slightly reduced kidney function are not conscious of having CKD. Therefore, strategies for early detection and cure of people with CKD are consequently required worldwide. A computer-aided diagnosis system based on sophisticated machine learning techniques is required for healthcare data to mine

K. M. Zubair Hasan · Md. Zahid Hasan (✉)
Daffodil International University, 102 Sukrabadh, Mirpur Road, Dhaka 1207, Bangladesh
e-mail: zahid.cse@diu.edu.bd

K. M. Zubair Hasan
e-mail: kmzubair.hasan@gmail.com

© Springer Nature Singapore Pte Ltd. 2019
N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_34

hidden pattern from data for effective decision-making. The purpose of our current study is to evaluate the performance of several ensemble-based machine learning techniques correctly classifying CKD patients based on clinical datasets. We consider five machine learning classifiers, namely Adaptive Boosting, Bootstrap Aggregating, Extra Trees, Gradient Boosting, and Random Forest Classifier. Finally, a large set of standard performance metrics is used to design the computer-aided diagnosis system for estimating the performance of each machine learning and artificial intelligence classifier. The metrics we used include Classification Accuracy, Sensitivity, Precision, Specificity, Negative Predictive Value, False Positive Rate, False Negative Rate, F1-Score, and Error Rate of Classification.

The paper is organized as follows. Section 2 discusses the related works. Section 3 introduces the materials and methods. Section 4 describes the proposed ensemble-based machine learning techniques applied to the dataset. Section 5 presents the classification performance matrices. These comparisons' study is discussed in Sect. 6. Finally, we concluded our work in Sect. 7.

2 Related Work

Ho et al. [1] presented a computer-aid diagnosis tool for CKD classification analyzing ultrasonography images. This system used for detecting and classifying distinctive various stages of CKD. The K-means clustering was performed for detecting regions in an ultrasonic image as preprocessing step. Estudillo-Valderrama et al. [2] proposed the feasibility study of using a distributed system for the management of alarms from chronic kidney disease patients inside Enefro project. Charleonnan et al. [3], explored four machine learning approaches including K-nearest neighbors (KNN), support vector machine (SVM), logistic regression (LR), and decision tree classifiers for predicting the chronic kidney disease. In Hsieh et al. [4] showed a real-time system to consider chronic kidney disease by using only ultrasound images. Support vector machine technique is used to predict and classify CKD stages from the ultrasound images. The authors in [5] recommended diverse methods to influence the hierarchical structure in ICD-9 codes to develop the performance of predictive models. A novel feature engineering approach is proposed in this study to influence this hierarchy, while immediately diminishing feature dimensionality. An intelligent system based on artificial neural networks in Chiu et al. [6] for detecting chronic kidney disease for evaluating the extremity of a patient. Three types of artificial neural networks have been used in this model including backpropagation network (BPN), modular neural network (MNN), and generalized feedforward neural networks (GRNN).

3 Materials and Methods

Our research uses a CKD dataset [7], which is openly accessible at UCI machine learning laboratory. This dataset comprises 400 instances with 150 samples without kidney disease (not presence) and 250 samples with kidney disease (presence). Missing values in the dataset may effect in the accuracy of disease prediction. So to avoid this, we consider 158 without missing samples and then apply Extra Tress algorithm for evaluating feature importance. After getting feature importance, we compute impute (mean) for missing values of numerical type attributes for increasing the number of samples as preprocessing steps. Thenceforth, for predicting kidney

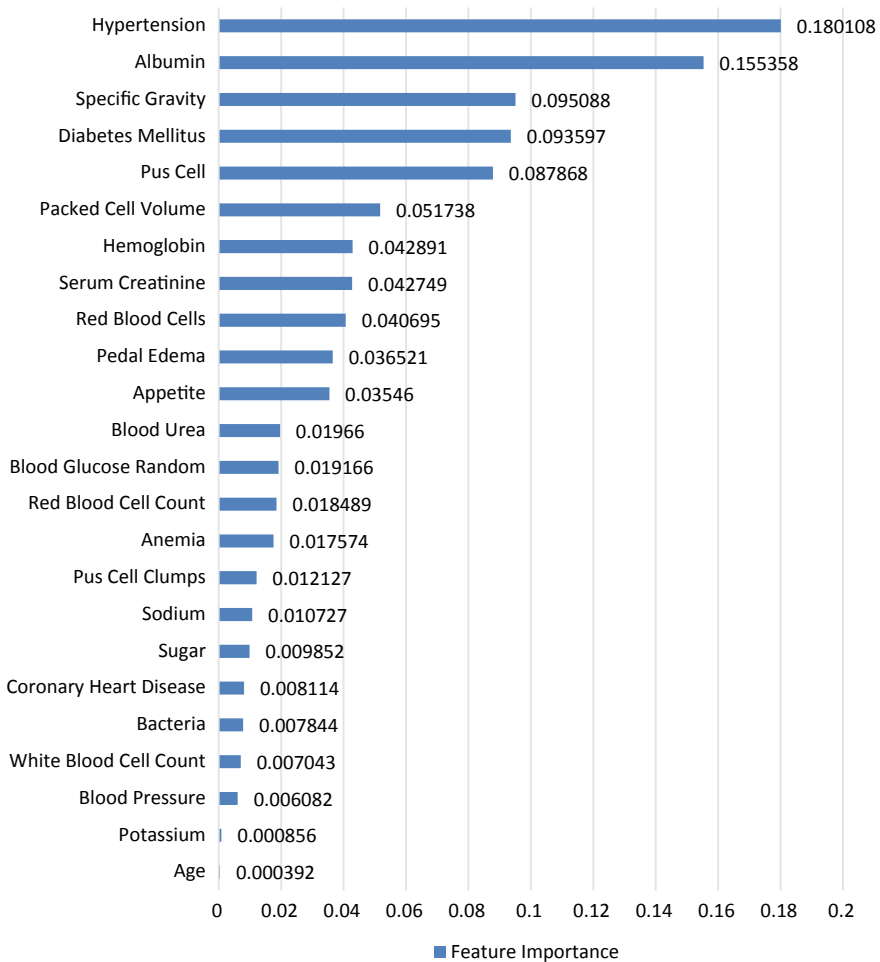


Fig. 1 Importance of attributes in CKD datasets

diseases, 13 distinct parameters (Hypertension, Albumin, Specific Gravity, Diabetes Mellitus, Pus Cell, Packed Cell Volume, Haemoglobin, Serum Creatinine, Red Blood Cells, Pedal Edema, Appetite, Blood Urea, and Blood Glucose Random) have been taken out of 24 attributes into account considering the feature importance. Figure 1 shows the importance of attributes in CKD dataset.

4 Machine Learning Techniques

4.1 Adaptive Boosting (AdaBoost)

Adaptive Boosting is an ensemble machine learning meta-algorithm technique that creates a robust classifier from a number of weak classifiers. For improving performance, it incorporated extra copies of the classifier on the same dataset, however where the weights of incorrectly classified samples are adjusted and adjusts them to represents the final output of the boosted classifier.

Given M training data $\{(x_1, y_1), \dots, (x_M, y_M)\}$, x_i is a vector corresponding to an input sample data, associated with Q input attributes, and y_i is a target variable with a class label of either -1 or $+1$ and a set of weak learners $\{K_1, \dots, K_L\}$ each of which outputs a classification $K_j(x_j) \in \{-1, 1\}$ for each classifier. An initial weight is set for $1/n$. Equal weight is assigned to all the instances in the kidney datasets. AdaBoost is explained using pseudocode [8].

For t in $1, \dots, T$:

- Choose $h_t(x) \rightarrow [-1, 1]$:
 - Find weak learners that minimize ϵ_t , the summation of errors in weights for misclassified points

$$\epsilon_t = \sum_{\substack{i=1 \\ h_t(x_i \neq y_i)}}^n w_{i,t}$$

- Choose $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
- Add to ensemble $F_t(x) + \alpha_t h_t(x)$
- Update weights for $i = 1, \dots, m$ using

$$w_{t+1}(i) = \frac{w_t(i) e^{-\alpha_t y_i h_t(x_i)}}{Z_t}$$

Where Z_t is a normalization factor.

- Final hypothesis:

$$H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right)$$

H is determined as a weight majority vote of the weak hypothesis h_t , where each classifier is assigned weight α_t .

4.2 Bootstrap Aggregating (Bagging)

Bagging is another ensemble meta-algorithm used in statistical classification for reducing variance and supports to avoid overfitting. At first, an initial weight is set for kidney datasets [9].

For every classifier as T rounds:

- Normalize the weights
- Train the classifier and evaluate training error
- Choose: lowest error classifier
- Update the weights of the training data

Then, the final classifier is formed as the linear combination of T classifiers.

4.3 Extra Trees Classifier

Extra Trees is another modification of bagging classifier with ordinary tree-based techniques as far as accuracy and computational efficiency. The main differences with other tree-based algorithm are that it can split the node by choosing cut-points randomly and build the trees using the total learning samples [10].

Split_node(S)

Input: Spited node equivalent to local subset S equivalent

Output: a split where a_c is greater than a or not anything

- If **End_split(S)** is true then it will not return anything
- Or else, Choose k attributes $\{a_1, \dots, a_k\}$ among all nonconstant (in S) candidate attributes.
- Draw k splits $\{s_1, \dots, s_k\}$, where $s_i = \text{Choice_random_split}(s, a_i), \forall_i = 1, \dots, k$;
- Return a split s_* such that s will be selected from the maximum score of S .

Choice_random_split(S,a)

Input: Local subset S and an attribute a .

Output: a split

- Draw a random cut-point a_C considering maximal attributes of subset S and minimal attributes of subset S
- Return the split based on $[a < a_C]$

End_split(S)*Input:* Local subset S *Output:* a Boolean value

- If the absolute value of S is less than the minimum sample size of the splitting node, then return TRUE;
- If every property is consistent at that point then return TRUE;
- If the outcome is steady in S , then return TRUE;
- Or else, return FALSE.

4.4 Gradient Boosting

Gradient boosting is an optimization machine learning algorithm on a suitable cost function for the prediction of kidney disease classification problem. In the process of gradient boosting, a series of predictor values are iteratively formed. The final predictor value is generated after iteratively calculating the weighted average values of the weak predictor. At every sequence, an extra classifier is invoked to boost the performance of the complete ensemble [11]. The algorithm for the predictive model is enlisted in Table 2.

Input: A training set of data points (x_i, y_i) from the given dataset*Output:* A classification tree

- Initialization: Initialize model with a constant low value of T classifier.
- **For** i to number of iterations M **do**
- Calculate new weights of (x_0, y_0) to (x_i, y_i) with minimal prediction accuracy rate of T .
- Draw new weak classifier h_i on the pre-weighted prediction accuracy of T .
- Calculate weight β_i of new classifier.
- Update the model as $h_i + \beta_i$.
- **end for**
- **return** T

Gradient boosting adds weak learners to minimize the bias in addition to the variance to some degree, thus reducing the error [12].

4.5 Random Forest

Random Forest (RF) is a variant of ensemble classifier consisting of a collection of tree-structured classifiers $h(x, y_k)$, which is defined as multiple tree predictors y_k such that each tree relies upon the estimations of an arbitrary vector inspected independently and with a similar distribution for all trees in the forest. The randomization works in two ways: an arbitrary sampling of data for bootstrap samples as it

is done in Bootstrap aggregating (bagging) in Sect. 4.2 and random selection of input attributes for producing individual base decision trees [13]. Random forests become different in a way from other methods that a modified tree learning algorithm is utilized that chooses the differentiable candidate in the learning procedure, a random subset of the features. The cause for doing this is the relationship of the trees in a standard bootstrap sample. For example, if one or a couple of features are extreme indicators for the response variable (target output), these features will be chosen in a considerable lot of the decision trees, reasoning them to end up correlated.

5 Classification Performance Measurements

Five ensemble-based supervised machine learning techniques were applied for the classification of chronic kidney disease attributes. Classification performance was estimated by tenfold cross-validation. The classification models were assessed on the nine quality measures [14]. These quality measures for the analysis of classification were examined closely. Absence attributes of chronic kidney disease were measured as non-ckd class, and attributes with the presence of chronic kidney disease were measured as ckd class. Here,

True positive (TP)—Correct positive prediction where samples with ckd predicted as ckd.

False positive (FP)—Incorrect positive prediction where samples with non-ckd predicted as ckd.

True negative (TN)—Correct negative prediction where samples with non-ckd predicted as non-ckd.

False negative (FN)—Incorrect negative prediction where samples with ckd predicted as non-ckd.

The quality measures are executed to evaluate the efficiency of each machine learning classifier in the distinction between presence and absence with samples of chronic kidney disease. They are expressed as follows:

$$\text{Classification Accuracy} = \frac{(TN+TP)}{(TN+FP+FN+TP)}$$

$$\text{Sensitivity} = \frac{TP}{(TP+FN)}$$

$$\text{Precision} = \frac{TP}{(TP+FP)}$$

$$\text{Specificity} = \frac{TN}{(TN+FP)}$$

$$\text{Negative Predictive Value} = \frac{TN}{(TN+FN)}$$

$$\text{False Positive Rate} = \frac{FP}{(FP+TN)}$$

$$\text{False Negative Rate} = \frac{FN}{(FN+TP)}$$

$$\text{Rate of Misclassification} = \frac{(FN+FP)}{(TN+FP+FN+TP)}$$

$$\text{F1 Score} = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

6 Result and Discussion

The efficiency of five ensemble-based machine learning approaches was accounted that all classifiers are trained with all 13 classes discussed in methods and materials section, following a tenfold cross-validation to illustrate statistically robust results for the prediction of kidney disease. We employed the Python software to execute all classification tasks. Predicted results based on the confusion matrix is arranged in Tables 1, 2, and 3 for Adaptive Boosting (AdaBoost), Bootstrap Aggregating (Bagging), Extra Trees, Gradient Boosting, and Random Forest, respectively.

Figure 2 shows the classification results of these ensemble machine learning algorithms. It is obvious from the outcomes that Bootstrap Aggregating (Bagging) predicts most noteworthy of true positives (presence of kidney disease classified as presence) (Table 2; Fig. 2) and combined Adaptive Boosting (AdaBoost) and Extra Trees algorithm predicts the most noticeable number of true negatives (absence of kidney disease classified as absence of kidney disease) (Tables 1 and 3; Fig. 2). Adaptive and gradient boosting confusion matrix (Tables 1 and 2) indicates that it has second-best true positives (Fig. 2). Tables 2 and 3 show the confusion matrix of Extra Trees and Random Forest classifier, which indicates that this classifier gives the third most elevated number of true positives (Fig. 2).

Table 1 Confusion matrix for tenfold cross-validation using adaptive boosting and bootstrap aggregating

Adaptive boosting				Bootstrap aggregating			
	Predicted class				Predicted class		
	ckd	Non-ckd	Actual total		ckd	Non-ckd	Actual total
Actual class				Actual class			
ckd	61	1	62	ckd	62	0	62
Non-ckd	0	38	38	Non-ckd	4	34	38
Total predicted	61	39	100	Total predicted	66	34	100

Table 2 Confusion matrix for tenfold cross-validation using extra trees and gradient boosting

Extra trees				Gradient boosting			
	Predicted class				Predicted class		
	ckd	Non-ckd	Actual total		ckd	Non-ckd	Actual total
Actual class				Actual class			
ckd	60	2	62	ckd	61	1	62
Non-ckd	0	38	38	Non-ckd	2	36	38
Total predicted	60	40	100	Total predicted	63	37	100

Table 3 Confusion matrix for tenfold cross-validation using random forest

Random forest			
		Predictive class	
	CKD	Non-CKD	Actual total
Actual class			
ckd	60	2	62
Non-ckd	3	35	38
Total predicted	63	37	100

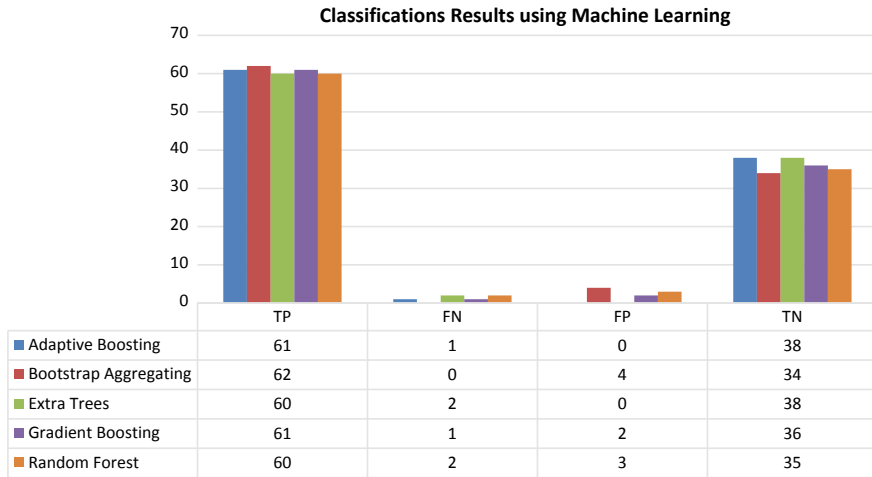


Fig. 2 Classification results of ensemble machine learning techniques

Figure 3 plots nine quality measures for classification performance measurements. Figure 3 specifies that Adaptive Boosting beats over all other machine learning algorithms with the most extreme classification accuracy of 99% while the second most noteworthy characterization exactness is accomplished by Extra Trees 98%. Furthermore, these two strategies have indicated the most extreme specificity and precision of 100% that this classifier is most appropriate for distinguishing proof of patients with chronic kidney disease (absence class). Bootstrap Aggregating achieves the highest sensitivity, the negative predictive value of 100% and the lowest false negative rate of zero percent, which indicates that this classifier is most appropriate for identification of people who are sick with chronic kidney disease (presence class). But, bagging also has the lowest specificity of 89% and false negative rate of 11% that is not fit for prediction of sick people with chronic kidney disease (absence class). Figure 3 associates F1 Score and Error Rate of Classification Adaptive Boosting shows the highest 99% and the lowest 1%, respectively.

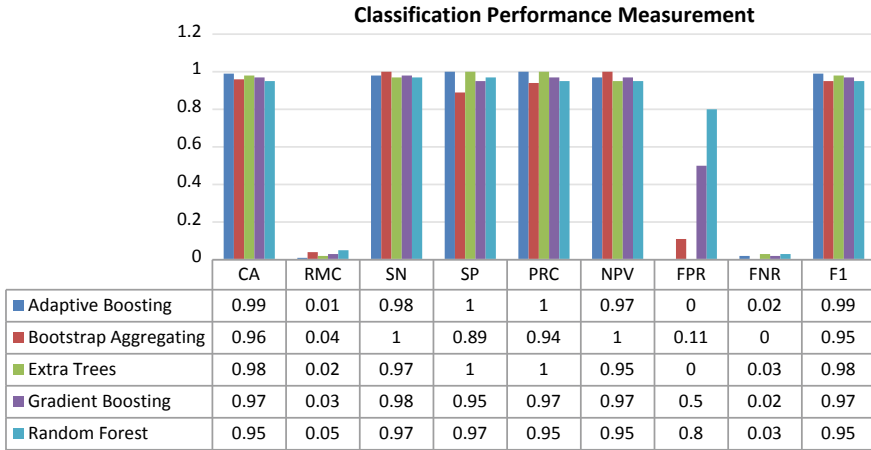


Fig. 3 Classification performance of ensemble machine learning techniques

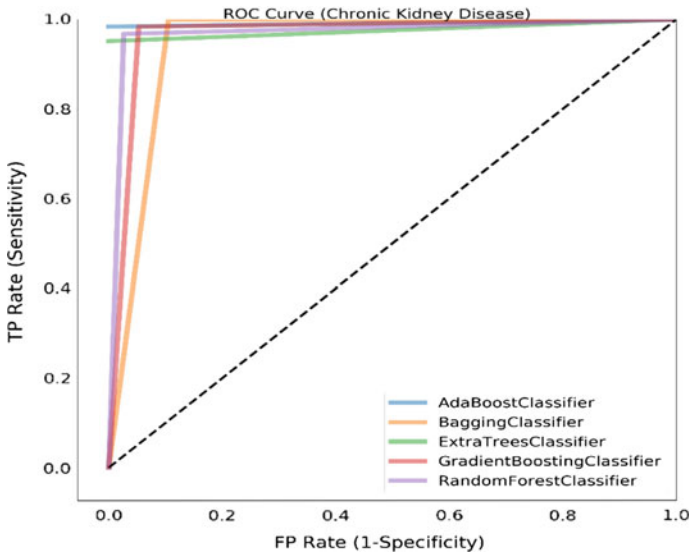


Fig. 4 ROC for ensemble machine learning techniques

In agreement to the previously mentioned assessment criteria, a receiver operating characteristic (ROC) curve [15] is used and the area under the curve (AUC) to evaluate the advantage and disadvantage of the classifier. ROC is unbiased of the two classes and significant when the number instances of the two classes change through training [16]. The region under ROC curve must be close to 1 for the best classifier. Figure 4 shows that AdaBoost classifier beats every single other procedure in expectation of

quality of kidney ailment and other ensemble-based learning techniques perform the approximate similar result in the prediction of the essence of kidney disease.

7 Conclusion

This prediction of kidney disease may spare the life of people and can have a real effect on its cure. In this paper, we propose an ensemble method based machine learning algorithm to improve the performance of the classifier for kidney disease. In contrast with quite a few classic prediction algorithms, the classification accuracy of our proposed ensemble learning algorithm achieves 99% classification accuracy.

References

1. Ho, C.-Y., Pai, T.-W., & Peng, Y.-C. Ultrasonography image analysis for detection and classification of chronic kidney disease. In *Sixth International Conference on Complex, Intelligent, and Software Intensive Systems* (pp. 624–629), Palermo.
2. Estudillo-Valderrama, M. A., et al. (2014, November). A distributed approach to alarm management in chronic kidney disease. *IEEE Journal of Biomedical and Health Informatics*, 18(6), 1796–1803.
3. Charleonnann, A., Fufaung, T., Niyomwong, T., Chokchueyattanakit, W., Suwannawach, S., & Ninchawee, N. (2016). Predictive analytics for chronic kidney disease using machine learning techniques. In *Management and Innovation Technology International Conference (MITicon)* (pp. 80–83), Bang-San.
4. Hsieh, J.-W., Lee, C.-H., Chen, Y.-C., Lee, W.-S., & Chiang, H.-F. (2014). Stage classification in chronic kidney disease by ultrasound image. In *International Conference on Image and Vision Computing*, New Zealand, ACM, 2014 (pp. 271–276).
5. Singh, A., Nadkarni, G., Gutttag, J., & Bottinger, E. (2014). Leveraging hierarchy in medical codes for predictive modeling. In *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics (BCB '14)* (pp. 96–103). ACM, New York, NY, USA, 2014.
6. Chiu, R. K., Chen, R. Y., Wang, S.-A., & Jian, S.-J. (2012). Intelligent systems on the cloud for the early detection of chronic kidney disease. In *International Conference on Machine Learning and Cybernetics* (pp. 1737–1742), Xian, China.
7. Asuncion, A., & Newman, D. J. UCI Machine Learning Repository [Online]. Available <http://www.ics.uci.edu/~mllearn/MLRepository.html>. (2007).
8. Rojas, R. (2009). AdaBoost and the super bowl of classifiers a tutorial introduction to adaptive boosting. Freie University, Berlin, Technical Report.
9. Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2), 123–140.
10. Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1), 3–42.
11. Manna, S., Biswas, S., Kundu, R., Rakshit, S., Gupta, P., & Barman, S. (2017). A statistical approach to predict flight delay using gradient boosted decision tree. In *International Conference on Computational Intelligence in Data Science (ICCIDS)* (pp. 1–5), Chennai, India.
12. Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4), 367–378.

13. Anbarasi, M. S., & Janani, V. (2017). Ensemble classifier with Random Forest algorithm to deal with imbalanced healthcare data. In *International Conference on Information Communication and Embedded Systems (ICICES)* (pp. 1–7), Chennai.
14. Kanmani, S., Uthariaraj, V. R., Sankaranarayanan, V., & Thambidurai, P. (2007). Object-oriented software fault prediction using neural networks. *Information and Software Technology*, 49(5), 483–492.
15. Metz, C. E. (1978). Basic principles of ROC analysis. In L. M. Freeman (Ed.), *Seminars in nuclear medicine* (Vol. 4, pp. 283–298). Amsterdam: Elsevier.
16. Dwivedi, A. K. (2016). Performance evaluation of different machine learning techniques for prediction of heart disease. *Neural Computing and Applications*, pp. 1–9.

Proficient Cooperative Caching in SWNET Using Twin Segments Approach



**B. N. Lakshmi Narayan, Prasad N. Hamsavath, Meher Taj, E. G. Satish,
Vivek Bharadwaj and S. Rabendranath**

Abstract Web caching is the main practice to scale the Internet. One chief performance aspect of web caches is the replacement strategy. The most prominent system to let an array of distributed caches to cooperate and serve each other using web requests is cooperative caching. In the existing cooperative caching schemes, most of them will not provide high network level availability and high node level availability at the dupe time. The feasibility of the object is ensured by network level in isolated network partitions and node level in individual nodes when they are completely detached from all of the networks. To reduce electronic content provisioning cost in Social Wireless Network (SWNET), it is recommended to use cooperative caching policies. Social Wireless Networks are designed by mobile devices like Android, iPhone, Kindle, etc., giving out similar interests in electronic information, and physically congregating in public areas. In Social Wireless Network, object caching proves that it reduces the content provisioning cost, which is massively dependent on the pricing factors within several stakeholders like End Consumers, Content Service Providers, and Network Service Providers.

Keywords SWNET · Twofold · Ad hoc network

B. N. Lakshmi Narayan (✉) · P. N. Hamsavath
Department of MCA, NMIT, Bengaluru, India
e-mail: narayan@nmit.ac.in

P. N. Hamsavath
e-mail: prasad.nh@nmit.ac.in

M. Taj
Department of Computer Science, GSSS FGC, Mysore, India

E. G. Satish
Department of CSE, NMIT, Bangalore, India

V. Bharadwaj · S. Rabendranath
Department of CSE, AMCEC, Bengaluru, India
e-mail: hn.vivekbharadwaj@gmail.com

S. Rabendranath
e-mail: raben_s@yahoo.com

1 Introduction

The establishment of mobile ad hoc networks (MANETs) is done with mobile hosts (MHs), such as notebook, PDAs, and so on. Without the assistance of any network infrastructure, these mobile devices can form a wireless network dynamically. The use of multi-hop wireless links helps to move every MH arbitrarily and communicate with one another. Within the scope of transmission area, the MH which acts as a router, forwards the data packets to other neighbors. MANET is very useful under certain environments, such as battles, disaster rescue, earthquake recovery, etc. SWNETs can be framed with the help of ad hoc wireless connections between the devices. In the existence of such SWNETs, a distinct way for content access by a device would be to first look into the local SWNET for the requested content before downloading from the Content Service Provider's server. The expected content provisioning cost of such an approach can be considered lesser, because the download from Content Service Provider would be avoided since the content is available within the local SWNET. This advent is named as cooperative caching.

Caching techniques is one of the optimum ways to trim the number of requests made to the server, which results in increasing the performance communication between the data. Search Engines, Web Browsers, Content Delivery Networks, and Web Proxies are some systems which widely cache web files. On the web, cache placement and replacement in proxy servers reduce the average amount of time taken for data query and the network traffic. The primary purpose of caching is to improve the performance of data communication. The data accessed by mobile hosts will reside in spatial and temporal memory. More the space in temporal and spatial memory, most of the accesses will go to the data that were recently accessed. Hence, the performance of data communication can be improved by caching frequently requested data.

Due to a smaller amount of memory space available in mobile devices all downloaded contents cannot be kept for a longer duration. For this reason, the device will delete cached data after downloading the content from the storage. So introducing a rebate mechanism to encourage the users to store the content as long as they can. This makes the popular content available for a long time and reduces the content provisioning cost.

2 Related Work

2.1 Cache Data

In this approach, recently and frequently accessed objects will be saved in the intermediate nodes. The objects will be saved based on some predefined criteria. By this way, the requests can be served by the intermediate nodes. But, storing a huge number of objects in the intermediate nodes is the major drawback of this approach and is not scalable.

2.2 Cache Path

In this approach, instead of storing the recently and frequently accessed objects as done in CacheData approach, the intermediate nodes store only the object paths of the nearest node where the objects are available. The primary idea of this approach is to reduce the average time taken to find the requested object. But, in a huge mobile network, this strategy will fail due to the replacement of new object paths frequently.

2.3 Hybrid Cache

In this approach, both the CacheData and CachePath approach will be used interchangeably based on the traversing objects through the intermediate node.

3 Optimized Object Placing for Caching

3.1 Replacement Using Twin Segments Cache

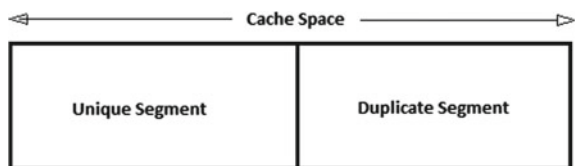
To put forward the optimal object placement we come up with the Twin Segment cache method in which the available cache space in each device is categorized into two segments. First is a Unique segment and second is a Duplicate segment (see Fig. 1).

As the name indicates, Unique segment can save only distinct objects and the Duplicate segment can save only the most frequently accessed objects without caring about the object replication.

In the TwinSegment Caching replacement approach, immediately posterior an object is downloaded from the Content Provider’s server, it will be considered as a rare object since there is only a single copy of this object in the network. When a node downloads the same object from one more SWNET node, that object will be considered as a duplicate object since there are twin replicate copies of that object in the network.

The path taken to place a new unique object is based on popularity. The comparison is done between the new object and the less popular object in the complete cache to

Fig. 1 Cache partition



find which one is less popular. If the selected object in the cache is less popular than the new object then it will be replaced with the new object.

On the other hand for a duplicate object, the candidate is selected only from the first duplicate segment of the cache.

To simplify, a unique object will never be ignored to save just because to save a duplicated object. The TwinSegment Cache ensures that the optimal object replacement mechanism is achieved. By making use of this approach, all mobile devices in their Duplicate segments preserve the same objects, but distinct objects in their Unique segments. In this way, popular contents will be obtainable for a longer duration in the network either from Unique segmented cache or Duplicate Segmented cache. This reduces the content provisioning cost when the content is locally available within a network. The TwinSegment cache makes it possible to keep track of popular content, which needs to be kept for a long time.

```

If (Op is downloaded directly || f == true
    Op. benefit = Vp + Uip
    Op. l = key
Else
    Op. benefit = Uip
    Op. l = derived
End
Om = Obj with low benefit
If (Op. benefit > Om. benefit)
    replace Om with Op
    send the change of status message
end
Algorithm 2: optimized caching policy
    
```

```

INPUT: Obj Oe
If (On is newly downloaded)
    Om = low popular Obj in replicate area
Else
    Om = low popular Obj in whole cache
End
If (On popularity > Om popularity)
    replace Om with Oe
Algorithm 1: TwinSegment cache object replacement
    
```

3.2 Object Provisioning Cost Along with Split Cache

Local and remote hit rate helps to know the object provisioning cost.

- Local hit rate is the folio of exclusive objects stored with a probability that a device can find the requested object.
- Remote hit rate is equal to the hit probability contributed by the objects stored in the unique area of all devices in the partition, minus the unique area of the local cache.

4 Optimized Caching Policy

The caching strategy based on content importance, presented in this section, when there is less memory in the cache for preserving a new object, the object which is already present in the cache with the less importance is identified and replaced with the new object which has more total importance. Based on the source, the importance of a newly downloaded object is calculated. When an object “p” is directly downloaded from the Content Provider’s server, the copy is labeled as key. On the other hand, the copy is labeled as derived, if the object is downloaded from another node in the SWNET partition. The new object is stored in the cache only if its importance is greater compared to any of the existing cached objects. Along with the object replacement based on content importance policy approach as exhibited in the previous sections, provisioning cost reduction needs that a key object should be cached within the node that is most possibly to produce requests for that object.

```

INPUT: Op
      Flag=f
If (Op is downloaded directly || f == true
  Op. benefit = Vp + Uip
  Op. l = key
Else
  Op. benefit = Uip
  Op. l = derived
End
Om = Obj with low benefit
If (Op. benefit > Om. benefit)
  replace Om with Op
  send the change of status message
end
Algorithm 2: optimized caching policy
    
```

5 Caching Based on Content Benefit with Cost and Rebate

The global popularity of objects with high content demand can be represented as

$$\text{Global popularity} = \frac{\text{total requests for object 'j'}}{\text{total request in network}}$$

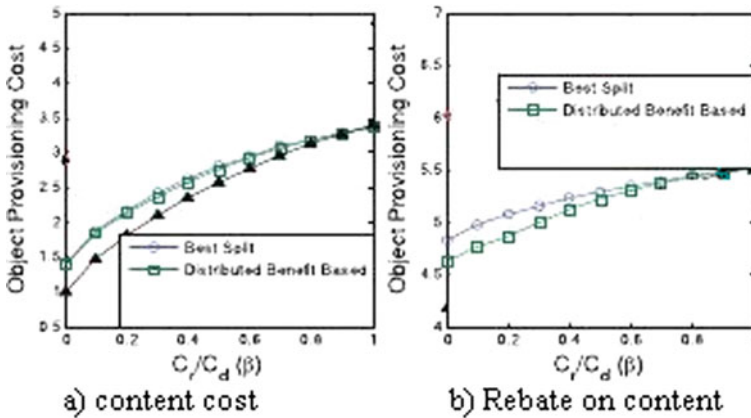
The local popularity of an object can be represented as

$$\text{Local popularity} = \frac{\text{total requests from node 'a' for object 'j'}}{\text{total request from 'a'}}$$

In comparison with other nodes, most of the time only few nodes will be extra active and produce extra requests per unit time. Individual node-definite request rates can have a considerable impact on the average object provisioning cost. Due to this

reason, it is important to consider this parameter in object placement algorithm as presented in Algorithm 2 for the content importance based strategy.

Because of mixed nature enfold proxy can never be a good option for deciding the optimal solution in TwinSegment Cache. The other best option to decide on the optimal solution for TwinSegment Cache would be to experimentally run the protocol for all possible values and then choosing the one that output the minimum cost. This minimum cost is shown as the best optimal TwinSegment cache.



6 Conclusion

The aspiration of work done is to come up with a cooperative caching strategy for provisioning cost minimization in Social Wireless Networks. The most excellent cooperative caching approach for provisioning cost reduction in requires the best way categorization between distinct and replicated objects. Hereby, we propose a Twin-Segment replacement which is evaluated for android mobile phones. It is proven that with the approach based on content importance, it provides improved performance in comparison with TwinSegment cache which is proposed mainly for content demand.

References

1. Zhao, M., Mason, L., & Wang, W. (2008). Empirical study on human mobility for mobile wireless networks. In *Proceedings of IEEE Military Communications Conference (MILCOM)*.
2. Cambridge Trace File, Human Interaction Study. (2012). <http://www.crawdad.org/download/cambridge/haggle/Exp6.tar.gz>.
3. Cohen, E., Krishnamurthy, B., & Rexford, J. (1998). Evaluating server-assisted cache replacement in the web. In *Proceedings of the Sixth Annual European Symposium Algorithms* (pp. 307–319).

4. Banerjee, S., & Karforma, S. (2008). A prototype design for DRM based credit card transaction in e-commerce. *Ubiquity*, 2008.
5. Breslau, L., Cao, P., Fan, L., & Shenker, S. (1999). Web caching and Zipf-like distributions: Evidence and implications. In *Proceedings of IEEE INFOCOM*.
6. Perkins, C., & Royer, E. (1999). Ad-Hoc on-demand distance vector routing. In *Proceedings of IEEE Second Workshop Mobile Systems and Applications*.
7. Podlipnig, S., & Boszormenyi, L. (2003). A survey of web cache replacement strategies. *ACM Computing Surveys*, 35, 374–398.
8. Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R., & Scott, J. (2007). Impact of human mobility on opportunistic forwarding algorithms. *IEEE Transactions on Mobile Computing*, 6(6), 606–620.
9. BU-Web-Client—Six Months of Web Client Traces. (2012). <http://www.cs.bu.edu/techreports/1999-011-usertrace-98.gz>.
10. Wolman, A., Voelker, M., Karlin, A., & Levy, H. (1999). On the scale and performance of cooperative web caching. In *Proceedings of 17th ACM Symposium Operating Systems Principles* (pp. 16–31).
11. Dykes, S., & Robbins, K. (2001). A viability analysis of cooperative proxy caching. In *Proceedings of IEEE INFOCOM*.
12. Korupolu, M., & Dahlin, M. (2002). Coordinated placement and replacement for large-scale distributed caches. *IEEE Transactions on Knowledge and Data Engineering*, 14(6), 1317–1329.

A Particle Swarm Optimization-Backpropagation (PSO-BP) Model for the Prediction of Earthquake in Japan



Abey Abraham and V. Rohini

Abstract Japan is a country that suffers a lot of earthquakes and disasters because it lies across four major tectonic plates. Subduction zones at the Japanese island curves are geologically complex and create various earthquakes from various sources. Earthquake prediction helps in evacuating areas, which are suspected and could save the lives of people. Artificial neural network is a computing model inspired by biological neurons, which learn from examples and can be able to do predictions. In this paper, we present an artificial neural network with PSO-BP model for the prediction of an earthquake in Japan. In PSO-BP model, particle swarm optimization method is used to optimize the input parameters of backpropagation neural network. Information regarding all major, minor and aftershock earthquake is taken into account for the input of backpropagation neural network. These parameters are taken from Japan seismic catalogue provided by USGS (United States Geological Survey) such as latitude, longitude, magnitude, depth, etc., of earthquake.

Keywords Tectonic plates · Artificial neural network · Particle swarm optimization · Backpropagation · Seismic catalogue

1 Introduction

The earthquake formation is one of the hectic geological disasters that occur on the surface of earth having various parameters associated with it. It causes unrecoverable life harm. Seismic changes, changing in the water's temperature, etc., are the factors that are associated with earthquake [1]. The effect of this natural disaster is high because it happens all of a sudden and unpredictable. High death rates, property damages, etc., can be caused by an earthquake. Basic living conditions, for instance, water, sanitation, vitality, correspondence, transportation, and so forth is influenced by this seismic disaster. Earthquakes smash urban communities and towns, and in addition, the effects provoke the destabilization of the money related and the com-

A. Abraham (✉) · V. Rohini
Department of Computer Science, CHRIST (Deemed to Be University), Bengaluru 560029, India
e-mail: abey.abraham@mca.christuniversity.in

© Springer Nature Singapore Pte Ltd. 2019
N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_36

435

munal surface of the nation. The impact of the event is awful in light of the way that it influences a far-reaching extent to happen all of a sudden and unpredictable [2]. Prediction of the place of event, force, and epicentral zone of future major quakes has been the subject of different legitimate undertakings with especially differentiating conclusions in the late years. There are persuaded technique which rely upon either the examination of precursory phenomena beforehand tremors, for instance, seismic peacefulness, changes in attractive and electric signals recorded at seismic goals, and sporadic creature conduct.

Artificial Neural Network (ANN) is interconnected neurons with duplication of some properties of natural neurons [3]. The neural network structure is an extraordinarily self-adaptable nonlinear progression. In this work, neural network models are shown for foreseeing the magnitude of the greatest seismic event in the following year in perspective of the examination of eight scientifically characterized seismicity factors. For a given seismic region, eight seismicity pointers computed from a pre-described number of vital seismic events are passed as contributions to anticipate the occurrence or non-occurrence of an earthquake. There has been work done as using other neural network algorithms such as probabilistic neural network and recurrent neural network for the prediction of the earthquake [4]. Artificial neural network is a computational model inspired on the biological neurons in human brain. The artificial neural network contains interconnected neurons, which are highly self-adaptable and can be used in various problem-solving scenarios [5].

2 Study Area

Pacific plate, North American plate, Eurasia plate, and the Philippine Sea plate are the four major tectonic plates in which Japan and its islands inhabit. The Pacific plate is forced sideways and downwards into the mantle, beneath Hokkaido and northern Honshu, along the eastern margin of the Okhotsk microplate. Toward south, the Pacific plate is forced sideways and downwards into the mantle beneath volcanic islands along the eastern margin of the Philippine Sea plate. Creation of deep offshore Ogasawara and Japan trenches is due to the 2200 km long zone force of sideways and downwards into the mantle by pacific plate. Similarly, the Philippine Sea plate is itself forced sideways and downwards into the mantle under the Eurasia plate. Subduction zones at japan area with respect to geology are too complex. They produce a lot of earthquakes from multiple sources of origin (Fig. 1).

3 Particle Swarm Optimization

Particle swarm optimization is an optimization technique based on the social behaviour of bird flocking. Particle swarm optimization technique has similarities with genetic algorithm. System starts with population of random solutions and finds

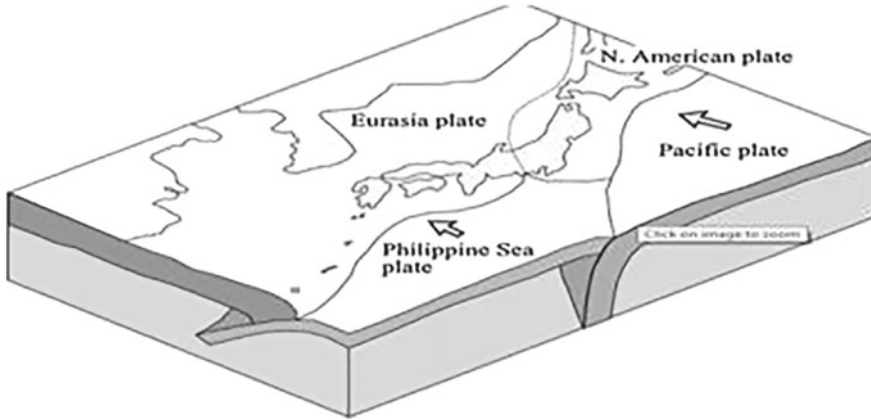


Fig. 1 Tectonic plates in Japan

the optimal solution by updating generations. Every particle stores coordinates in particle with which best solution is found so far [6]. The values are changed from iteration to iteration. To find the optimal solution, with respect to the previously best position (pbest) and global best (gbest) position every particle moves.

$$\begin{aligned}
 \text{pbest}(i, t) &= \arg_{k=1, \dots, t} \min[f(p_i(k))], \quad i \in \{1, 2, \dots, N_p\}, \\
 \text{gbest}(t) &= \arg_{k=1, \dots, t} \min_{i=1, \dots, N_p} [f(P_i(k))]
 \end{aligned}
 \tag{1}$$

where i denotes the index of the particle, N_p is the total number of particles, p is the position, t is the current iteration and f is the fitness function. The position P and velocity V is updated by following equations:

$$\begin{aligned}
 V_i(t + 1) &= \omega V_i(t) + c_1 r_1 (\text{pbest}(i, t) - P_i(t)) + c_2 r_2 (\text{gbest}(t) - P_i(t)), \\
 P_i(t + 1) &= P_i(t) + V_i(t + 1)
 \end{aligned}
 \tag{2}$$

where V is the velocity, w is the inertia weight, c_1 and c_2 are the acceleration coefficients.

4 Backpropagation Neural Network

The backpropagation algorithm is used as a piece of layered feedforward Artificial Neural Networks. Backpropagation has multiple layers with a supervised learning framework in light of gradient descent learning rule [7]. We give the algorithm instances of the sources of inputs and for outputs we require the framework to enlist, and the error generated afterward is found out. The likelihood of this algorithm is to

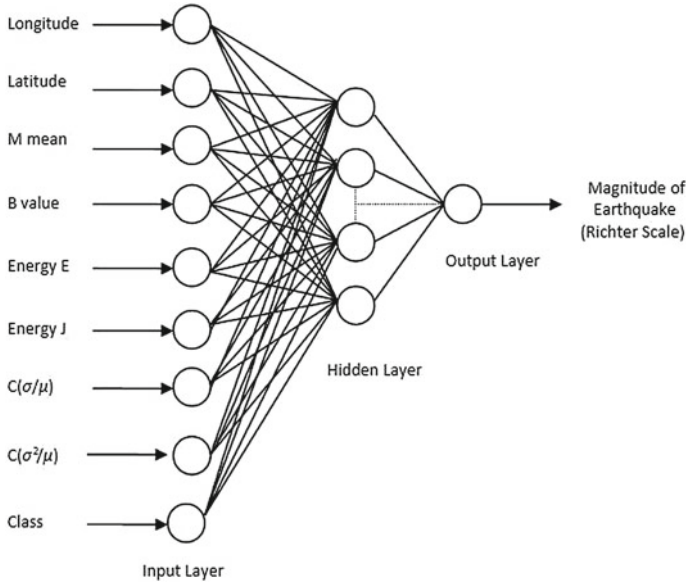


Fig. 2 Backpropagation neural network model with parameters optimized after PSO

diminish this error, until the point when the moment that the Artificial Neural Network takes in the training information [8]. The neural network model is demonstrated in the figure (Fig. 2).

The sum of the inputs is multiplied with corresponding weights W_{ji} for generating activation function.

$$A_j(\bar{x}, \bar{w}) = \sum_{i=0}^n x_i w_{ji} \tag{3}$$

Sigmoidal function is used as the output function for this work

$$O_j(\bar{x}, \bar{y}) = \frac{1}{1 + e^{A_i(\bar{x}, \bar{w})}} \tag{4}$$

Since the error refinement among genuine and expected results, the error relies upon the weights and we have to alter the weights so as to limit the error. We can portray the error function for the output of every neuron:

$$E_j(\bar{x}, \bar{w}, d) = (O_j(\bar{x}, \bar{w}) - d_j)^2 \tag{5}$$

Then it is found how error value depends on the inputs outputs and weights.

$$\Delta W_{ji} = -\eta \frac{\partial E}{\partial W_{ji}} \quad (6)$$

ΔW_{ji} is the adjustment of each weight and ‘ η ’ is the constant eta. Now we have to find out how much error depends on output.

$$\frac{\partial E}{\partial O_j} = 2(O_j - d_j) \quad (7)$$

Now, to find out how much the output depends upon the activation and then on weights we compute the following:

$$\frac{\partial O_j}{\partial W_{ji}} = \frac{\partial O_j}{\partial A_j} \frac{\partial A_j}{\partial W_{ji}} = O_j(1 - O_j)x_i \quad (8)$$

The difference with respect to each weight will be

$$\Delta W_{ji} = -2\eta(O_j - d_j)O_j(1 - O_j)x_i \quad (9)$$

If we have to change V_{ik} , the weights (V_{ik}) of a past layer, we expect first to register how the error depends not on the weight, yet in the observation from the past layer, i.e., supplanting W by x as appeared in the below equation.

$$\Delta V_{ik} = -\eta \frac{\partial E}{\partial V_{ik}} = -\eta \frac{\partial E}{\partial x_i} \frac{\partial x_i}{\partial V_{ik}} \quad (10)$$

where

$$\frac{\partial E}{\partial W_{ji}} = 2(O_j - d_j)O_j(1 - O_j)W_{ji} \quad (11)$$

$$\frac{\partial x_i}{\partial V_{ik}} = x_i(1 - x_i)V_{ik} \quad (12)$$

5 Computed Parameters

To do the expectation of magnitude in the examination zone, the data source stock collected in the midst of 2010–2016 is used. The test connection between magnitude, frequency, and vitality of seismic tremor occasions is remarkable as the Gutenberg–Richter (G–R) connections. The recorded earthquake data for the examination area are divided into different pre-undefined periods of time, for instance, occasions in light of magnitude more than 3.5, the input to the neural networks are eight computational parameters called seismicity input vectors [9].

***b*-value**

Since Gutenberg and Richter surveyed the parameters a and b , the appraisal of the parameters has been a great part of the time used as a piece of true figuring of seismicity. Greatest probability b -value were prepared using the following equation [10].

$$b = \frac{1}{M_{\text{mean}} - M_{\text{mean}}} \log_e \quad (13)$$

Energy e

Energy E (E in ergs) released during an earthquake can be calculated from the following equation:

$$\log E = 5.8 + 2.4m \quad (14)$$

Energy j

The Seismic wave energy J (ergs) can be calculated from the following equation:

$$\log_{10} J = 9 + 1.8M \quad (15)$$

Longitude and latitude

Longitude specifies the east–west position of a point on the Earth’s surface and latitude specifies north–south position supports several clustering-algorithm implementations, all written in MapReduce, each with its own set of goals and criteria.

6 Conclusion

This prediction of the earthquake using Particle Swarm Optimization-Backpropagation model is efficient. As the optimized input parameters are given as input to the backpropagation neural network, this model is more accurate than a simple backpropagation network. The accuracy for prediction depends on the relevance of the input parameters. This model allows to clear the area indicated according to future earthquake prediction by it, moreover allows to give awareness to the people in that location and to do precautions for the loss of lives and financial misfortunes.

Acknowledgements The authors would like to thank the Department of Computer Science at Christ University, Bengaluru, India for their wholehearted support.

References

1. Pannakkat, A., & Adeli, H. (2007). Neural network models for earthquake magnitude prediction using multiple seismicity indicators. *International Journal of Neural Systems*, 17(1), 13–33.
2. Arjun, C. R., & Kumar, A. (2009, March). Artificial neural network—Based estimation of peak ground acceleration. *ISSET Journal of Earthquake Technology*, Paper no. 501, 46(1), 19–28.
3. Alarifi A. S., & Alarifi, N. S. (2009). Earthquake magnitude prediction using artificial neural network in northern red sea area. American Geophysical union, fall meeting 2009.
4. Adeli, H., & Panakkat, A. A probabilistic neural network for earthquake magnitude prediction. *Neural Networks*, 22 (in press).
5. Widrow, B., & Lehr, M. A. (1990). 30 years of adaptive neural networks: Perceptron, madaline, and backpropagation. *Proceedings of the IEEE*, 78(9), 1415–1442.
6. Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. In *Proceeding of the 1995 IEEE International Conference on Neural Networks*, Perth, Australia (pp. 1942–1948).
7. Xu, F., & Song, X. (2010). Neural network model for earthquake prediction using DMETER data and seismic belt information. In *SecondWRI Global Congress on Intelligent Systems*.
8. Moatti, A., & Amin-Nasseri, M. R. (2013). Pattern recognition on seismic data for earthquake prediction purpose. In *International Conference on Environment, Energy, Ecosystems and Development*.
9. Kanmani, K., Vasavi, J., & Ramya, V. (2016). Forecasting earth quake using back propagation algorithm based on artificial neural network. *International Word Press, IJCTA*, 9(37).
10. Narayanakumar, S., & Raja, K. (2016). A BP artificial neural network model for earthquake magnitude prediction in Himalayas, India. In *International Conference on Circuits and Systems*.

Analysis and Detection of Diabetes Using Data Mining Techniques—A Big Data Application in Health Care



B. G. Mamatha Bai, B. M. Nalini and Jharna Majumdar

Abstract In digitized world, data is growing exponentially and Big Data Analytics is an emerging trend and a dominant research field. Data mining techniques play an energetic role in the application of Big Data in healthcare sector. Data mining algorithms give an exposure to analyse, detect and predict the presence of disease and help doctors in decision-making by early detection and right management. The main objective of data mining techniques in healthcare systems is to design an automated tool which diagnoses the medical data and intimates the patients and doctors about the intensity of the disease and the type of treatment to be best practiced based on the symptoms, patient record and treatment history. This paper emphasises on diabetes medical data where classification and clustering algorithms are implemented and the efficiency of the same is examined.

Keywords Big data health care · Data mining techniques · Gaussian Naïve Bayes · OPTICS · BIRCH

1 Introduction

In India, healthcare systems have gained importance in the recent years with the emergence of Big Data analytics [1]. Diabetes mellitus is posing a unique health problem in the country today, and hence India ranks top in the world. Diabetes is a chronic medical condition which can be administered and controlled through changes in lifestyle at an initial stage. At advanced stages, diabetes can be controlled by timely detection, right diagnosis and proper medication. Statistics as per today quotes that approximately 145 million people worldwide are affected by diabetes mellitus and 5% of Indian population contributes towards this rate [2].

B. G. Mamatha Bai (✉) · B. M. Nalini · J. Majumdar
Department of CSE, Nitte Meenakshi Institute of Technology, Bangalore, India
e-mail: mamathamane@gmail.com

J. Majumdar
e-mail: jharna.majumdar@gmail.com

Diabetes is a condition in which the human body will not be able to generate the available amount of insulin which is necessary to balance and monitor the quantity of sugar in the body. This disease accounts for various diseases such as heart disease, nerve damage, kidney disease, blindness and blood vessels damage and many more [3]. Diabetes depends on two reasons:

- Required amount of Insulin is not produced by the pancreas. This specifies Type-1 diabetes and occurs in 5–10% of people.
- In Type-2, cells become inactive to the insulin being produced. Gestational diabetes occurs in women when they develop high sugar level during the time of pregnancy and may disappear after that.

Interpretation and analysing the presence of diabetes is a significant problem to classify. A classifier is essential and intended in such a way that is more cost efficient, convenient and precise. Big Data and data mining techniques provide a great deal to human-related application. These methods find the most appropriate space in the medical diagnosis which is one of the classification phenomena. A physician is supposed to analyse many factors before actual diagnosis of the diabetes leading to a difficult task. In recent times, designing of automatic diabetes diagnosis system has been designed by adopting machine learning and data mining techniques.

2 Literature Survey

The predictive analysis of diabetic patients based on the age and gender using various data mining classification algorithms techniques are discussed [4]. Diabetes is characterised by abnormal metabolism, mostly hyperglycaemia, and an associated risk for specific complications affecting the eyes, kidney, and nervous system and many more. Classification by decision tree induction and classification by Bayesian network are used for predicting the diabetes for people of different age groups and gender.

Ordering points to identify the clustering structure (OPTICS) algorithm is used for the cluster analysis and it does not explicitly produce the clusters, rather it creates an improved ordering of the data objects that depicts the structure of the clusters obtained based on density [5, 6]. This ordering of clusters contains information equal to the density-based clustering and equivalent range of parameter settings. This suits well for automatic and interactive analysis of clusters.

OPTICS [7] is a method based on density and an improved version of DBSCAN. It eliminates the negative aspects of DBSCAN, like forming the clusters in varying density environment. It specifies the shortest distance from the core comprising the minimum number of data points and calculates the mean distance among the points which are placed in and around the core and results in a new distance as the new core distance.

Balanced iterative reducing and clustering using hierarchies (BIRCH) is an efficient and accountable data clustering method, which is based on a new in-memory

data structure termed CF tree that summarises the data distribution [8]. BIRCH diminishes the problem of identifying the clusters from the initial dataset into one by clustering the set of summaries, which are much slighter than the original dataset.

BIRCH [9] eventually clusters the inward multidimensional data points and produces the clustering with best acceptable quality from the available resources such as memory and time constraints. BIRCH typically finds a better clustering by scanning the data once and enhances the quality with additional scans ahead [10, 11]. BIRCH was initially designed to form clusters by handling noise in the dataset.

3 Methodology

Data mining is a new pattern for analysing medical data and achieving useful and practical patterns. Data mining helps us to predict the type of disease and do not try to confirm the already identified patterns; rather tries to find already non-identified patterns. The objective of the proposed methodology is to analyse the medical dataset and predict whether the patient is suffering from the diabetes disease or not. This prediction is done using data mining algorithms such as Gaussian Naïve Bayes, BIRCH and OPTICS.

Naïve Bayes data mining technique is applied to the dataset to predict whether the patient is diabetic or not. BIRCH and OPTICS clustering algorithms are used to cluster similar kind of diseased people into one cluster and identify which algorithm is more efficient by calculating the performance measures.

3.1 *Input Dataset*

The source of medical dataset on diabetes is obtained from the UCI machine learning repository from the link <https://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes> [12]. The residents of Arizona, USA and those suffering from diabetes are tested and their data is obtained. It is observed that major percentage of the population suffer from diabetes and the reason behind this is overweight. The dataset under study comprises nine attributes as shown in Table 1.

3.2 *Algorithms*

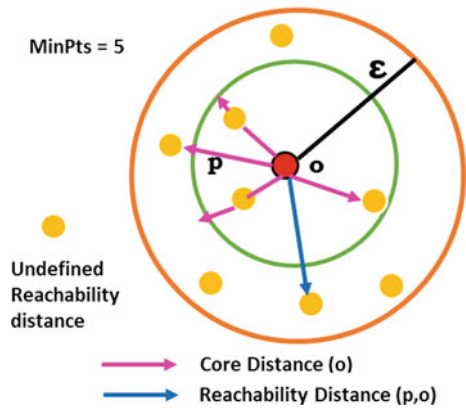
3.2.1 **Gaussian Naïve Bayes**

Naïve Bayes is used for constructing the classifier. Naive Bayes classifiers belong to a family of simple probabilistic classifiers which works on applying the Bayes' theorem with complete (naive) independence assumptions amongst the features [13].

Table 1 Attributes of diabetic medical dataset

S. No.	Attributes
1.	No. of times being pregnant
2.	Glucose tolerance test
3.	Blood pressure level (mmHg)
4.	Body mass index (BMI)
5.	Triceps (mm)
6.	Insulin (mu U/ml)
7.	Age (years)
8.	Pedigree function w.r.t diabetes
9.	Indicating presence or absence of diabetes

Fig. 1 Core distance and reachability distance



Gaussian Naïve Bayes is used for the continuous dataset. Detailed steps of the algorithm are given in Appendix 1.

3.2.2 OPTICS

OPTICS algorithm is used to group the data points considering the density of the objects [14]. It needs two parameters viz. ϵ -Maximum distance (radius) and MinPts (number of points necessary for forming clusters). A point ‘P’ is said to be a core point if it has at least minimum number of points within the neighbourhood. Two values are needed to be considered for each of the object as shown in Fig. 1.

- Core distance

The core distance refers to the least distance ϵ' between point p and a data object in its ϵ -neighbourhood where p becomes a core object.

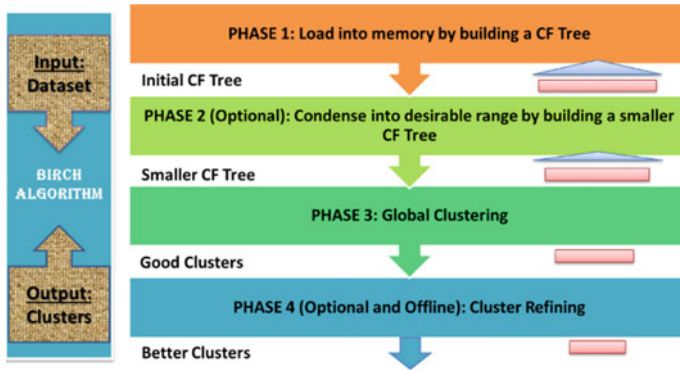


Fig. 2 Overview of BIRCH algorithm

- Reachability distance

It is the distance between the two points p and o or core distance of p , whichever is bigger.

Detailed algorithm steps are given in Appendix 2.

3.2.3 BIRCH

The BIRCH clustering is a type of hierarchical clustering. The algorithm uses the clustering feature and it builds a CF tree. It works for large datasets. It is the first algorithm which is used to handle the noise [15]. There are four phases in BIRCH algorithm as shown in Fig. 2.

The working description of each phase is as follows:

Phase 1: Scan the dataset and build an initial in-memory CF tree.

Phase 2: Scan all the leaf entities of the CF tree and build a new CF tree which is smaller in size. Eliminate all the outliers and form the clusters.

Phase 3: Use the clustering algorithm to cluster all the leaf entities. This phase results in creating a set of clusters.

Phase 4: The cluster centroids obtained in Phase 3 are used as seeds and the data points are redistributed to its closest neighbour seeds to form new cluster representations. Finally, each leaf entity signifies each cluster class.

Details of the algorithm steps are given in Appendix 3.

4 Experimental Analysis

4.1 Gaussian Naïve Bayes

The Gaussian Naïve Bayes analyses the input dataset, segments the continuous values into classes and predicts whether the patient is diabetic or not. The feature value in our example is considered to be the age attribute and the output is shown in Fig. 3. The output depicts that the people are affected by diabetes around the age of 20 and above considering the various factors such as hereditary, lifestyle, food habits, stress, etc.

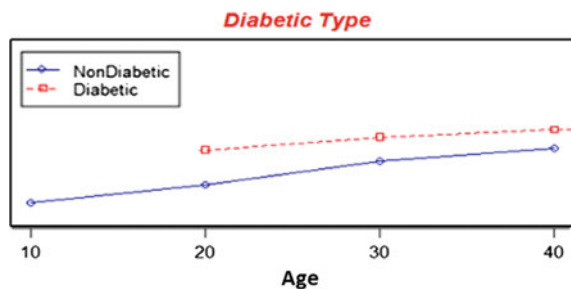
4.2 OPTICS

The entire dataset is grouped into three clusters based on different stages of diabetes w.r.t glucose tolerance test. The clusters formed are normal, prediabetes and diabetes. The cluster plot and each cluster representation are shown in Fig. 4a, b, respectively. The clusters are formed by considering the associated symptoms for different categories of people such as normal, prediabetes and diabetes for detailed cluster analysis and the same is shown in Fig. 5. Figure 6 represents the plot of insulin dependency with the age factor. Analysis of Fig. 3 with 6 shows that people are affected by diabetes at an early age of 20.

4.3 BIRCH

The BIRCH algorithm builds a CF tree and works with large dataset which is capable enough to handle noise. By analysing the diabetic dataset, the three clusters are formed as shown in Fig. 7a, b. Figure 8 refers to the plot of glucose tolerance test with age to illustrate whether the victim is diabetic or not. Figure 9 represents the

Fig. 3 Predicting diabetic or non-diabetic



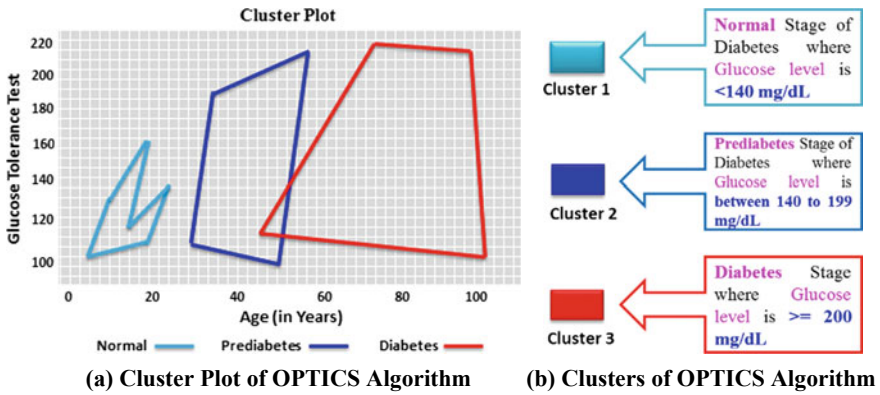


Fig. 4 Results obtained for OPTICS algorithm

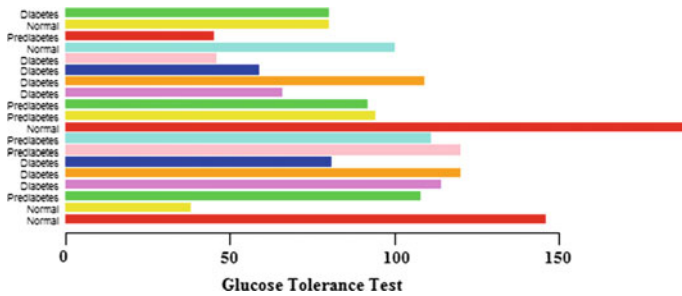


Fig. 5 Cluster analysis of OPTICS algorithm

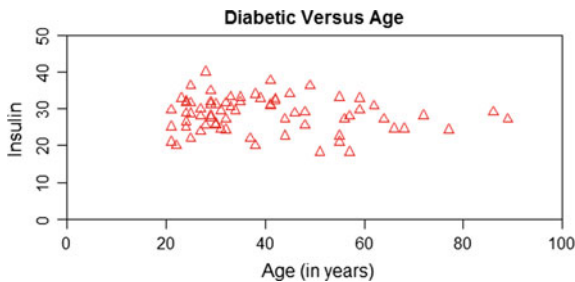
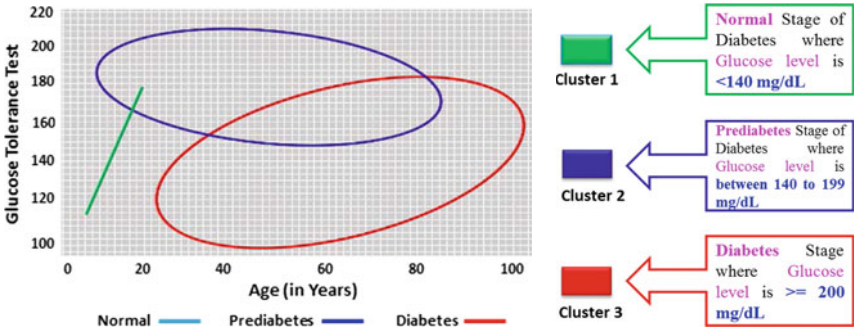


Fig. 6 Plot of insulin dependency with age



(a) Cluster Plot of BIRCH Algorithm

(b) Clusters of BIRCH Algorithm

Fig. 7 Results obtained for BIRCH algorithm

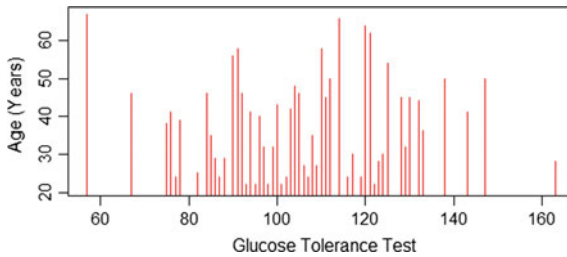


Fig. 8 Glucose tolerance test versus age

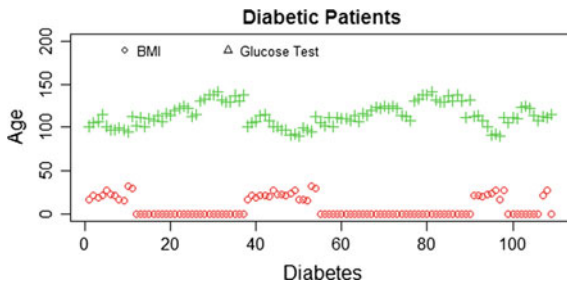


Fig. 9 Plot of BMI and glucose test

plot of body mass index (BMI) with glucose test. Figure 10 depicts different age groups of people having comparably good health factor.



Fig. 10 Scatter plot of normal health

4.4 Performance Analysis Statistics

4.4.1 Silhouette Method

Silhouette methods are used by the researchers to interpret and validate the consistency of the data within the clusters. This method pictorially represents how well each data object fits into the cluster [16]. The range of Silhouette width lies between -1 and $+1$ where the higher value represents the strong similarity and lower value represents the weak similarity of data objects within a cluster.

The results of Silhouette method for OPTICS and BIRCH algorithm are as shown in Figs. 11 and 12, respectively. The average Silhouette width for OPTICS algorithm is 0.44 with less outliers pointing towards the negative scale of the X-axis, whereas the Silhouette width is 0.31 in the case of BIRCH algorithm with few outliers. This clearly indicates that the similarity of the data objects is high in OPTICS.

Fig. 11 Results of Silhouette method for OPTICS algorithm

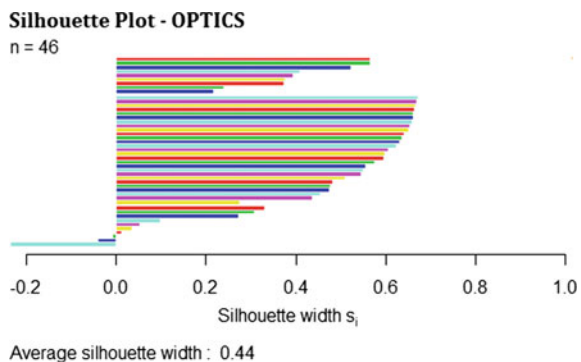


Fig. 12 Results of Silhouette method for BIRCH algorithm

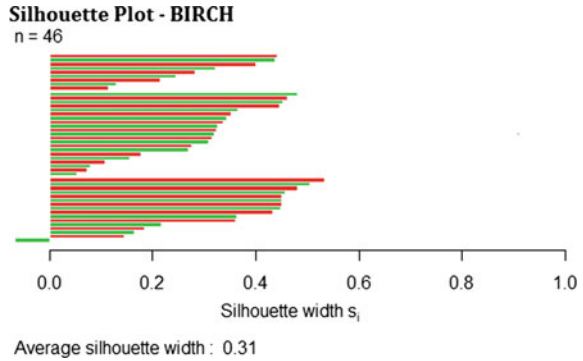
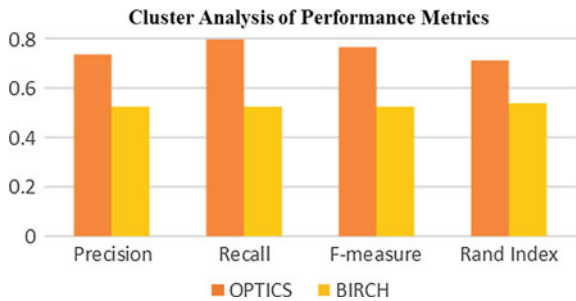


Table 2 Performance metric values

	Precision	Recall	F-measure	Rand index
OPTICS	0.735849	0.795918	0.7647058	0.7108433
BIRCH	0.523809	0.523809	0.5238095	0.5381062

Fig. 13 Cluster analysis of various performance metrics



4.4.2 Performance Metrics

Performance metrics are necessary to compare and analyse the clustering quality of various clustering methods. Higher the performance metric values, superior is the clustering quality. The performance metrics like precision, recall, *F*-measure and Rand Index are used to obtain the best clustering algorithm as shown in Table 2 and Fig. 13. From the analysis, it is found that OPTICS algorithm performs most efficiently compared to BIRCH algorithm.

5 Conclusion and Future Work

In this paper, we have demonstrated the usefulness of data mining algorithms like Gaussian Naïve Bayes, BIRCH and OPTICS for the prediction of diabetic dis-

ease. Data mining techniques are effective in diagnosing and clustering the diabetic patients. Gaussian Naïve Bayes classifier is used for predicting based on the probabilities. BIRCH and OPTICS are used to cluster similar kind of people, where BIRCH is based on the CF tree and OPTICS is based on the ordering of the points in the cluster.

Analysis and comparison of clustering algorithms are performed by considering various performance metrics. It is observed that for the same number of clusters obtained by different clustering techniques, OPTICS is most efficient and is suitable for diagnosis of diabetes.

This work helps the doctors to diagnose and provide the recommended medicine at an early stage. The main aim is to reduce the cost and provide better treatment. In future, this work can be extended with more number of classification algorithms and their accuracy can be compared to find the optimal one.

Acknowledgements The authors express their sincere gratitude to Prof. N. R. Shetty, Advisor and Dr. H. C. Nagaraj, Principal, Nitte Meenakshi Institute of Technology for giving constant encouragement and support to carry out research at NMIT. The authors extend their thanks to Vision Group on Science and Technology (VGST), Government of Karnataka to acknowledge our research and providing financial support to set up the infrastructure required to carry out the research.

Appendix 1

Algorithm: Gaussian Naïve Bayes

Input: Dataset

Output: Classification into different categories

Algorithm Steps:

Step 1. Segment the data by the class.

Step 2. Calculate the probability of each of the class.

$$Class\ Probabilit = \frac{Class\ Count}{Total\ Count}$$

Step 3. Find the average and variance of individual attribute x belonging to a class c.

- a. Let μ_x be the average of the attribute values in x allied with class c.
- b. Let σ_x^2 represent the variance of the attribute values in x related with class c.

Step 4. The probability distribution is computed by

$$p(x = v|c) = \frac{1}{\sqrt{2\pi\sigma_x^2}} e^{-\frac{(v-\mu_x)^2}{2\sigma_x^2}}$$

Step 5. Calculate the probability of the attribute x

$$P(x_1, x_2, \dots, x_n | c) = \prod_i P(x_i | c)$$

Naïve Bayes Classifier = $\text{argmax } P(c) \prod_i P(x_i | c)$

Step 6. End.

Appendix 2

Algorithm: OPTICS

Input: Dataset

Output: Clusters

Algorithm Steps:

Step 1. Initially ϵ and MinPts need to be specified.

Step 2. All the data points in the dataset are marked as unprocessed.

Step 3. Neighbours are found for each point p which is unprocessed.

Step 4. Now mark the point as processed.

Step 5. Initialize the core distance for the data point p.

Step 6. Create an Order file and append point p to the file.

Step 7. If core distance initialization is unsuccessful, return back to Step 3 otherwise go to Step 8.

Step 8. Calculate the reachability distance for each of the neighbours and update the order seed with the reference of new values.

Step 9. Find the neighbours for each data point in the order seed and update the point as processed.

Step 10. Fix the core distance of the point and append to the order file.

Step 11. If undefined core distance exists, go back to Step 9, else continue with Step 12.

Step 12. Repeat Step 8 until there is no change in the order seed.

Step 13. End.

Appendix 3

Algorithm: BIRCH

Input: Dataset

Output: Clusters

Algorithm steps:

Step 1. Set an initial threshold value and insert data points to the CF tree w.r.t the Insertion algorithm.

Step 2. Increase the threshold value if the size of the tree exceeds the memory limit assigned to it.

Step 3. Reconstruct the partially built tree according to the newly set threshold values and memory limit.

Step 4. Repeat Step 1 to Step 3 until all the data objects are scanned and form a complete tree.

Step 5. Smaller CF trees are built by varying the threshold values and eliminating the Outliers.

Step 6. Considering the leaf entities of the CF tree, the clustering quality is improved by applying the global clustering algorithm.

Step 7. Redistribution of data objects and labelling each point in the completely built CF tree.

References

1. Raghupathi, W., & Raghupathi, V. (2014). Big data analytics in healthcare: Promise and potential. *Health Information Science and Systems*.
2. Diabetes Mellitus. https://en.wikipedia.org/wiki/Diabetes_mellitus.
3. Agicha, K., et al. Survey on predictive analysis of diabetes in young and old patients. *International Journal of Advanced Research in Computer Science and Software Engineering*.
4. Iyer, A., Jeyalatha, S., & Sumbaly, R. (2015, January). Diagnosis of diabetes using classification mining techniques. *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, 5(1).
5. Ankerst, M., Breunig, M. M., Kriegel, H.-P., & Sander, J. OPTICS: Ordering Points To Identify the Clustering Structure. Institute for Computer Science, University of Munic.
6. Alzaalan, M. E., & Aldahdooh, R. T. (2012, February). EOPTICS “Enhancement ordering points to identify the clustering structure”. *International Journal of Computer Applications (0975–8887)*, 40(17).
7. Senthil kumar, M., & Rangarajan, R. (2011). Ordering points to identify the clustering structure (OPTICS) with ant colony optimization for wireless sensor networks. *European Journal of Scientific Research*, 59(4), 571–582 (ISSN 1450-216X).
8. Zhang, T., Ramakrishnan, R., & Livny, M. (1997). BIRCH: A new data clustering algorithm and its applications. *Data Mining and Knowledge Discovery*, 1, 141–182.
9. Zhang, T., Ramakrishnan, R., & Livny, M. BIRCH: An efficient data clustering method for very large databases.
10. Du, H. Z., & Li, Y. B. (2010). An improved BIRCH clustering algorithm and application in thermal power. In *2010 International Conference on Web Information Systems and Mining*.
11. Feng, X., & Pan, Q. The algorithm of deviation measure for cluster models based on the FOCUS framework and BIRCH. In *Second International Symposium on Intelligent Information Technology Application*.
12. UCI Machine Learning Repository Pima Indians Diabetes Database <https://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes>.
13. Naïve Bayes. https://en.wikipedia.org/wiki/Naive_Bayes_classifier.
14. Optics Algorithm. https://en.wikipedia.org/wiki/OPTICS_algorithm.
15. Birch Algorithm. <https://people.eecs.berkeley.edu/~fox/summaries/database/birch.html>.
16. Silhouette Method. [https://en.wikipedia.org/wiki/Silhouette_\(clustering\)](https://en.wikipedia.org/wiki/Silhouette_(clustering)).

Cyclic Scheduling Algorithm



Ravin Kumar

Abstract CPU Scheduling has played a critical role in making efficient systems, It is the mechanism which allows one process to use the CPU for execution while other processes are put on hold because of unavailability of any required resource, with the aim of maximizing the CPU utilization and reducing the waiting time and the turnaround time. This paper presents a new Scheduling Algorithm, which supports preemption, reduces the turnaround time and the waiting time. To show its effectiveness, its comparison is done with other traditional scheduling algorithms including First Come First Serve, Shortest Job First, and Round Robin Scheduling Algorithm, and as a result it was found that the proposed algorithm provides a new and effective scheduling approach which reduces the average waiting time and average turnaround time in a much better way than the traditional approaches.

Keywords Preemption · Scheduling · Resource utilization · Waiting time

1 Introduction

Computers have become an important part of today's Information Age, due to their ability to perform several tasks simultaneously. Inside a computer, processors play a crucial role in decision-making, and calculation-related tasks. When processor has more than one process to execute, at that time scheduling algorithms [1] are required to manage execution efficiency of processes. CPU Scheduling is one of the fundamental concepts of Operating System, which are designed with the aim of maximizing the utilization of available system resources, resulting in a better, fair, and fast use of shared resources among the available processes. There are various traditional scheduling algorithms which were designed with the aim of better resource utilization, some of them are FCFS, SJF, and Round Robin scheduling algorithm [2]. There are many factors that require attention during the scheduling process, this

R. Kumar (✉)

Department of Computer Science, Meerut Institute of Engineering and Technology, Meerut
250005, Uttar Pradesh, India
e-mail: ravin.kumar.cs.2013@miet.ac.in

© Springer Nature Singapore Pte Ltd. 2019

N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_38

457

includes burst time, arrival time, and priority of the process, with the aim of reducing the average waiting time and average turnaround time [3]. We have studied various scheduling algorithms from existing literature.

FCFS (First Come First Serve)—This is one of the most earlier approaches which was used for scheduling purpose. This follows the ideology of “First Come First Serve”, i.e., assigning CPU time first to that process, which requested them first [4].

SJF (Shortest Job First)—In this approach, CPU is allocated to that process which has the smallest burst time available, i.e., the job with the shortest computation time is executed first. When a job first arrives, it is put inside the ready queue, and then the job which has the smallest burst time is chosen for execution [5].

Round Robin Scheduling—In this approach, a preemptive time-slicing is performed with the help of a time quantum or time slice. A time slice or time quantum is a small unit of time, due to which it can give the effect of all processors sharing the CPU equally with its given time quantum.

This paper proposes a new CPU scheduling algorithm, which can be used for the same purpose, and when compared with other approaches, it shows comparatively good results.

2 Related Work

Kumar et al. [6] designed a method to suggest the length of the next CPU burst in SJF, and showed that approximate length of the next CPU burst is similar to the length of the previous request. Khan et al. [7] performed a comparative study over SJF and FCFS for similar priority jobs. Li et al. [8] improvised the FIFO algorithm by using fuzzy logic, and showed that it reduces the execution time for tasks and increases the resource utilization. Indusree et al. [9] suggested an improvement to RR by calculating the dynamic time quantum each burst time for processes present in the ready queue. Singh et al. [10] developed a multi-queue-based scheduling approach for cloud architectures. Lin et al. [11] showed that weighted RR performs better than FCFS in Hama architecture. Jha et al. [12] proposed a hybrid algorithm of RR and priority scheduling, which was an improvement over both traditional RR. In servers based on OpenFlow, Peng et al. [13] suggested a weighted RR algorithm that can be used for load balancing. Chaturvedi et al. [14] utilized improved RR in workflow applications in the cloud. Rao et al. [15] proposed a dynamic time calculation mechanism for RR for improving its performance.

3 Proposed Scheduling Algorithm

Let us assume a set of processes $P_0, P_1, P_2, \dots, P_n$ that are needed to be executed using the proposed scheduling algorithm. QUEUE is representing a queue type data

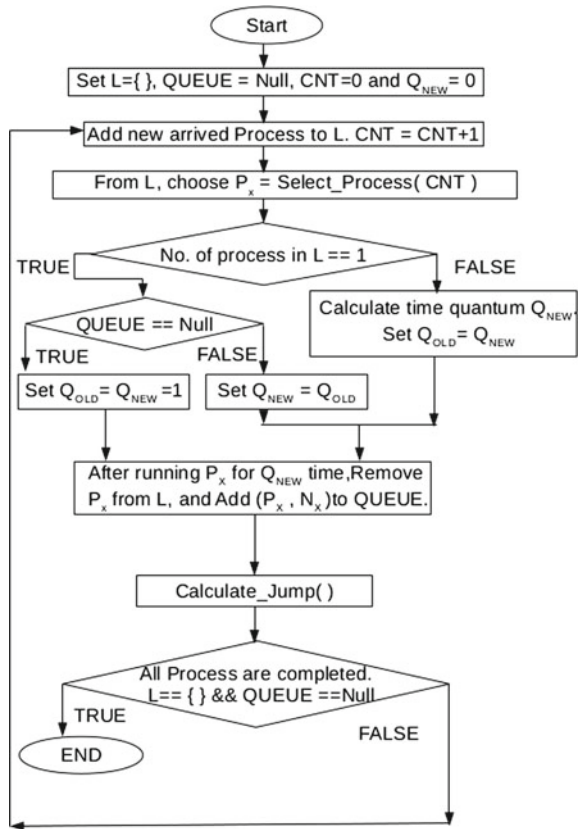
structure which holds those process which has been executed partially. Now, let us assume that there is a set 'L', which holds all those processes that are not completed and their Arrival Time is either less than or equal to the current time. Flowchart of the proposed algorithm is given in Fig. 1.

Cyclic Scheduling Algorithm ()

Parameters Q_{OLD} and Q_{NEW} represent the dynamic time quantum, CNT represents an integer value that is used to decide the dynamic time quantum. RT_x is the remaining time of the selected process. Max and Min are maximum and minimum remaining time among processes that are either present in L or are in QUEUE. N represents the total number of active and arrived processes.

Given below is the flowchart of the proposed scheduling algorithm (Fig. 1).

Fig. 1 Flowchart of Cyclic Scheduling Algorithm



BEGIN

Step 1: Set $L = \{ \}$, QUEUE = Null, CNT = 0 and $Q_{NEW} = 0$.

Step 2: Add all the new process that had arrived, in set L, and update $CNT = CNT + 1$

Step 3: $P_x = \text{Select_Process}(CNT)$

Step 4: IF number of process in $L == 1$

then

IF QUEUE == Null

then

$Q_{OLD} = Q_{NEW} = 1$

ELSE

$Q_{NEW} = Q_{OLD}$

end IF

ELSE

// calculating new time quantum.

$Q_{NEW} = \text{minimum}(RT_x, (Max - Min) / 2, \Sigma RT / N)$

$Q_{OLD} = Q_{NEW}$.

// in above formula remove all zero values, and calculate $Q_{NEW} (>= 1)$.

end IF

Step 5: After running P_x for Q_{NEW} time, remove P_x from L, and add (P_x, N_x) to QUEUE, here N_x can be calculated as :

$N_x = N_{TEMP} + \text{number of } (P, N) \text{ pairs present in QUEUE.}$

Where,

$N_{TEMP} = \text{Number of process in L} + \text{Number of process that just arrived during this execution.} + 1$

Step 6: Calculate_Jump ()

Step 7: IF All Processes are completed and $L == \{ \}$ and QUEUE == Null

then

Exit.

Else

Goto step 2.

END

Select_Process (CNT)

This function is used to decide which process to be chosen for execution from set L.

BEGIN

Step 1: IF $CNT \% 2 == 1$

then

return Process which have minimum remaining time in set L.

ELSE

return Process present in L and have arrival time closer to
(Maximum Arrival time + Minimum Arrival time) / 2.

Maximum and Minimum are calculated among processes present in L.

END

Calculate_Jump ()

This function is used to make use of dynamic time quantum to place processes back from QUEUE to set L.

BEGIN

```

Step 1: For each ( PT, NT ) present in QUEUE
    IF NT <= Number of ( P , N ) pairs present in QUEUE.
        Then
            For each ( PV, NV ) present in QUEUE
                NV = NV - 1
            end For
            Remove ( PT, NT ) from QUEUE.
            IF Remaining time of PT > 0
                then
                    Put process PT in set L.
            end IF
        end IF
    end For
END
    
```

4 Working Demonstration

To better understand the proposed algorithm, we have considered a set of processes with burst and arrival times, so that the behavior of the new proposed scheduling algorithm can be easily understood (Table 1).

Initially, we have $P = \{ \}$, QUEUE = Null, CNT = 0 and $Q_{NEW} = 0$.

Table 2 shows the status of processes, after applying the proposed CSA.

After applying the proposed scheduling algorithm, the waiting time and turnaround time of each process are evaluated, it is given in Fig. 2.

Table 1 Sample data to demonstrate the procedure of Cyclic Scheduling Algorithm

Process	Arrival time	Burst time
P0	0	2
P1	0	3
P2	2	1

Table 2 Applying Cyclic Scheduling Algorithm

Current time	CNT	Q_{NEW}	Set L { }	QUEUE
0	0	0	{P0, P1}	Null
1	1	1	{P1}	(P0, 2)
2	2	1	{P0, P2}	(P1, 2)
3	3	1	{P1, P2}	(P0, 2)
4	4	1	{P1}	(P2, 2)
5	5	1	{ }	(P1, 1)
6	6	1	{ }	Null

Fig. 2 Representation of individual turnaround and waiting time using Cyclic Scheduling Algorithm

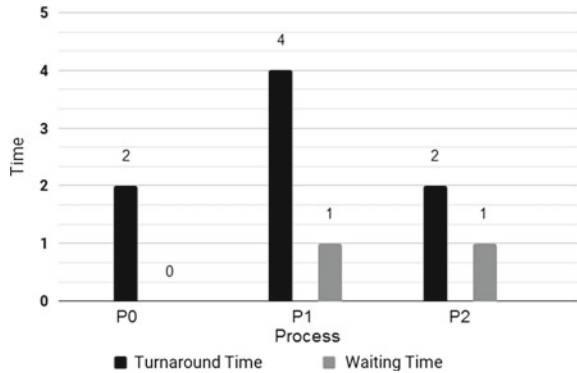


Table 3 Overall performance of the proposed scheduling algorithm on sample data

Parameter	Value
Average turnaround time	2.66
Average waiting time	0.66

The overall performance of the algorithm can be determined by calculating the total waiting time and total turnaround time for the given sample of data. Using the above values of waiting time and turnaround time of the individual processes, the average turnaround time and average waiting time can be calculated, so that the overall performance of the proposed algorithm can be measured. In Table 3, the average turnaround time and average waiting time of the above processes is shown.

5 Result and Discussion

In the proposed algorithm, the outer loop containing set ‘L’ and queue ‘QUEUE’ makes sure that the algorithm has a cyclic nature for the execution of process, similar to that of round-robin algorithm. But instead of using the order in which the processes first arrived, our algorithm has an inner mechanism which helps in improving the algorithm’s performance by providing dynamic time quantum, and then selecting a process on the basis of arrival time and remaining time to execute in the outer loop.

Cyclic Scheduling Algorithm is tested with various test cases on real data sets, and then the obtained result is further compared with FCFS, SJF, and Round Robin Algorithm. To show its effectiveness, four samples of data are taken for applying the scheduling algorithm, and then a comparison is done with FCFS, SJF and Round Robin scheduling algorithms (Table 4).

In Fig. 3, average turnaround and average waiting time are obtained after applying each algorithm is given.

Table 4 Sample 1—Burst time in increasing order

Process	Arrival time	Burst time
P0	0	2
P1	1	7
P2	2	8
P3	3	14

Fig. 3 Comparison of average turnaround and average waiting time of FCFS, SJF, RR, and CSA scheduling algorithms

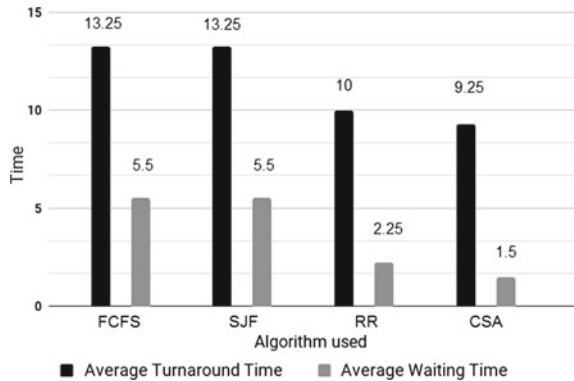


Table 5 Sample 2—Burst time in decreasing order

Process	Arrival time	Burst time
P0	0	14
P1	1	8
P2	2	7
P3	3	2

Table 6 Sample 3—Burst time in random order

Process	Arrival time	Burst time
P0	0	2
P1	1	8
P2	2	7
P3	3	14

Let us consider another sample of data to further check the result, Given below is the set of assumed data on which proposed scheduling algorithm is applied and then the comparison is done with other algorithms (Table 5).

Given below is the graphical representation of the average turnaround and average waiting time obtained after applying each algorithm (Fig. 4).

Similarly, consider another sample of data on which we first apply the proposed scheduling algorithm, and then a comparison is done with other algorithms (Table 6).

For the above sample, the comparative analysis of performance obtained after implementing different scheduling algorithms is given in Fig. 5.

Fig. 4 Comparison of average turnaround and average waiting time of various scheduling algorithms

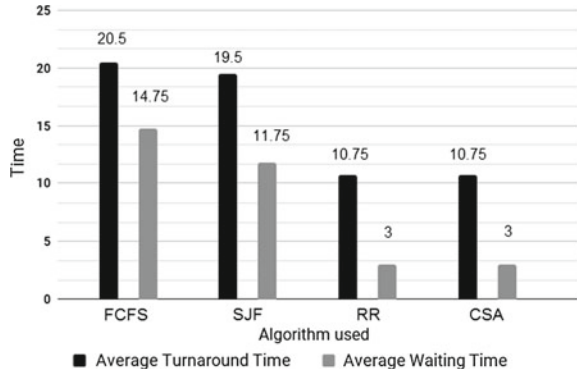


Fig. 5 Comparison of average turnaround and average waiting time of various scheduling algorithms

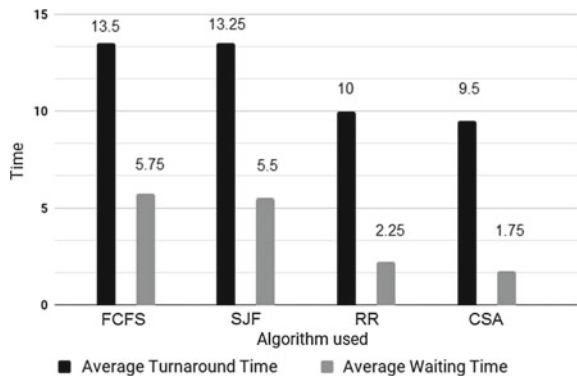


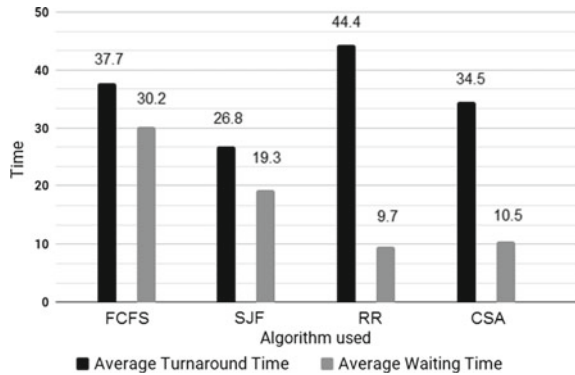
Table 7 Sample 4—Set of ten processes assigned with random values

Process	Arrival time	Burst time
P0	0	3
P1	1	14
P2	2	7
P3	2	2
P4	3	11
P5	3	5
P6	4	8
P7	5	13
P8	6	9
P9	6	3

The fourth sample is generated by randomly assigning arrival and burst time to a set of ten processes (Table 7).

Given below is the comparative analysis of different scheduling algorithms over the above set of processes (Fig. 6).

Fig. 6 Comparison of average turnaround and average waiting time of various scheduling algorithms



6 Conclusion

In this paper, an advanced method is introduced which helps in the reduction of average turnaround time, and average waiting time required by a set of processes (having the same priority) of completing their tasks. Proposed algorithm provides a better approach for scheduling processes as compared to other traditional approaches. This algorithm provides a new way of scheduling processes, and also opens a way for developing new hybrid scheduling algorithms by combining the proposed algorithm with other traditional algorithms. Future research work can be done on adding a priority mechanism to the proposed algorithm.

References

1. Silberschatz, A., & Gagne, G. *Operating system concepts* (8th ed.). WILEY.
2. Mittal, N., Garg, K., & Ashish America (2005). A paper on modified round robin algorithm. *IJLTEMAS*, 4(X1). ISSN 2278–2540.
3. Seltzer, M., Chen, P., & Outerhout, J. (1990). Disk scheduling revisited in USENIX. In H. M. Shamim (ed.), *Operating system, DCSA-2302. Winter technical conference*.
4. Raman, & Mittal, P. K. (2014). An efficient dynamic round robin CPU scheduling algorithm. *IJARCSSE*. ISBN: 2277 128X.
5. Tong, W., & Zhao, J. (2007). Quantum varying deficit round robin scheduling over priority queues. In *International Conference on Computational Intelligence and Security* (pp. 252–256). China.
6. Kumar, M. R., & Renuka Rajendra, B. (2014). Prediction of length of the next CPU burst in SJF scheduling algorithm using dual simplex method. In *2nd International Conference on current Trends in Engineering and Technology ICCTET*.
7. Khan, R., & Kakhani, G. (2015). Analysis of priority scheduling algorithm on the basis of fcs and sjf for similar priority jobs. *International Journal of Computer Science and Mobile Computing*, 4, 324–331.
8. Li, J., Ma, T., Tang, M., Shen, W., & Jin, Y. (2017). Improved FIFO scheduling algorithm based on fuzzy clustering in cloud computing. *Information*, 8(1), 25.

9. Indusree, J. R., & Prabadevi, B. (2017). Enhanced round robin CPU scheduling with burst time based time quantum. In *IOP Conference Series: Materials Science and Engineering* (Vol. 263, No. 4, p. 042038). IOP Publishing.
10. Singh, J., & Gupta, D. (2017). A smarter multi queue job scheduling policy for cloud computing. *International Journal of Applied Engineering Research*, 12(9), 1929–1934.
11. Lin, Z., & Huang, H. (2017). Research on weighted rotation fair scheduling algorithm based on Hama parallel computing framework. *DEStech Transactions on Engineering and Technology Research* *apop*.
12. Jha, H., Chowdhury, S., & Ramya, G. (2017). Survey on various scheduling algorithms. *Imperial Journal of Interdisciplinary Research*, 3(5).
13. Peng, R., & Ding, L. (2017). Research on virtual network load balancing based on OpenFlow. In *AIP Conference Proceedings* (Vol. 1864, No. 1, p. 020014). AIP Publishing.
14. Chaturvedi, P., & Sharma, S. (2017). A provision for workflow scheduling using round robin algorithm in cloud. *International Journal of Advanced Research in Computer Science*, 8(5), 667–673.
15. Rao, G. S. N., Srinivasu, N., Srinivasu, S. V. N., & Rao, G. R. K. (2015). Dynamic time slice calculation for round robin process scheduling using NOC. *International Journal of Electrical and Computer Engineering*, 5(6), 1480–1485.

A Framework for Monitoring Clustering Stability Over Time



K. Namitha and G. Santhosh Kumar

Abstract Mining data streams and arriving at intelligent decisions is becoming more and more important nowadays as a lot of applications produce large volume data streams. Data stream clustering has been considered to be very useful for online analysis of streams. Monitoring the cluster transitions over time provide good insight into the evolving nature of the data stream. This paper introduces a framework for monitoring the stability of individual clusters and clusterings over time, along with the progress of the stream. Tracking the historical evolution of clustering structures is the main focus of this framework. Two real-world datasets have been used for conducting the experiments. The results point up the fact that monitoring the stability of clustering structures will help to get an important hint of the physical events happening in the environment. This information can be used to predict the future clustering structure changes and in turn the upcoming events.

Keywords Data streams · Clusters · Evolution tracking

1 Introduction

With the rapid growth in hardware and software technologies, it has become easier and cheaper to produce data at large scale. Data arriving as continuous, fast stream makes its online analysis an inevitable need of the era. Classification and clustering are the important machine learning approach tried in case of data streams. Different approaches can be seen in the literature for online clustering of the data streams [1–5].

The concept drift happening along the evolution of the stream is an important concern in data stream mining. Change in the underlying data distribution is the basic reason for concept drift. In data stream clustering scenarios, concept changes can be detected only by identifying changes in the data distribution. Since the data distribu-

K. Namitha (✉) · G. Santhosh Kumar
Department of Computer Science, Cochin University of Science and Technology,
Kochi, Kerala, India
e-mail: namithak@cusat.ac.in

tion is closely related to the clustering structure, monitoring the clustering structure has good scope in identification and anticipation of concept change. In addition to this, understanding the changes happening in the clustering structure can be useful for studying the characteristics of the domain in detail and especially decision-making. It has application in many areas like customer relationship management, fraud discovery, healthcare systems, etc. Studying the interrelationship between the clustering structure changes and the real events can help deriving better business strategies and decisions.

MONIC [6] and MEC [7] are two important approaches discussed in the literature for identifying the different category of cluster transitions. Various possible internal and external cluster transitions are defined in these papers. But they do not assess the stability of clustering structures. In this paper, we are proposing a framework that monitors the lifetime of individual clusters and clusterings over time. From the beginning of the stream, each cluster and clustering are tracked from their birth till disappearance. This helps to track the evolution of the data stream. Experiments conducted on two real-world datasets supports the fact that monitoring the clustering stability and hence tracking the evolution helps to derive a relationship between the clustering structures and the physical events. This kind of analysis is useful to understand the changes happening over the stream.

Rest of this paper is organized as follows. Section 2 elaborates the related work. Section 3 describes the proposed framework in detail. Experiments and discussion are included in Sect. 4 and finally, Sect. 5 concludes the paper.

2 Related Work

Plenty of algorithms are available for data stream clustering, a brief account of which can be seen in [8, 9]. But relatively few works can be seen in identifying and tracking cluster transitions. MONIC [6] is a framework introduced for modelling cluster transitions. They have categorized cluster transitions to be internal and external. Clusters generated at two consecutive time points are compared to decide the kind of transitions happened. Spiliopoulou et al. [10] have listed some category of application areas which used MONIC framework for cluster transition analysis. This includes evolution in social networks, change prediction in data stream mining, spatiotemporal analysis, topic evolution, etc. MEC [7] is another similar framework which models different cluster transitions. MEC uses bipartite graphs for tracking cluster transitions. In addition to the usual enumeration method, cluster representation by summary is also considered in this framework.

MONIC+ [11] is an extension of the MONIC framework. It defines different types of clusters, based on which the cluster overlap and cluster transitions are redefined. Held et al. [12] present the visualization method for dynamic clusterings. In addition to the cluster transitions discussed in MONIC, this paper introduces one more transition named ‘rebirth’ of a cluster. ReDSOM method [13] identifies clustering structure changes by kernel density estimation approach. This paper also defines the cluster

transitions like emergence, disappearance, split, absorption, cluster compaction and cluster enlargement, etc.

3 Proposed Framework

This section contains the details of the proposed framework which is intended to be useful for monitoring the stability of clustering structures over time. Here, both the individual clusters and the whole clusterings are considered for tracking the stability. Stability refers to the total lifetime—right from its birth till death/disappearance. To identify and nomenclature the cluster transitions between two consecutive time points, this framework is using the cluster transition models introduced by MONIC and MEC. Particularly ‘survival’ of clusters to the next time point decides the stability of clustering structures. This framework has four components as shown in Fig. 1 and subsequently discussed in the following sections.

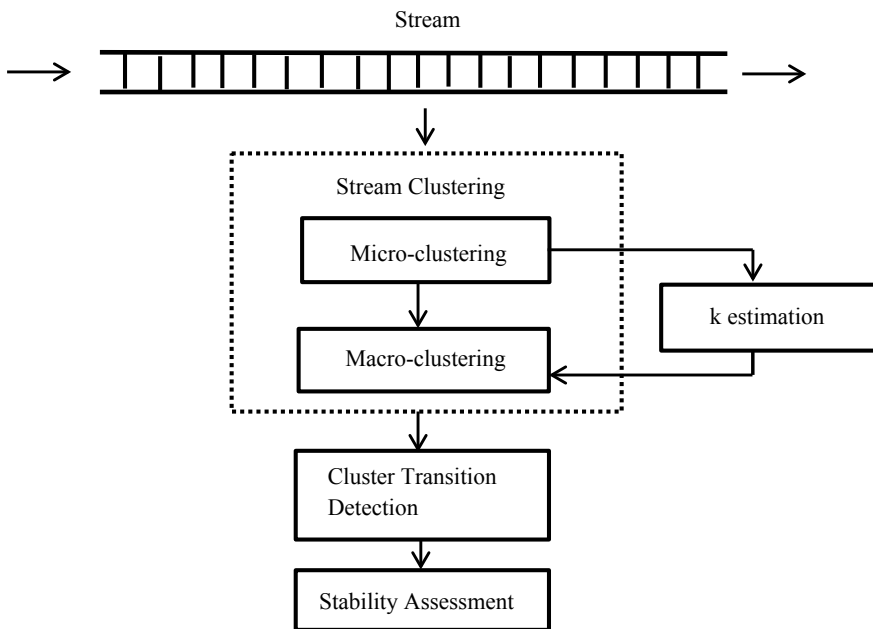


Fig. 1 Components of the framework

3.1 Online Clustering

Online clustering of the data stream is done using CluStream algorithm [3]. CluStream is one of the most popular stream clustering algorithms available in the literature. Like many other stream clustering algorithms, CluStream also processes the stream by using two components—an online component which performs micro-clustering of the stream and an offline component which creates macro-clusters out of the available micro-clusters. Micro-clusters are created and maintained in an online way, and they represent the summary of the stream. Cluster feature vectors [1] are used to store micro-clusters in memory. To create an overall clustering of the past stream, macro-clustering can be performed with the micro-clusters relevant to that period. The main peculiarity of the CluStream algorithm is that it provides the user with an opportunity to mention the period over which the offline macro-clustering has to be performed. The proposed framework utilizes this flexibility and it has employed fixed-size window based macro-clustering.

3.2 Deciding the Number of Clusters Dynamically

CluStream algorithm uses k-means clustering at two different stages—during initialization of the micro-clusters at the beginning of the stream and during the offline clustering phase to create macro-clusters from micro-clusters. The usual assumption while using k-means clustering is that the number of clusters or the value of k is fixed and it is provided by the user. But there are limitations while applying this assumption to data streams, as the streams are highly dynamic in nature and concept changes are possible. To overcome this difficulty, we are using a method of deciding the value of k dynamically depending on the current characteristics of the stream. Whenever k-means clustering algorithm has to be called, it is preceded by a step of calculating the most suitable value for k .

This framework uses Ordered Multiple Runs of k-Means (OMRk) algorithm [14] as the approach to decide the number of clusters, k . In this approach, k-means algorithm is run multiple times with value of k varying from 2 to \sqrt{N} where N is the total number of points to be clustered. To validate the relative quality of clusterings created at each execution of k-means, Simplified Silhouette [15, 16] is used as the relative clustering validity criteria. Silhouette value of a clustering is measured as follows. Suppose x_i is a data element belonging to the cluster C_a , and $a(x_i)$ is the average distance of x_i to all other data elements within the same cluster. Similarly for each cluster C_b other than C_a , find the average distance between x_i and all the points in C_b . Let $b(x_i)$ be the smallest among these distances. The cluster with this lowest average dissimilarity is termed as the neighbouring cluster of x_i . Low value for $a(x_i)$ and high value for $b(x_i)$ is the desirable feature of a good clustering. Silhouette is defined as

$$s(x_i) = \frac{b(x_i) - a(x_i)}{\max\{a(x_i), b(x_i)\}}$$

Higher values of $s(x_i)$ indicate better clustering. In order to reduce the computational complexity, instead of calculating distance to each data point in a cluster, distance to the cluster centre is taken. So $a(x_i)$ is the distance between the data point and its own cluster's centroid. Similarly to calculate $b(x_i)$, distances between the data point and all the other cluster centroids are considered. This variant of silhouette computation is called Simplified Silhouette (SS). Average of all the $s(x_i)$ over $i = 1, \dots, N$ is taken as the SS value of that partition.

$$SS = \frac{1}{N} \sum_{i=1}^N s(x_i)$$

Partition with the highest value of SS is the best clustering, and that particular value of k is chosen. Experiments supported the fact that keeping the value of k dynamic helps to adapt easily to the changes or evolution of the stream.

3.3 Transition Model

This framework is motivated by the cluster transition models introduced by MONIC [6] and MEC [7]. It tracks the five categories of external transitions defined in MONIC namely—survival, absorption, split, disappearance and creation of new clusters. Clusters are represented by the enumeration method, i.e. a cluster is characterized by the data points assigned to it. Cluster characteristics like centre, radius etc., are calculated from its member elements [17].

Macro-clustering is performed after processing each fixed-size window of samples from the stream. Macro-clusters created at such consecutive time points are compared to identify the type of transition each cluster has undergone. To decide the survival of a cluster, a survival threshold value should be provided by the user. Also, a split threshold is used to decide if a cluster at a particular time point is split into more than one cluster in the next time point.

3.4 Stability Monitoring

Cluster stability is decided by checking how long it survives. Any transition other than survival is considered to be the end of the lifetime of that particular cluster. That is, cluster identity is strictly followed and even if it gets split or absorbed, its identity is considered to be lost. If all the clusters in a clustering survive to the next time point, then it is taken as clustering survival. In this case, none of the clusters

disappears and no new clusters get created in between. Since the number of clusters is allowed to vary according to the dynamic characteristics of the stream, clustering as a whole survives only if the clustering structure is perfectly consistent. When the clustering structure gets disturbed slightly, it might be indicating a new emerging trend or a new event. Experiments are conducted to check this relationship.

4 Experiments and Discussion

Experiments are conducted to check the capability of the proposed framework for assessing the stability of clusters and clustering over time in data stream mining scenarios. Two real-world datasets have been used to conduct the experiments. Results are visualized by plotting how long clusters and clusterings survive. Stability of a cluster/clustering is shown by using thick straight line, length of which represents the duration of stability.

4.1 Datasets

KDD-CUP'99 intrusion detection dataset:

The KDD-CUP'99 intrusion detection dataset is a standard one, available online and being used as a benchmark dataset in data stream clustering experiments. 10% version of this dataset is used in our experiments as it is more concentrated than the original one [18]. Each entry in the dataset refers either to a normal connection or an intrusion. Intrusion attacks are classified into 22 categories, making a total of 23 classes including the normal class. The dataset contains 494,020 records in total, each having 41 attributes, in which 34 attributes are continuous and 7 attributes are symbolic. In these experiments, we have considered only the continuous attributes. Min-max normalization is performed to prepare the data for experiments [19]. The stream is generated from the dataset by keeping the same order as the data records are in the dataset.

Weather data from the Automatic Weather Station:

The second dataset used in the experiments is a weather dataset that contains the parameters collected by the Automatic Weather Station (AWS) at Advanced Centre for Atmospheric Radar Research (ACARR) of Cochin University of Science and Technology, India. Data collected by the AWS at 1-minute interval is used for this study. From the original data, the most relevant six parameters are identified with the help of domain experts and these six parameters are used in our study. These six parameters include temperature, wind speed, wind direction, solar radiation, net radiation and cloud radiation. Data collected during the period 4 April 2016–20 June 2016 is used for the experiment. This time period is chosen because the south-west

monsoon enters Kerala during the first week of June and notable changes happen in the atmospheric conditions during this period.

4.2 Experimental Setup

Experiments mainly focused on tracking cluster transitions and hence, monitoring the historical evolution of clusters and clusterings along with the progress of the stream. The lifetime of every single cluster, from its birth to disappearance is being tracked. The disappearance of a long-lived cluster might be giving a hint on an important physical event happening in the environment that produces the data. In addition to individual clusters, the stability of the whole clustering is also being monitored as part of this framework. Obviously, the stability and decay of a clustering have physical significances. Experiments are conducted to establish this kind of relationship between the clustering characteristics of the dataset and the physical events inferred from the data.

As discussed in Sect. 3.1, CluStream is used as the basic stream clustering algorithm in all the experiments. In the offline clustering phase, whenever it uses k-means clustering, the value of k is allowed to vary dynamically depending on the current concept that drives the stream. That is, k is not a fixed parameter supplied by the user. Clustream is a landmark window based [8] clustering algorithm. It gives the flexibility that the user can mention the period over which the offline clustering has to be performed. In our experiments, we used windows of fixed size. This window size is decided on the basis of the characteristics of the dataset. For KDD-CUP intrusion detection dataset, the window size is set to 5000 and for the weather dataset, it is set to 1500. Since the weather dataset contains data at 1-minute interval, approximately 1500 data points constitute one day.

Cluster transitions are identified by using the methodology introduced by MONIC framework [6]. MONIC needs some threshold values to be provided by the user based on which the category of transition is decided. In our experiments, different values from 0.50 to 0.80 are tried for the survival threshold and the split threshold is set to 0.25.

4.3 Results and Discussion

KDD-CUP'99 intrusion detection dataset:

This dataset contains 494,020 records in total and Fig. 2 shows the class labels of these records. Intrusion classes are numbered 1–22 in the order as they are listed in the dataset description¹ and 23 represents ‘normal’ class.

¹http://kdd.ics.uci.edu/databases/kddcup99/training_attack_types.

A stream is simulated from this dataset by keeping the same order as the data points are there in the dataset. The timestamp of the data records is generated, starting from zero and incrementing by one. In Fig. 2, class labels are shown against this timestamp of the stream. Stream is clustered online by using the CluStream algorithm. Offline clustering phase of CluStream is performed on processing every 5000 samples from the stream. Clusterings produced by these offline clustering stages are fed as input to the cluster/clustering stability monitoring framework proposed here.

Results of cluster and clustering survival monitoring processes applied to KDD-CUP intrusion detection dataset are shown in Figs. 3 and 4, respectively. By cross-checking Figs. 2 and 3, it can be inferred that clusters survive quite reasonably for some classes. For some intrusion categories, clusters appear and disappear too frequently and for some others, they remain stable for a long time. Figure 3 shows the survival history and hence the stability of individual clusters.

Survival of the whole clustering in KDD-CUP intrusion detection dataset is depicted in Fig. 4. When all the clusters in a clustering survive over time, it can be considered as a period of clustering stability. As evident from the figure, there are two considerably large periods and three small periods of clustering stability while processing KDD-CUP intrusion detection dataset. It can be observed from Figs. 2 and 4 that the whole clustering remains stable for long duration for class label 18. Class label 18 corresponds to the intrusion type ‘smurf’. For this category of attack,

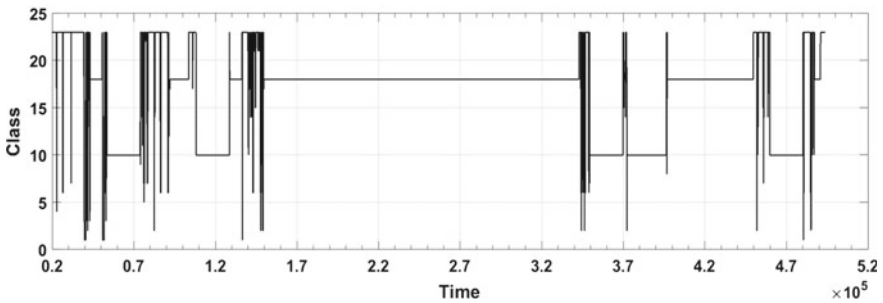


Fig. 2 Class labels of KDD-CUP intrusion detection dataset

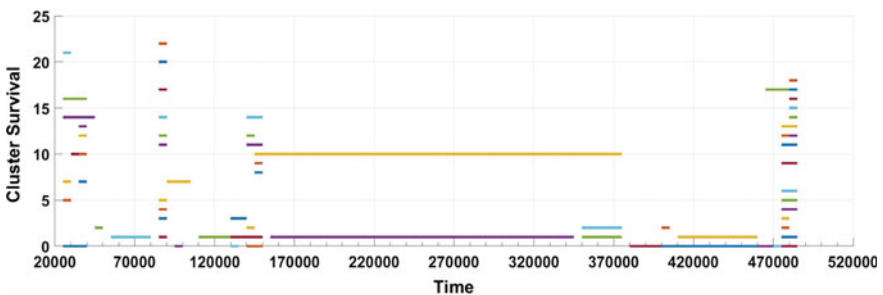


Fig. 3 Cluster survival in KDD-CUP intrusion detection dataset

clustering structure is consistent and reappearance of the same clustering structure indicates the reoccurrence of the same category of attack.

Weather data from the Automatic Weather Station:

Data collected by the AWS contains the rain measurements as well. Rainfall during the period chosen for analysis is shown in Fig. 5. In 2016, south-west monsoon was preceded by a pre-monsoon shower which starts during the week of May 14.

Atmospheric conditions changed considerably during this period of pre-monsoon shower and a corresponding change is seen in the clustering structures as well. Figure 6 shows the cluster survival for this dataset. Most of the clusters remain fairly stable from April 15 to May 13. Clusters disappear and new clusters get created during the week of May 14.

Clustering survival shown in Fig. 7 also supports the fact that the whole clusterings exist fairly stable from April 23 to May 14, i.e. till the beginning of the pre-monsoon shower. Once the rain starts, the atmospheric conditions come to stability again after one week.

Experiments support the fact that stability and the sudden disappearance of clusters and clusterings have relation with the physical events happening in the environment.

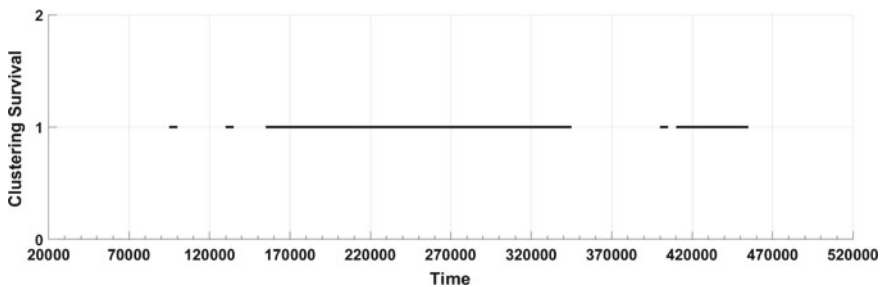


Fig. 4 Periods of clustering survival in KDD-CUP intrusion detection dataset

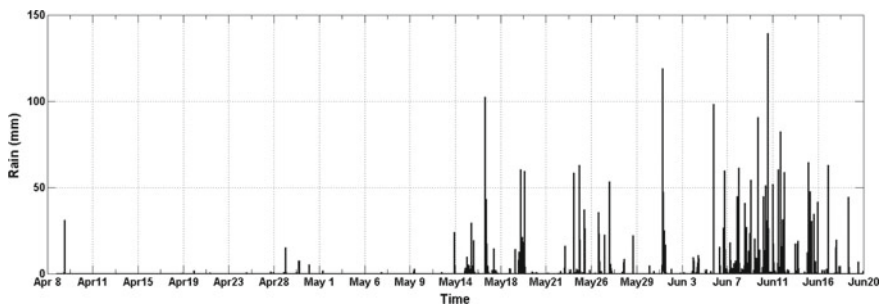


Fig. 5 Rainfall during the period 8-04-2016 to 20-06-2016. Pre-monsoon shower starts during the week of May 14

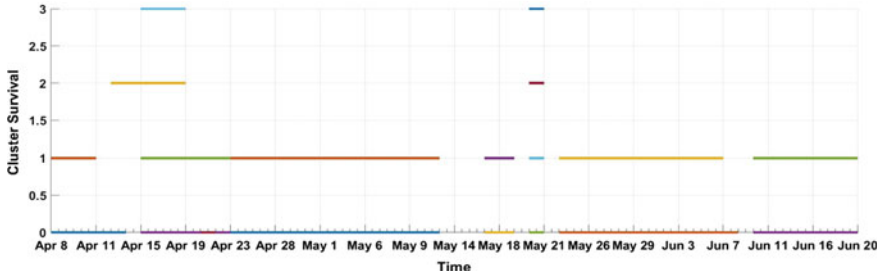


Fig. 6 Cluster survival in weather dataset

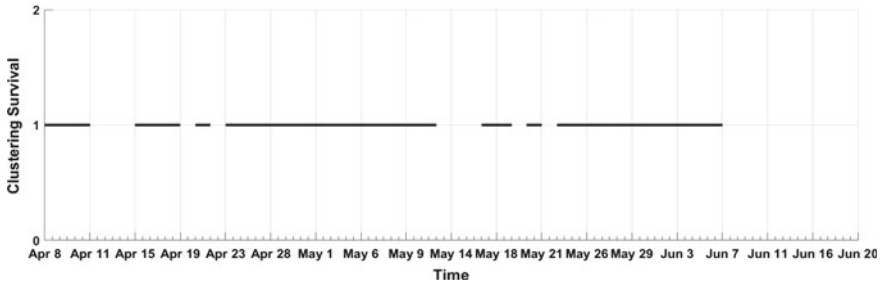


Fig. 7 Periods of clustering survival in weather dataset

This leads to the conclusion that monitoring the clustering characteristics can help the prediction of physical events.

5 Conclusion

This paper proposes a framework for monitoring the stability of clusters and clusterings over time when dealing with data streams. Applications that produce continuous streams of data are becoming very common nowadays. Monitoring the lifetime of individual clusters and clusterings over the stream and finding its relationship to the actual events happening in the environment will help the anticipation of upcoming changes. Stability of the clustering structure has been studied for two datasets with different nature. Experiments emphasize the fact that stability and decay of clustering structures have close relationship to the changes happening in the physical environment. In future, we are planning to use this framework for prediction of creation and disappearance of clustering structures and thus predict the upcoming physical events.

Acknowledgements We would like to thank the faculty and technical staff at Advanced Centre for Atmospheric Radar Research (ACARR), Cochin University of Science and Technology for supporting this work.

References

1. Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data* (Vol. 1, pp. 103–114). <http://doi.org/10.1145/233269.233324>.
2. Guha, S., & Mishra, N. (2000). Clustering data streams. In *Proceedings 41st Annual Symposium on Foundations of Computer Science* (pp. 359–366). <http://doi.org/10.1109/SFCS.2000.892124>.
3. Aggarwal, C. C., Watson, T. J., Ctr, R., Han, J., Wang, J., & Yu, P. S. (2003). A framework for clustering evolving data streams. In *Proceedings of the 29th International Conference on Very Large Data Bases* (pp. 81–92). <http://doi.org/10.1.1.13.8650>.
4. Cao, F., Ester, M., Qian, W., & Zhou, A. (2006). Density-based clustering over an evolving data stream with noise. *Sdm*, 326–337. <http://doi.org/10.1145/1552303.1552307>.
5. Ackermann, M. R., Lammersen, C., Sohler, C., Swierkot, K., & Raupach, C. (2012). StreamKM++: A clustering algorithm for data streams. *Journal of Experimental Algorithmics (JEA)*, 17, 173–187. <https://doi.org/10.1145/2133803.2184450>.
6. Spiliopoulou, M., Ntoutsi, I., & Schult, R. (2006). *MONIC—Modeling and monitoring cluster transitions* (pp. 706–711). <http://doi.org/10.1145/1150402.1150491>.
7. Oliveira, M., & Gama, J. (2010). MEC—Monitoring clusters' transitions. In *Proceedings of the 2010 Conference on STAIRS 2010: Proceedings of the Fifth Starting AI Researchers' Symposium* (pp. 212–224).
8. Silva, J. A., Faria, E. R., Barros, R. C., Hruschka, E. R., De Carvalho, A. C., & Gama, J. (2013). Data stream clustering. *ACM Computing Surveys*, 46(1), 1–31. <https://doi.org/10.1145/2522968.2522981>.
9. Ghesmoune, M., Lebbah, M., & Azzag, H. (2016). State-of-the-art on clustering data streams. *Big Data Analytics, I*(1), 13. <https://doi.org/10.1186/s41044-016-0011-3>.
10. Spiliopoulou, M., Ntoutsi, E., Theodoridis, Y., & Schult, R. (2013). MONIC and followups on modeling and monitoring cluster transitions. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8190 LNAI (PART 3) (pp. 622–626). http://doi.org/10.1007/978-3-642-40994-3_41.
11. Ntoutsi, I., Spiliopoulou, M., & Theodoridis, Y. (2009). Tracing cluster transitions for different cluster types. *Control and Cybernetics*, 38(1), 239–259.
12. Held, P., & Kruse, R. (2013). Analysis and visualization of dynamic clusterings. In *Proceedings of the Annual Hawaii International Conference on System Sciences* (pp. 1385–1393). <http://doi.org/10.1109/HICSS.2013.93>.
13. Wicaksono, P., & Manurung, R. (2014). *Automatic detection of cluster structure changes using relative density self-organizing maps (AusDM)* (pp. 9–17).
14. Naldi, M. C., Fontana, A., & Campello, R. J. G. B. (2009). Comparison among methods for k estimation in k-means. In *ISDA 2009—9th International Conference on Intelligent Systems Design and Applications* (pp. 1006–1013). <http://doi.org/10.1109/ISDA.2009.78>.
15. Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20, 53–65. [https://doi.org/10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7).
16. Vendramin, L., Jaskowiak, P. A., & Campello, R. J. G. B. (2013). On the combination of relative clustering validity criteria. In *Proceedings of the 25th International Conference on Scientific and Statistical Database Management—SSDBM* (p. 1). <http://doi.org/10.1145/2484838.2484844>.
17. Leskovec, J., Rajaraman, A., & Ullman, J. (2014). *Clustering. Mining of massive datasets* (pp. 240–280). http://doi.org/10.1007/SpringerReference_34708.
18. Masud, M., Gao, J., Khan, L., Han, J., & Thuraisingham, B. M. (2011). Classification and novel class detection in concept-drifting data streams under time constraints. *IEEE Transactions on Knowledge and Data Engineering*, 23(6), 859–874. <https://doi.org/10.1109/TKDE.2010.61>.
19. Chen, Y., & Tu, L. (2007). *Density-based clustering for real-time stream data*. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining—KDD '07* (p. 133). <http://doi.org/10.1145/1281192.1281210>.

Efficient Dynamic Double Threshold Energy Detection of Cooperative Spectrum Sensing in Cognitive Radio



Shahbaz Soofi, Anjali Potnis and Prashant Diwivedy

Abstract With the increasing number of wireless users every day, cognitive radio serves as an approach to solve the spectrum crunch problem. Spectrum sensing serves as the heart of cognitive radio with it being able to decide if an unlicensed user is to be given access to a licensed band or not without causing interference. Various spectrum sensing techniques have been discussed in the literature. In this paper, energy detection is considered for spectrum sensing, which conventionally works on a fixed threshold without requiring any prior knowledge of the signal. It is found the performance of conventional energy detection falls in regions with noise uncertainty. In this paper, a dynamic double threshold scheme along with cooperative spectrum sensing at the fusion center is proposed. The dynamic threshold selection works on the parameter of noise uncertainty for practical cases by creating a noise variance history. Also, the fusion center uses a dynamic threshold to make a final decision compared to a fixed threshold for all the energy values lying between the two thresholds at the local nodes. A simulation model has been discussed to compare the proposed scheme with traditional energy detection and other detection schemes as well. A 20% improvement in probability of detection at -22 dB SNR and 0.5 probability of false alarm is achieved using the proposed scheme.

Keywords Cognitive radio · Spectrum sensing · Fusion center · Noise uncertainty · Probability of detection · Probability of false alarm

1 Introduction

The ever-increasing number of users in wireless communication has led to the shortage of existing spectrum. However, it is not the shortage but the underutilization of the spectrum which has led to this problem based on a report by FCC [1]. Cognitive

S. Soofi (✉) · A. Potnis
Department of EEE, NITTTR, Bhopal 462001, Madhya Pradesh, India
e-mail: shahbaz_261@yahoo.com

P. Diwivedy
Department of ECE, Allen House Institute of Technology, Kanpur 208008, India

© Springer Nature Singapore Pte Ltd. 2019
N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_40

radio (CR) has been proposed to accommodate the increasing number of users by providing dynamic spectrum access to unlicensed users called secondary user (SU) to a licensed band. This access is provided based on the fact there are no licensed users called primary user (PU) using that band in this duration to avoid interference [2].

The first usage of the term “cognitive radio” was seen in 1999 and 2000 by Joseph Mitola in [3]. The abilities of cognitive radio allow it to be an intelligent system aware of its surroundings. To make this technology possible, however, the cognitive radio has to sense if the spectrum is available or not and is not currently being held by a PU before giving access to an SU. Thereby placing spectrum sensing at the heart of a cognitive radio network. Such opportunities in which an SU can transmit without creating any interference are called “Spectrum Holes” [4]. Hence the job of CR is to efficiently identify these spectrum holes in which SU can transmit their signals. Also, this allocated band has to be vacated by the SU immediately after its utilization. At no cost, the PU should incur any interference in its transmission.

Spectrum sensing hence allows the SU to detect the presence or absence of a primary user in its band. Various spectrum sensing techniques have been proposed in the literature [5]. Energy detection among all techniques serves as the most simple and widely implemented technique owing to its easy implementation and not requiring any prior knowledge of the signal, unlike matched filter. Cyclostationary and matched filter are techniques explored in literature requiring exact parameters of primary signal leading to increased complexity and computation at the local nodes [6]. Also, this information is always not available.

While considering traditional energy detection for spectrum sensing, the measured energy is compared to a fixed threshold. If the energy is above a fixed threshold it indicates the presence of a PU. However, if the energy lies below this set threshold it indicates the absence of a PU allowing the spectrum to be allocated to an SU. With its simplicity of implementation however traditional energy detection fails in two scenarios. First in a low SNR region, the CR will not be able to differentiate between the presence of signal along with noise or just noise due to a similar PSD. Hence the presence of a fixed threshold will lead to the failure of CR system irrespective of its sensing intervals [7]. Also if the threshold is set to a low value the CR will detect the presence of PUs even in their absence leading to an increased error called probability of false alarm (P_{fa}). This error causes more and more missed opportunities for the SU. On the contrary, if it is set to a high value the SU will not be able to detect the presence of a PU leading to an increased error called probability of miss detection (P_{md}) causing more interference. To mitigate the noise uncertainty problem as well as the threshold value is set to either of the extremes a technique of double threshold is proposed in the literature [8]. Second, the energy detection suffers from hidden node terminal problem and multipath fading. Due to a hidden node, there might not be a direct LOS between the PU and the SU causing the energy level to drop suddenly indicating the absence of a PU, whereas, it may be present. In such a scenario, the SU may start transmitting leading to interference. To overcome this CR can be made to not rely on the decision made by a single secondary node; instead combines decisions made by several nodes at a fusion center and then make a final

decision. In this technique of cooperative spectrum sensing, the fusion center makes a final decision based on certain rules [9].

Owing to noise uncertainty problem, the threshold as we have seen cannot be kept fixed. Also, the double threshold techniques proposed in literature choose fixed thresholds making the spectrum sensing problem more difficult [10]. Defines both thresholds based on parameter ρ called noise uncertainty parameter. It is observed that ρ is kept fixed, however and cannot be varied (0.5) making it impractical for real noise environments. In this paper based on the works of [11, 12] working toward a single dynamic threshold adaption based on noise variance a dynamic double threshold is proposed which takes noise variance history for $N - 1$ nodes and calculates noise uncertainty parameter according to which both thresholds are varied making them dynamic.

Using cooperation scheme to make a final decision in double threshold method is proposed in [8, 13]. Although both schemes show improved detection performance authors fail to address setting of both individual thresholds for cognitive users [14]. For values lying in between the two thresholds referred to as the confused region in case of detection using double threshold, various schemes have been proposed. In [8, 10, 13] if the observed energy values lie in between the two thresholds, then no decision is made and they are forwarded to the fusion center without making any decision on it. The FC center then works on these values to make a final decision based on schemes proposed. For example, in [10], the fusion center combines all such energy values from local nodes and makes a final decision on the presence or absence by comparing the average energy with a fixed threshold. The fusion center will now have K local 1-bit decisions where the SUs were able to make a decision and one decision of the FC for all the values on which no decision was made by the SUs. The FC combines these two decisions to make a final decision. However, these schemes make the use of a fixed threshold on the FC as well. To bring noise uncertainty parameter into the picture at the FC as well, we further propose a dynamic threshold on the FC based on average noise variance.

To summarize the contributions of this paper, we propose a framework for spectrum sensing using a dynamic double threshold based on noise uncertainty parameter. This noise uncertainty parameter was found to be fixed in our research and not varied with different noises obtained at the detector. By being able to vary this parameter, we were able to make both the decision thresholds dynamic and hence more adaptive to noise.

2 System Model

2.1 Traditional Energy Detector

A traditional energy detector works as a squaring and integrating device where the output is observed energy value O_i . This observed value is then compared to a set threshold. For this, consider a deterministic signal say $x(n)$ [15]. The impulse

response of the channel over which the signal is transmitted is say $h(n)$ and $w(n)$ is the additive white Gaussian noise with zero mean and variance σ_n^2 . The spectrum sensing problem is modeled as a binary hypothesis testing problem [16] where the detector has to make a distinction between the following two hypothesis [17]:

$$y(n) = \begin{cases} w(n) & H_0; \text{ PU Absent} \\ x(n)h(n) + w(n) & H_1; \text{ PU Present} \end{cases} \quad (1)$$

H_0 is called the null hypothesis when the spectrum is free from any PU transmission and H_1 is called alternative hypothesis which indicates transmission from the PU. The test statistic is given as follows

$$X = \sum_{n=1}^N |y(n)|^2 \quad (2)$$

where N is the number of samples and X is the received energy of the signal. Received energy is then compared with a single threshold (λ). If it is greater than the threshold H_1 is decided otherwise H_0 as shown in Fig. 1.

While testing for H_0 and H_1 there are two errors which can occur. The first error Probability of false alarm (P_f) will occur when decision H_1 is made but H_0 was true. In this case, the receiver will detect the spectrum to be occupied when in fact it was free leading to missed opportunities for transmission. The other error involved is Probability of miss detection (P_{md}) and will occur if the receiver detects the spectrum to be free and allows SU to transmit signals when in fact it was occupied by a PU leading to interference. A high value of P_f will lead to spectrum underutilization and a high value of P_{md} on another hand will lead to interference. To describe performance, we will consider Probability of detection (P_d) instead of P_{md} .

$$P_d = 1 - P_{md} \quad (3)$$

A trade-off between the two probabilities is made as a decrease in P_{fa} will lead to a decrease in P_d as well and vice versa. This trade-off is made by using the Neyman–Pearson criterion by fixing P_{fa} and focusing on keeping P_d as high as

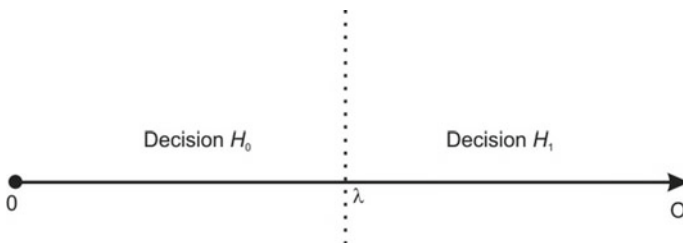


Fig. 1 Single threshold energy detector

possible. This is because at no cost the PU can suffer from interference. This selection process is called constant false alarm rate (CFAR) [18]. Using this approach plot between P_d and P_{fa} called receiver operating characteristics (ROC) is plotted for performance.

The central limit theorem can be applied to approximate the test statistic X as N is usually large. Hence for a fixed threshold λ , the expressions for P_{fa} and P_d can be given as [19]

$$P_{fa} = P(X > \lambda | H_0) = Q\left(\frac{\lambda - N\sigma_w^2}{\sqrt{2N(\sigma_w^4)}}\right) \tag{4}$$

$$P_d = P(X > \lambda | H_1) = Q\left(\frac{\lambda - N(\sigma_n^2 + \sigma_w^2)}{\sqrt{2N(\sigma_n^2 + \sigma_w^2)^2}}\right) \tag{5}$$

where $Q(\cdot)$ is the Gaussian Q function. The probability of error (P_e) will be given by

$$P_e = P_{fa} + P_{md} \tag{6}$$

Using Eq. (4), we can solve for λ

$$\lambda = \sigma_w^2\left(Q^{-1}(P_{fa})\sqrt{2N} + N\right) \tag{7}$$

2.2 Double Threshold Energy Detector

Two thresholds λ_1 and λ_2 are considered based on noise variance. If the observed energy values now say O_i exceed upper thresholds λ_2 decision H_1 is reported and if O_i lies below lower threshold λ_1 decision H_0 is reported. For the energy values lying in between the two thresholds, the SU takes no decision and this value to be reported to the FC as illustrated in Fig. 2.

To define thresholds λ_1 and λ_2 noise uncertainty parameter ρ is introduced where $0 < \rho < 1$. This parameter is based on the practical estimation of noise variance

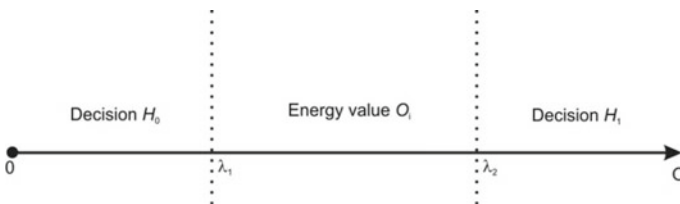


Fig. 2 Double threshold energy detector

and exists in the interval $[(1 - \rho)\sigma_n^2, (1 + \rho)\sigma_n^2]$ taking the varying levels of noise power incurred due to changing location and time into consideration. In this paper, ρ is varied based on average noise variance received at each node, whereas in [10, 14], it is set to 0.5. Using ρ the equations for λ_1 and λ_2 can be given as follows

$$\lambda_1 = (1 - \rho)\lambda \quad (8)$$

$$\lambda_2 = (1 + \rho)\lambda \quad (9)$$

2.3 Cooperative Spectrum Sensing (CSS)

CSS enables a two-stage process for sensing. The local nodes make their decision and send it to an FC [11, 20]. The FC then combines the decision of each SU using either OR rule, AND rule or Majority rule. The optimum decision rule is discussed in [21] taken as OR rule which decides the presence of PU if even one SU reports its presence. For cooperative scheme, the probability of detection (Q_d) and probability of false alarm (Q_{fa}) based on probability of detection (P_{di}) and probability of false alarm (P_{fi}) for i th SUs and N number of sensors is given in [20] as

$$Q_d = 1 - \prod_{i=1}^N (1 - P_{di}) \quad (10)$$

$$Q_f = 1 - \prod_{i=1}^N (1 - P_{fi}) \quad (11)$$

3 Proposed Dynamic Double Threshold Energy Detection

Algorithms discussed in the literature which use fixed threshold need the exact knowledge of noise power which is not possible due to changing noise power. The proposed algorithm works to take the noise uncertainty effect into account based on which the SU and FC both make decisions.

From [12] which uses average noise variance to vary noise uncertainty parameter in determining dynamic single threshold two threshold levels are varied in this scheme. By storing noise variances for $L - 1$ instances, noise variance history is created. For i th instance, the noise variance will be $\sigma_i^2(n)$. For $L - 1$ instances, the average variance can be calculated from

$$\sigma_{\text{avg}}^2 = \frac{1}{L} \sum_{i=1}^L \sigma_i^2(n) \quad (12)$$

Using this stored noise variance history, the maximum noise variance σ_{\max}^2 is calculated from the $L - 1$ nodes and the L th node as well.

$$\sigma_{\max}^2 = \max\{\sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2, \dots, \sigma_L^2\} \quad (13)$$

Then, the noise uncertainty parameter based on average noise variance is defined as

$$\rho_{\text{new}} = \frac{\sigma_{\text{avg}}^2}{\sigma_{\max}^2} \text{ where } 0 < \rho < 1 \quad (14)$$

The dynamic thresholds λ_1 and λ_2 are defined using noise uncertainty parameter ρ_{new} as follows

$$\lambda_1 = (1 - \rho_{\text{new}})\lambda \quad (15)$$

$$\lambda_2 = (1 + \rho_{\text{new}})\lambda \quad (16)$$

Then, the observed energy value O_i is compared to λ_1 and λ_2 . If it is found to be greater than higher λ_2 bit “1” is transmitted to the FC indicating the spectrum is occupied. On the other hand, if O_i is found less than lower threshold λ_1 then bit “0” is reported. If for instance, O_i lies between the two thresholds, the SU makes no decision and reports this value to the FC. At the FC, two kinds of information are hence received the local decisions say “ K ” as “0”s or “1”s and the observed “ $N-K$ ” energy values O_i for N nodes. The FC combines these observed $N-K$ energy values by taking their average and compares with dynamic threshold λ_2 instead of traditional fixed energy detection threshold λ . Here, the upper threshold λ_2 is chosen to increase the reliability and accuracy as λ_2 corresponds to upper noise variance. The simulation model works with the following algorithm:

- Each observed energy value O_i is received at the local node.
- Noise variance is stored and calculated for every observed energy value as $\sigma_i^2(n)$.
- Traditional energy threshold λ is calculated using Eq. (7).
- Calculate σ_{avg}^2 and σ_{\max}^2 using Eqs. (12) and (13).
- Noise uncertainty parameter based on noise variance is found using Eq. (14).
- Dynamic double thresholds λ_1 and λ_2 are defined for every sensing instance using Eqs. (15) and (16).
- Each SU makes a decision (L_i) on the presence or absence of PU by comparing observed energy O_i with λ_1 and λ_2 .

$$L_i = \begin{cases} 0 & O_i < \lambda_1 \\ 1 & O_i > \lambda_2 \end{cases} \quad (17)$$

- If O_i lies in between λ_1 and λ_2 , then the local node makes no decision and reports the value to the FC.

- Assuming the FC receives K hard decisions from N nodes, it will receive $N-K$ energy values. The FC center takes the average of all $N-K$ energy values and calculates O_{avg}

$$O_{\text{avg}} = \frac{1}{N - K} \sum_{i=1}^{N-K} O_i \tag{18}$$

- The FC compares the average energy values of the confused region with upper threshold λ_2 . If the decision is denoted by M then,

$$M = \begin{cases} 0 & O_{\text{avg}} < \lambda_2 \\ 1 & O_{\text{avg}} > \lambda_2 \end{cases} \tag{19}$$

- The FC makes a final decision based on OR rule defined [22] as follows

$$F = \begin{cases} 1 & \text{if } D + M > 1 \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

The decision $F = 1$ corresponds to hypothesis H_1 ; PU present and $F = 0$ corresponds to hypothesis H_0 ; PU absent. The FC reports the decision back to the SU in the backward reporting phase informing the SU if the spectrum is free for transmission or not. From [8], probability of detection P_d^{new} and probability of false alarm P_f^{new} can be derived as

$$P_d^{\text{new}} = P\{O_i > \lambda_2 | H_1\} = Q(\sqrt{2\gamma}, \sqrt{\lambda_2}) \tag{21}$$

where $\gamma = \text{SNR}$ and $Q(a, b)$ is generalized Marcum function.

$$P_f^{\text{new}} = P\{O_i > \lambda_2 | H_0\} = \frac{\chi(\mu, \frac{\lambda_2}{2})}{\chi(\mu)} \tag{22}$$

where $\chi(a, b)$ and $\chi(a)$ are complete and incomplete gamma functions. Also if Q_d^{new} and $Q_{\text{fa}}^{\text{new}}$ represent the probability for detection and false alarm, respectively, for cooperative sensing, then we have

$$Q_d^{\text{new}} = P\{F = 1 | H_1\} \tag{23}$$

$$Q_{\text{fa}}^{\text{new}} = P\{F = 1 | H_0\} \tag{24}$$

The performance of the proposed algorithm is based on the changing values of noise variance for practical environments such as AWGN and Rayleigh Fading Channel. The system focuses on fixing P_{fa} and maximizing the value of P_d .

4 Simulation Results and Analysis

For simulations, AWGN communication channel is considered using BPSK signal. To obtain various samples of energy levels Monte Carlo simulations for 100 samples is run on MATLAB. To prove the superiority of the algorithm in low SNR plots between P_d and P_{fa} called region receiver operating characteristics (ROC) curves are plotted.

Figures 3, 4 and 5 compare the ROC of traditional energy detection and proposed scheme for nodes $N = 10$, $N = 20$ and $N = 50$, respectively, at $SNR = -22$ dB.

Figures 3, 4, and 5 show how the proposed dynamic double threshold can improve the probability of detection by 0.7 in case of 10 SU nodes, 0.8 in case of 20 SU nodes, and 0.9 in case of 50 SU nodes in low $SNR = -22$ dB for the probability of false alarm of 0.2. For the purpose of evaluation of the proposed scheme at various SNRs ROC curve at $SNR = -28$ and -14 dB is shown in Fig. 6 using 10 SUs. Extra detection probability of 0.05 at $P_f = 0.2$ is achieved with improvements in SNR.

In Fig. 7, a comparison of the proposed scheme at varying SNR and changing number of secondary user nodes is made using the ROC curve. Due to prediction nature of the proposed algorithm, large performance improvements are observed in the probability of detection for both increases in the number of SUs (50) and SNR (-4 dB).

In Fig. 8, the effectiveness of the proposed algorithm its performance is compared with other energy detection schemes including conventional energy detection using fixed single threshold and fixed double threshold at $SNR = -22$ dB.

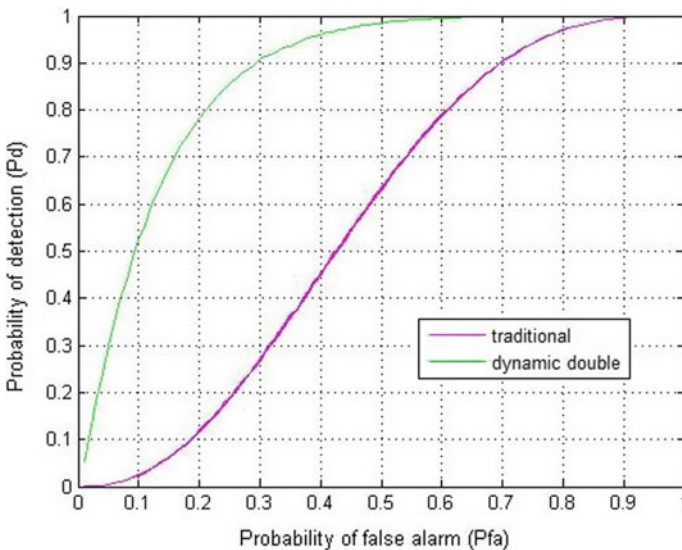


Fig. 3 ROC of traditional energy detection and proposed scheme for nodes $N = 10$

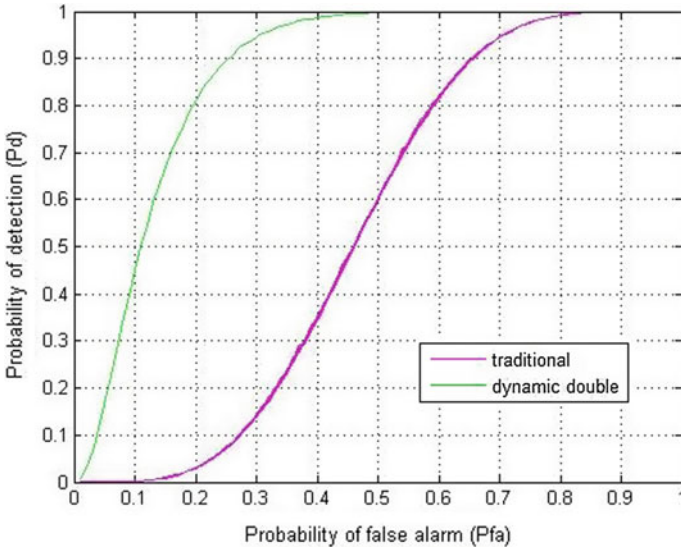


Fig. 4 ROC of traditional energy detection and proposed scheme for nodes $N = 20$

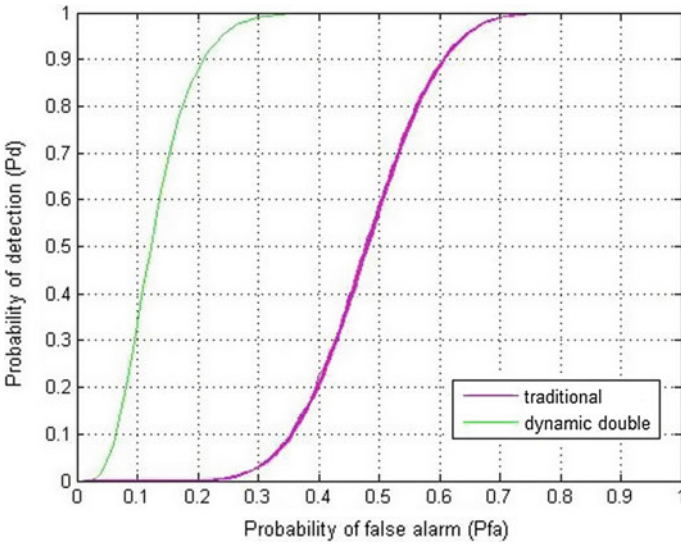


Fig. 5 ROC of traditional energy detection and proposed scheme for nodes $N = 50$

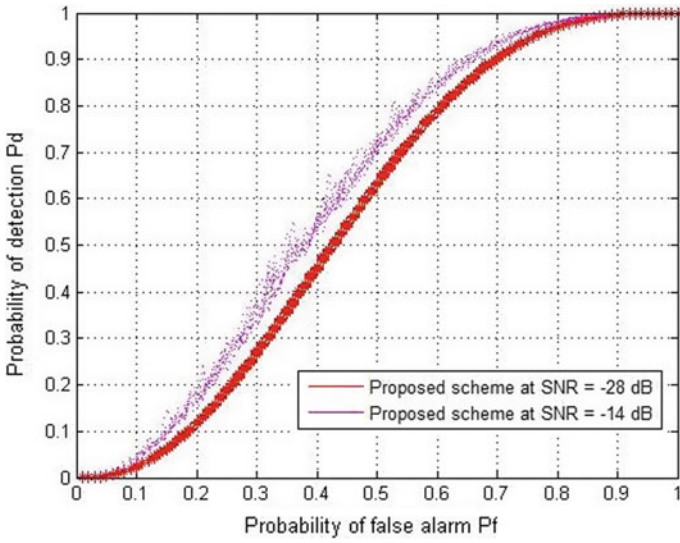


Fig. 6 Proposed scheme ROC at SNR -14dB and -28dB

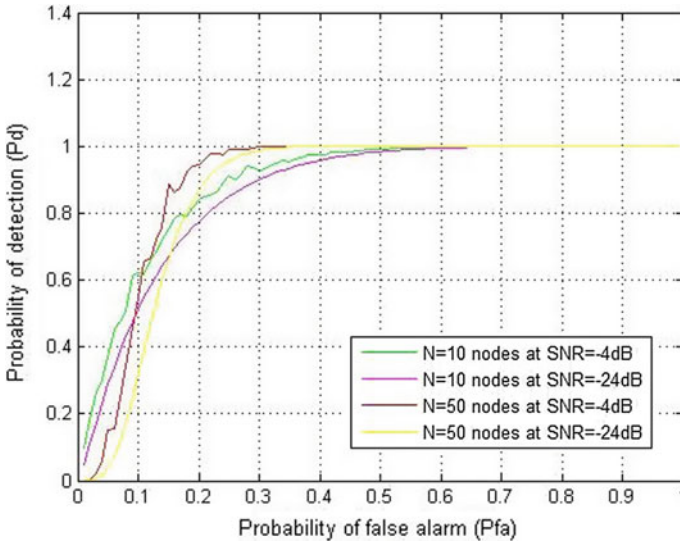


Fig. 7 ROC of the proposed scheme at varying SNR and changing number of secondary user nodes

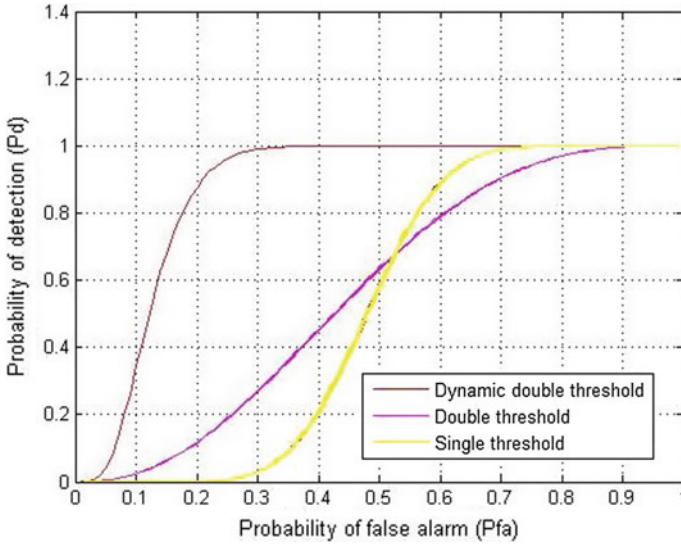


Fig. 8 Comparison of the proposed algorithm with other energy detection schemes at $\text{SNR} = -22\text{dB}$

It is interesting to note the proposed scheme offers improved P_d even at low SNR scenarios using cooperative scheme. Figure 8 shows high P_d up with 20% improvement at -22 dB SNR for $0.5 P_{fa}$ compared to other conventional schemes proposed in the literature.

5 Conclusion

In this paper, we have presented a dynamic selection of double thresholds which are varied using noise variance history. This proposed scheme outperforms other traditional energy detection schemes offering improvement of up to 20% for $0.5 P_{fa}$ at -22 dB SNR. This improves the performance of spectrum sensing, especially in low SNR. Also for energy values lying in between the two thresholds, the FC combines all these values and compares them with higher dynamic double threshold compared to fixed threshold enabling the FC to vary its threshold with previous noise instances. The storing and forwarding of energy values to FC will lead to increase in complexity, however, the sensing performance increases by 3.5 times at $P_{fa} = 0.2$ for this small increase in complexity.

In future, the impacts of the proposed algorithm are to be evaluated for fading channels. Furthermore, the tradeoffs between an increase in sensing performance and complexity will be evaluated using practical implementations.

References

1. Federal Communications Commission. (2002, November). Spectrum policy task force report, FCC 02-155.
2. Haykin, S. (2005). Cognitive radio: Brain-empowered wireless communications. *IEEE Journal on Selected Areas in Communications*, 23, 201–220.
3. Mitola, J., & Maguire, G. Q. (1999). Cognitive Radio: Making software radios more personal. *IEEE Personal Communications*, 6(4), 13–18.
4. Tandra, R., Sahai, A., & Mishra, S. M. (2009). What is a spectrum hole and what does it take to recognize one. *Proceedings of the IEEE*, 97(5), 824–848.
5. Yucek, T., & Arslan, H. (2009). A survey of spectrum sensing algorithms for Cognitive radio applications. *IEEE Communications Surveys and Tutorials*, 11(1), 116–130 (first quarter).
6. Sutton, P. D., Nolan, K. E., & Doyle, L. E. (2008). Cyclostationary signatures in practical cognitive radio applications. *IEEE Journal on Selected Areas in Communications*, 26(1), 13–24.
7. Tandra, R., & Sahai, A. (2008, February). SNR Walls for Signal Detection. *IEEE Journal of Selected Topics in Signal Processing*, 2, 4–17.
8. Zhu, J., Xu, Z., & Wang, F. (2008, May). Double threshold energy detection of cooperative spectrum sensing in cognitive radio. In *Proceedings of the 3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom 2008)*.
9. Sun, C. H., Zhang, W., Ben Letaief, K. (2007, March). Cooperative spectrum sensing for cognitive radios under bandwidth constraints. In *IEEE Conference on Wireless Communications and Networking (Vols. 1–9, pp. 1–5)*.
10. Plata, D. M. M., & Reátiga, Á. G. A. (2012). Evaluation of energy detection for spectrum sensing based on the dynamic selection of detection-threshold. *Procedia Engineering*, 35, 135–143.
11. Joshi, D. R., Popescu, D. C., & Dobre, O. A. (2010, March). Adaptive spectrum sensing with noise variance estimation for dynamic cognitive radio systems. In *Proceedings of the 44th Annual Conference on Information Sciences and Systems (CISS 2010)*.
12. Farag, H. M., & Ehab, M. (2014). An efficient dynamic thresholds energy detection technique for cognitive radio spectrum sensing. In *International Conference on Computer Engineering (pp. 139–144)*.
13. Vien, Q. T., Nguyen, H. X., & Le-Ngoc, T. (2014). An efficient hybrid double-threshold based energy detection for cooperative spectrum sensing. In *2014 27th Biennial Symposium on Communications (QBSC)*, Kingston, ON (pp. 42–46).
14. Urkowitz, H. (1967). Energy detection of unknown deterministic signals. *IEEE Proceedings*, 55(4), 523–531.
15. Ghasemi, A., Sousa, E.S. (2005, November). Collaborative spectrum sensing for opportunistic access in fading environment. In *Proceedings of IEEE DySPAN 2005* (pp. 131–136).
16. Cabric, D., Tkachenko, A., & Brodersen, R. W. (2006, August). Experimental study of spectrum sensing based on energy detection and network cooperation. In *Proceedings of the First International Workshop on Technology and Policy for Accessing Spectrum (TAPAS 2006)*.
17. Poor, H. V. (1994). *An introduction to signal detection and estimation* (2nd ed.). Berlin: Springer.
18. Nuttall, A. H. (1975). Some integrals involving the QM function. *IEEE Transactions on Information Theory*, 21(1), 95–99.
19. Xuping, Z., & Jianguo, P. (2007, December). Energy-detection based spectrum sensing for cognitive radio. In *IET Conference on Wireless, Mobile and Sensor Networks* (pp. 944–947).
20. Zhang, W., Mallik, R. K., & Letaief, K. B. (2009). Optimization of cooperative spectrum sensing with energy detection in cognitive radio networks. *IEEE Transactions on Wireless Communication*, 8(12), 5761–5766.
21. Sun, C., Zhang, W., Letaief, K. B. (2007). Cluster-based cooperative spectrum sensing in cognitive radio systems. In *Proceedings of IEEE ICC'07* (pp. 2511–2515).

22. Dean, J., & Ghemawat, S. (2008, January). Mapreduce: Simplified data processing on large clusters. *ACM Communications*, *51*, 107–113.
23. Verma, P., & Singh, B. (2015). Simulation study of double threshold energy detection method for cognitive radios. In *2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN)*, Noida (pp. 232–236).

ConvFood: A CNN-Based Food Recognition Mobile Application for Obese and Diabetic Patients



Kaiz Merchant and Yash Pande

Abstract In recent years, obesity and health of diabetic patients have become major issues. To address these issues, it is very important to know the intake of calories, carbohydrates, and sugar. We propose a novel deep learning convolutional neural network based image recognition system that can run on Android smartphones that not only provides the appropriate nutritional estimates to users after passing a food image as input but also suggests alternative food recipes for diabetic patients. We have implemented transfer learning as well as fine-tuning and our CNN model was able to achieve comparatively higher accuracy than other approaches that used a similar setup on the Food-101 dataset. By user experiments and approval from well-known doctors, effectiveness of the proposed system was confirmed. The future scope includes expanding to more food categories and optimizing the model for better results.

Keywords Convolutional neural networks · Deep learning · Obesity · Diabetic patients · Smartphone

1 Introduction

Due to better standards of living, obesity rates are getting higher and its impact is increasing [1]. To keep a check on this, people must have a daily estimate of the calories that they consume in every meal. Also, diabetic patients are assigned specific quantities of sugar and carbohydrates that they are allowed to eat by their doctors. However, before eating a meal they often do not know the amount of carbohydrates and sugar that they are about to consume. Furthermore, even if a diabetic patient can see the nutritional estimates, learning that he or she cannot consume the meal may alter his or her sentiment in a negative way. Thus, alternative healthier options

K. Merchant · Y. Pande (✉)

Dwarkadas J. Sanghvi College of Engineering, Mumbai, India
e-mail: yash3096@gmail.com

K. Merchant

e-mail: merchant.kaiz@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_41

appropriate for diabetic patients should be made available to them. Due to recent trends in technology and increased usage of smartphones, many mobile applications have been developed that record everyday meals. Many of them ask the users to give several text inputs and assume that the users are ready to undergo a tedious process before getting the required output such as in [2–4]. Some applications have aimed to carry out the process using food images. However, most of them have problems related to usability. Also, they do not provide alternative options for diabetic patients. Our proposed system aims to correct both the above issues by providing a mobile application that can estimate calories, sugar, and carbohydrate levels from food images and provide healthier options for diabetic patients.

2 Background

Many methods have been suggested in the literature for recognizing food from images using hand-engineered features along with traditional image processing techniques [5–7]. Yang et al. [5] proposed to learn spatial relationships between ingredients for 61 food categories using pairwise features followed by feature fusion which is bound to work only for standardized meals. Matsuda et al. [6] used miscellaneous ranking based approach for recognizing multiple foods. However, this type of solution is computationally intensive and may not be practically deployable within the mobile cloud computing platform. Bossard et al. [8] reported classification accuracy of 50.76% on test set by mining discriminative components using random forests. Clustering the superpixels of training dataset was done using random forest approach to train the component models. Random forest was discarded after mining and during classification multiclass SVM with spatial pooling predicted the final class. Here, the main problem was with noisy training images not being cleaned and the accuracy of predictions being very low comparatively. Liu et al. [9] successfully applied Inception-model-based CNN approach to two real-world food image data sets (UEC-256 and Food-101).

There have been a number of available applications that measure daily food's dietary information but most of these approaches involve lots of tedious manual operations to be performed. One example, which is typical of current clinical approaches, is the 24-hour dietary recall [10]. The issue with this approach is that people have to remember what they have eaten for a long period and consult a dietician very frequently. Pouladzadeh [11] proposed a mobile-based application for measuring calorie and nutrition. The measurement system used the patient's thumb for first-time calibration which was then used to calculate the amount of food consumed. However, the problem is that the density of the food is equally important for calculating the calorific value. Thus, although many of these approaches have achieved highly impressive accuracies and tried to measure nutritional value from fast food recognition and their volume calculation, none of them has aimed to make life of a diabetic patient easier.

3 Proposed System

3.1 Dataset and Image Preprocessing

We have used ETHZ Food-101 dataset for training which was newly introduced by Bossard et al. [8]. It contains 101,000 labeled images of food dishes which belong to 101 food categories. Each category has 1000 images which are already rescaled to have maximum 512 pixels side length. These are the most 101 popular food categories all over the world which have been sourced from food picture sharing website named foodspotting.com [12]. We initially make use of the histogram equalization algorithm to increase the contrast and luminance of images [13]. We implemented this using the OpenCV library in Python. Moreover, before giving images as input to the neural network, image augmentation is carried out for entire dataset which includes rotation, flipping, and scaling the images to 299×299 pixels size. Logarithmic correction [13] has also been applied to input images to enhance the low pixel values with not much loss of information in high pixel values.

3.2 Food Segmentation Using CNN

Initially, we were motivated to use the bag-of-words approach [14] for the food recognition task. However, CNNs perform the same task as the bag-of-words approach using a much deeper understanding of the input data [15]. The working of CNNs [16] can be compared to the functioning of the human brain. A CNN consists of convolution, pooling, and ReLU layers. A convolution layer searches for features everywhere in an image by making use of a filter. The work of a pooling layer is to reduce the size of the image making sure that important features are preserved. The rectified linear unit eliminates negative values.

Different model architectures exist under the domain of CNNs. These include LeNet-5 [17], which successfully carried out digit recognition before other ones did, AlexNet [16], which is similar to but deeper than LeNet, VGG16 [18], which uses a very large number of parameters for training, ResNet50 [19], that makes use of skip connections for a deeper network and Inception [20] that combines different convolutions and reduces the total number of parameters. In our system, we make use of the Inception network which is shown in Fig. 1. The Inception module is based on the concept of pattern recognition. After passing several images as input, the network starts getting used to the small details. As we can see, the Inception network allows the model to take advantage of all of the convolutions simultaneously. We make use of the Inception V3 network specifically, which is a 48 layer deep network as compared to the Inception V1 network which is only 22 layers deep.

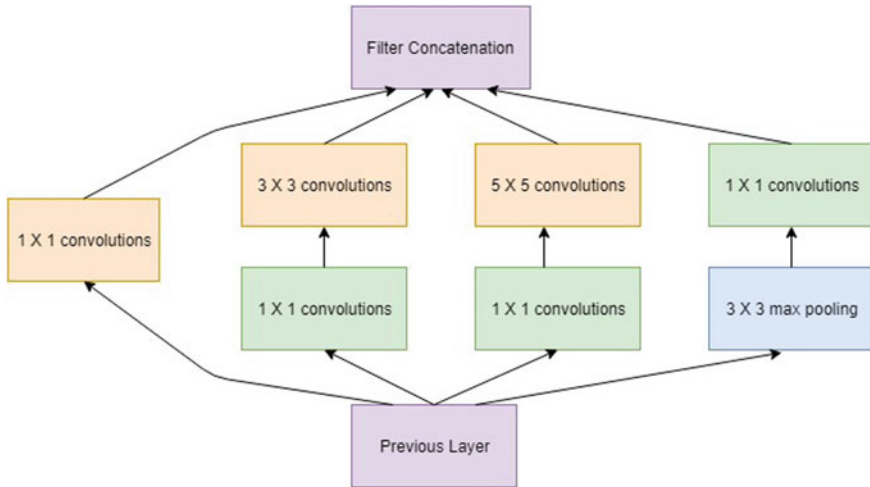


Fig. 1 Inception module

3.3 *Nutritional Estimation*

Nutritional values for food categories are fetched from the USDA Food Composition Database [21]. United States Department of Agriculture Agricultural Research Service provides us with lab-tested nutritional estimates for different food groups. First, it gives us nutrient reports which provide lists of food dishes pertaining to different brands and their respective nutrient values for a specified set of nutrients. Second, it provides us with food reports which list nutrient values including calories, fats, carbohydrates, protein, and sugar values per 100 grams for a queried food dish from the particular food outlet. The nutrient data is obtained after laboratory tests were conducted on the USDA branded food products database and is thus a certified trusted source. Being in the public domain there are no permissions needed to use the USDA food composition data. The suggested citations are mentioned in the references [22, 23].

3.4 *Alternative Recipe Recommendation*

If diabetic patients feel that they cannot consume food safely after getting the nutritional estimates, they have an option to view alternative healthier versions of the identified junk foods. This will satisfy their taste buds without compromising their health. We obtain multiple low and complex carbohydrate recipe recommendations from Yummly [24]. Ingredients for these are also listed so that patients know what option suits them best. Moreover, Yummly's patent-pending food intelligence tech-

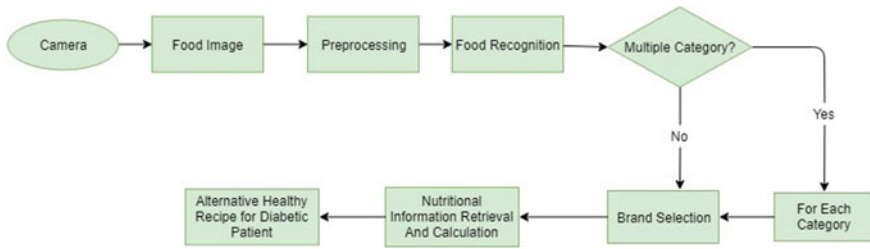


Fig. 2 Overall system flow

nology guides users with images and directions to prepare food dishes easily at home. The nutritional facts for these alternative recipes are calculated using the USDA nutrient database and displayed for one serving. Thus, both our initial estimation and alternative recipe nutrition counts reference the same database, and hence the link between modules is justified.

As shown in Fig. 2, the user takes an image of a food item from an Android device having active Internet connection. This image is sent to the server, where our trained CNN model returns predicted probabilities and the food category is recognized. The user then views the lab-tested nutritional values of selected brand and diabetic patients can further see a list of alternative healthy recipes.

4 Experimental Results

In the following subsections, we present our experimental setup, model comparisons, and running application analysis.

4.1 Experimental Setup

TensorFlow provides us with a publicly available Inception V3 based model that was pretrained on the large ImageNet dataset that contains 1000 classes. Image recognition with convolutional neural networks on small datasets works better when fine-tuned on such existing networks. This helps to generalize and prevent overfitting. Thus, we fine-tune this network and adjust the dimensions of the last softmax layer to fit our dataset which consists of 101 food categories. The experiment was conducted using a learning rate of 0.01, which ensures that the training converges at an optimal point with a moderate pace. We ran the code for about 100,000 iterations using 80% of the images for training and the remaining 20% for testing. We selected a batch size of 32 and used the Adam optimizer, which was chosen using a trial and error procedure.

4.2 Results and Analysis

The cross-entropy loss, which is a good pick for classification tasks, decreased over time from 4.5 to 0.96. Initially, as the number of iterations increased, the loss decreased significantly; however, above 20,000 iterations the decrease in loss became steadier. Similarly, the accuracy increased slowly after 20,000 iterations. However, it still did not converge, which shows that the model would have performed even better if we ran the code for more number of iterations. We were finally able to achieve a testing accuracy of about 70% for the food recognition task. The accuracy and loss curves are presented in Figs. 3 and 4, respectively, with a smoothing of 0.9.

The server corresponding to the Android application responds with either one or multiple categorical probabilistic predictions within 5 s. The user is then accordingly provided a display of different food groups and can view the lab-tested nutritional values instantaneously for each of them. No other manual operations are required and all this happens with just the click of an image. Furthermore, diabetic patients discover unprecedented healthier version/recipes of the classified food. The ingredients displayed help diabetic patients to strictly obey complex low carb diets. All this functionality comes in a small Android application of size 6.9 megabytes and

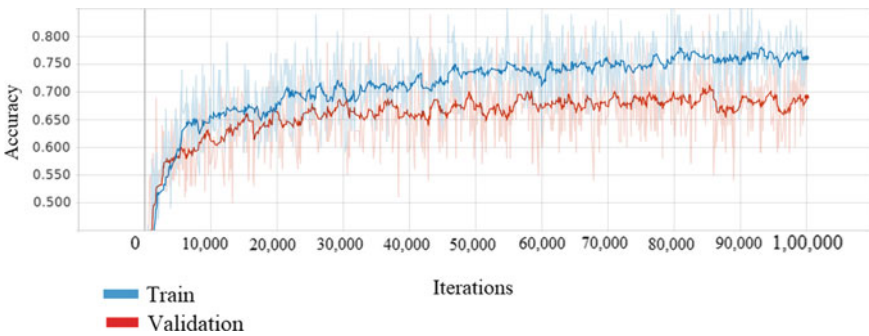


Fig. 3 Accuracy of each step in fine-tuned Inception V3 network

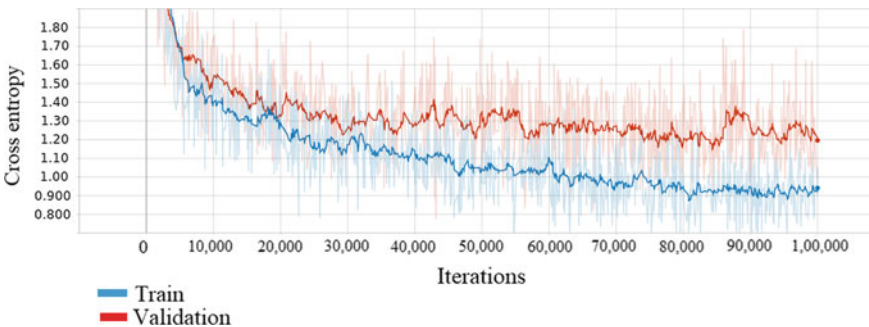


Fig. 4 Cross-entropy loss of each step in fine-tuned Inception V3 network

can easily work on Android devices supporting versions greater than 4.0 (Ice Cream Sandwich). Finally, the proposed system was approved and provided with positive feedback from renowned doctors.

4.3 Model Comparisons

Before convolutional neural networks came into the picture, the various approaches tried out for this task were not able to reach even a moderately high accuracy. In fact, as can be seen in the comparison table, most of the accuracies were below 50%. AlexNet, which contains five convolution layers, was able to achieve a decent high accuracy on the food recognition problem. Among other CNNs, the Inception and GoogLeNet models have been able to outperform all other ones.

In our proposed system, we have made use of the Inception module, precisely the Inception V3 network. The reason why we are able to perform better than most other methodologies is due to the advantages of the way the Inception network is built. There are three convolution layers to which the input is fed—the $1 * 1$ layer, the $3 * 3$ layer, and the $5 * 5$ layer. Due to this, the network is able to not only capture the general features but also the local ones. Also, because of this structural benefit, the depth of the module increases without increase in the number of dimensions. The model is able to perform well and achieve a considerably high accuracy. However, Liu et al. [9] were able to achieve an accuracy of about 77%, which is some percent higher than the proposed model. This is mainly due to the fact that they were able to run their program for very large number of iterations. We were not able to do the same due to the lack of resources available. Similarly, Yu et al. [25] were able to get an output whose value is very close to ours. However, they were able to reach this accuracy when retraining all layers of the Inception network, while we have retrained only the final layer. In comparison to other models that were retrained only on the final layer, our model performs significantly better as can be seen from Table 1.

Table 1 Comparison of various models with accuracies

Method	Top-1 accuracy (%)
Bag-of-words histogram [8]	28.51
Improved fish vectors [8]	38.88
Random forest classification [8]	32.72
Randomized clustering forests [8]	28.46
Mid-level discriminative superpixels [8]	42.63
Discriminative components with random forests [8]	50.76

(continued)

Table 1 (continued)

Method	Top-1 accuracy (%)
GoogleLeNet [9] (10,000 steps)	70.2
GoogleLeNet [9] (300,000 steps)	77.4
Inception V3 [25] (last layer training)	35.32
Inception V3 [25] (full layer training)	70.60
Inception-Resnet [25] (last layer training)	42.69
Inception-Resnet [25] (full layer training)	72.55
AlexNet [26]	66.40
MLDS [27]	42.63
Inception V3 (our model) (last layer training)	70

5 Conclusion

Obesity and diabetes are two of the most common health issues in today's world. To tackle these issues, we need a technical assistant that can help us get the nutritional content of everyday meals that we eat. To solve this problem, we first develop a novel algorithm for food category recognition which is based on convolutional neural networks. We perform our experiment on the challenging Food-101 dataset and are able to achieve a comparatively high accuracy with lesser model parameters and retraining. However, preprocessing the image with logarithmic correction and histogram equalization did not help in improving accuracy, which was mainly due to the fact that most of the feature recognition task had already been carried out by the starting layers of the pretrained network. Second, to make the life of users easier, we incorporate our model into a mobile application that provides nutritional quantities for each image of food clicked and helps assist diabetic patients by providing them with alternative food recipes. In the future, we hope to incorporate more datasets into our system in order to increase the number of food categories. We also aim to further improve the model accuracy and expand our mobile application to provide options for patients of other health disorders.

Acknowledgements We would like to thank Doctor Harshal Joshi, M. D. and Doctor Sanjay Gulhane, M. D. for their support and guidance.

References

1. Hruby, A., & Hu, F. B. (2015). The epidemiology of obesity: A big picture. *PharmacoEconomics*, 33, 673–689. <https://doi.org/10.1007/s40273-014-0243-x>.
2. MyFitnessPal.com: Free Calorie Counter, Diet & Exercise Tracker, available at <http://www.myfitnesspal.com/>.

3. MyNetDiary: The easiest and smartest free calorie counter and free food diary for iPhone, iPad, Android, and BlackBerry applications, available at <http://www.mynetdiary.com/>.
4. FatSecret: All Things Food and Diet, available at <http://www.fatsecret.com/>.
5. Yang, S., Chen, M., Pomerleau, D., & Sukthankar, R. (2010). Food recognition using statistics of pairwise local features. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. <https://doi.org/10.1109/cvpr.2010.5539907>.
6. Matsuda, Y., & Yanai, K. (2012). Multiple-food recognition considering co-occurrence employing manifold ranking. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR 2012)*, Tsukuba (pp. 2017–2020).
7. TADA: Technology Assisted Dietary Assessment at Purdue University, West Lafayette, Indiana, USA, available at <http://www.tadaproject.org/>.
8. Bossard, L., Guillaumin, M., & Van Gool, L. (2014). Food-101—Mining discriminative components with random forests. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer Vision—ECCV 2014. ECCV 2014. Lecture Notes in Computer Science* (Vol. 8694). Cham: Springer. https://doi.org/10.1007/978-3-319-10599-4_29.
9. Liu, C., Cao, Y., Luo, Y., Chen, G., Vokkarane, V., & Ma, Y. (2016, May). DeepFood: Deep learning-based food image recognition for computer-aided dietary assessment. In *International Conference on Smart Homes and Health Telematics* (pp. 37–48). Springer International Publishing. https://doi.org/10.1007/978-3-319-39601-9_4.
10. Livingstone, M., Robson, P., & Wallace, J. (2004). Issues in dietary intake assessment of children and adolescents. *British Journal of Nutrition*, 92(S2), S213–S222. <https://doi.org/10.1079/BJN20041169>.
11. Pouladzadeh, P., Shirmohammadi, S., & Almaghrabi, R. (2014, August). Measuring calorie and nutrition from food image. *IEEE Transactions on Instrumentation and Measurement*, 63, 1974–1956. <https://doi.org/10.1109/tim.2014.2303533>.
12. Foodspotting: find and share great dishes, available at <http://www.foodspotting.com/find>.
13. Chaudhury, S., Raw, S., Biswas, A., & Gautam, A. (2015). An integrated approach of logarithmic transformation and histogram equalization for image enhancement. In K. Das, K. Deep, M. Pant, J. Bansal, & A. Nagar (Eds.), *Proceedings of Fourth International Conference on Soft Computing for Problem Solving. Advances in Intelligent Systems and Computing* (Vol. 335). New Delhi: Springer. https://doi.org/10.1007/978-81-322-2217-0_6.
14. Gao, H., Dou, L., Chen, W., & Sun, J. (2013). Image classification with Bag-of-Words model based on improved SIFT algorithm. In *2013 9th Asian Control Conference (ASCC)*, Istanbul (pp. 1–6). <https://doi.org/10.1109/ascc.2013.6606268>.
15. Okafor, E., et al. (2016). Comparative study between deep learning and bag of visual words for wild-animal recognition. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, Athens (pp. 1–8). <https://doi.org/10.1109/ssci.2016.7850111>.
16. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *NIPS* (p. 4).
17. Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324. <https://doi.org/10.1109/5.726791>.
18. Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *ICLR*. [arXiv:1409.1556](https://arxiv.org/abs/1409.1556).
19. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV (pp. 770–778). <https://doi.org/10.1109/cvpr.2016.90>.
20. Szegedy, C., et al. (2015). Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA (pp. 1–9). <https://doi.org/10.1109/cvpr.2015.7298594>.
21. United States Department of Agriculture Agricultural Research Service Food Composition Databases: available at <https://ndb.nal.usda.gov/ndb/>.
22. U.S. Department of Agriculture, Agricultural Research Service. 20xx. USDA National Nutrient Database for Standard Reference, Release. Nutrient Data Laboratory Home Page, <http://www.ars.usda.gov/nutrientdata>.

23. U.S. Department of Agriculture, Agricultural Research Service. 20xx. USDA Branded Food Products Database. Nutrient Data Laboratory Home Page, <http://ndb.nal.usda.gov>.
24. Yummly: personalized recipe recommendation and search, available at <https://www.yummly.com/recipes>.
25. Yu, Q., Mao, D., & Wang, J. Deep Learning Based Food Recognition.
26. Ao, S., & Ling, C. X. (2015, November). Adapting new categories for food recognition with deep representation. In *Proceedings of the IEEE International Conference on Data Mining Workshop* (pp. 1196–1203). <https://doi.org/10.1109/icdmw.2015.203>.
27. Singh, S., Gupta, A., & Efros, A. A. (2012). Unsupervised discovery of mid-level discriminative patches. In *Computer Vision–ECCV 2012* (pp. 73–86). Berlin: Springer. [arXiv:1205.3137](https://arxiv.org/abs/1205.3137).

Segmentation and Recognition of *E. coli* Bacteria Cell in Digital Microscopic Images Based on Enhanced Particle Filtering Framework



Manjunatha Hiremath

Abstract Image processing and pattern recognitions play an important role in biomedical image analysis. Using these techniques, one can aid biomedical experts to identify the microbial particles in electron microscopy images. So far, many algorithms and methods are proposed in the state-of-the-art literature. But still, the exact identification of region of interest in biomedical image is a research topic. In this paper, *E. coli* bacteria particle segmentation and classification is proposed. For the current research work, the hybrid algorithm is developed based on sequential importance sampling (SIS) framework, particle filtering, and Chan–Vese level set method. The proposed research work produces 95.50% of average classification accuracy.

Keywords Image segmentation · Sequential importance sampling · Particle filtering · Chan–Vese level set method · Minimum distance classifier

1 Introduction

The image processing and pattern recognitions are related to algorithmic development and putting abstract region of interest into categories. Generally, the categories are expected to be known in advance as supervised learning. There are methods and methodologies to understand the clusters. Current days, the pattern recognition is useful in so many scenarios (Applications) such as information processing, document image exploration and identification, forensics, biometric analysis, and bioinformatics analysis.

Present days, so many real-world problems are automated with pattern recognition system, where tedious tasks are made easy. For example, in a chemical industry, checking and verifying the chemical product by human beings are too dangerous for some extent. By automating the process by computer vision and pattern recognition overcomes such problems. Biomedical imaging is current buzzword in medical industry.

M. Hiremath (✉)

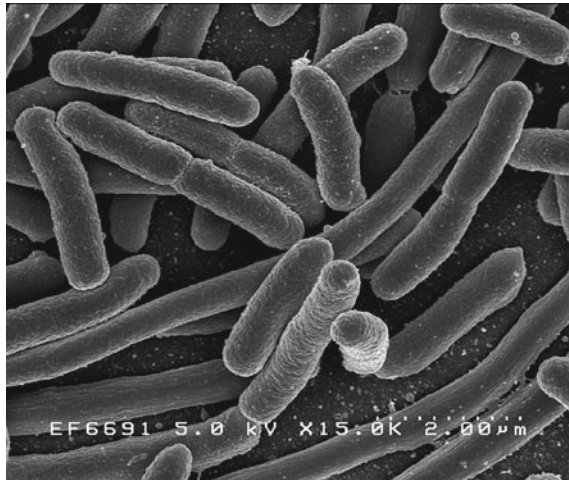
Department of Computer Science, CHRIST (Deemed to be University), Bengaluru, India
e-mail: manju.gmtl@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,

https://doi.org/10.1007/978-981-13-5953-8_42

Fig. 1 Microscopic (SEM) image of *E. coli* [11]



As a part and partial of biomedical imaging and processing, microscopy images are also used. In the current study, the *E. coli* microscopy images are considered. *Escherichia coli* are a gram-negative, facultative anaerobic, rod-shaped and coliform bacterium of the genus *Escherichia* that is commonly found in the lower intestine of warm-blooded organisms (endotherms). Figure 1 shows the generic microscopic image of *E. coli*.

2 Materials and Methods

Before proceeding to the computational model, it is necessary to develop a dataset of stained *E. coli* images. This section presents abstract technique of staining method of bacterial gram staining method. This technique is widely used bacteriological preparation system for microscopic vision. These particular staining methods are proposed by Dr. Christian Gram in the period of 1884.

In the current research work, Gram-negative staining of *E. coli* bacteria is considered. Figure 2 presents the typical staining methodologies.

3 Proposed Method

- *Sequential Importance Sampling (SIS) Framework:*

The framework requires only one user interaction on the slice where vessel tracking begins. In the initialization step, the particles and their weights are initialized using the seed. In the proposed framework, the user needs to roughly select a region

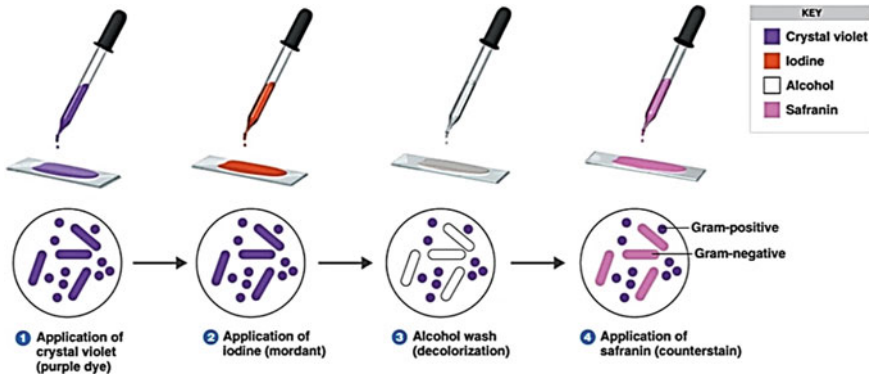


Fig. 2 The staining general method used for microorganisms

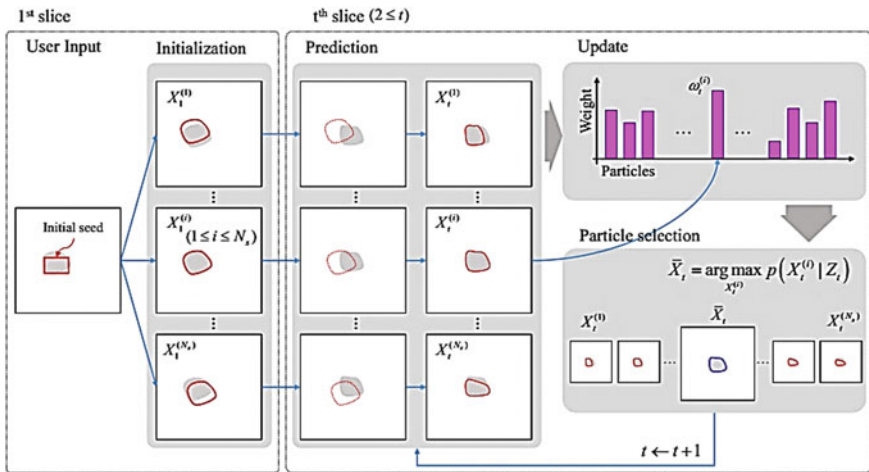


Fig. 3 SIS PF framework with four steps: initialization, prediction, update and particle selection

around the vessel on the first slice. Then, the particles are initialized based on the contour obtained by exploiting the CV algorithm, and the particle weights are set uniformly. In the prediction step, each particle (contour) moves toward the vessel boundary on the consecutive slice based on the dynamics. In the update step, the particle weights are updated based on the particles predicted for that slice. In the proposed method, the CV algorithm is used to predict the particle changes on the slice. By combining the updated weights and predicted particles, a particle with the maximum a posteriori (MAP) probability is selected in the particle selection step. These prediction, update and particle selection steps are repeated for the following consecutive slices to track along the vessel until vessel tracking terminates [1–9] (Fig. 3).

- *Particle Filtering:*

The particle filtering is a progressive Monte Carlo framework in which the crucial indication is to characterize the essential subsequent density function by a group of random samples with related loads (weights). Here, let X_t and Z_t be the state and measurement, respectively, at time index 't'. The unknown state is usually estimated by the posterior density function $p(X_t|Z_{1:t})$ which can be calculated recursively using the following Bayesian formula:

$$p(X_t|Z_{1:t}) \propto p(Z_t|X_t)Z_p(X_t|X_{t-1})p(X_{t-1}|Z_{1:t-1})dX_{t-1}.$$

In the PF framework, it is approximated recursively using a set of particles. Let N_s be the number of particles, and $n \in \{1, \dots, N_s\}$ be the set of particles and corresponding weights describing their relevance [10]. Then, it is approximated as follows:

$$p(X_t|Z_{1:t}) \simeq \sum_{n=1}^{N_s} \omega_t^{(n)} \delta(X_t - X_t^{(n)})$$

where $\delta(\cdot)$ denotes the Dirac delta measure. With this particle approximation, object tracking is performed by the following procedure:

Prediction: move the swarm of particles based on the dynamics as

$$X_t^{(n)} \sim q(X_t|X_{t-1}^{(n)}, Z_t).$$

Update: calculate the particle weights based on the likelihood as

$$\omega_t^{(n)} \propto \omega_{t-1}^{(n)} \cdot p(Z_t|X_t^{(n)})$$

where $q(\cdot)$ is the importance density. Note that the selection of importance density significantly affects PF tracking accuracy. The above-mentioned PF is known as the sequential importance sampling (SIS) algorithm [11]. If the importance function is set to $q(X_t|X_{t-1}, Z_t) = p(X_t|X_{t-1}, Z_t)$, the SIS algorithm becomes a bootstrap filter, which is the most common choice. The main advantage of a bootstrap filter is its simple implementation because it only requires sampling from the distribution $p(X_t|X_{t-1}, Z_t)$ and the evaluation of $p(Z_t|X_t)$. However, it incurs the following disadvantages.

- (1) The prediction step uses only the previous state and does not consider the observed information, and
- (2) A degeneracy problem can arise. In other words, after a few iterations, most particles have negligible weights; thus significant computational effort is required to update particles with a very small contribution to posterior density.

- *Chan–Vese Level Set Method:*

For a variational level set formulation, Chan and Vese proposed region-based image segmentation [12]. This approach has grown into very widespread in the field of digital image processing community principally due to its capability to identify objects not necessarily defined by a gradient. The basic concept of the segmentation method is to divide a particular partition of a given image I into two regions, where one region represents the objects to be detected and the other region represents the background. Then, the contour of the object is defined as the boundary between these two regions [13]. For a given image I , they proposed minimizing the following energy functional:

$$E_{CV}(\Phi, I) = \lambda_1 \int_{\Omega} (I - c_1)^2 H(\Phi) dx dy + \lambda_2 \int_{\Omega} (I - c_2)^2 (1 - H(\Phi)) dx dy + \nu \int_{\Omega} |\nabla H(\Phi)| dx dy$$

where Ω is the image domain, and λ_1, λ_2 and ν are positive, user-defined eights. In addition, c_1, c_2 and $H(\Phi)$ are defined as follows:

$$c_1 = \frac{\int I(x, y) H(\Phi) dx dy}{\int H(\Phi) dx dy}, c_2 = \frac{\int I(x, y) (1 - H(\Phi)) dx dy}{\int (1 - H(\Phi)) dx dy},$$

$$H(\Phi) = \begin{cases} 1 & \Phi \geq 0 \\ 0 & \text{Otherwise} \end{cases}$$

where $I(x, y)$ is the pixel intensity and Φ is the level set function. If we regularize $H(x, y)$ and $\delta(x, y)$ using suitable smooth functions, such as $H_{\epsilon}(\cdot)$ and $\delta_{\epsilon}(\cdot)$, (5) can be minimized using a calculus of variations. The resulting Euler–Lagrange equation is given as follows:

$$\frac{\partial \Phi}{\partial t} = \delta_{\epsilon}(\Phi) \left[\mu \cdot \operatorname{div} \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) - \nu - \lambda_1 (I - c_1)^2 + \lambda_2 (I - c_2)^2 \right].$$

Then, the contour is deformed to the desired boundary using repetitive iterations of (7) until the energy reaches its local minimum point, or its iteration number (L) does not exceed the predefined maximum iteration number (L_{\max}). The CV algorithm has many advantages, such as the ability to perform topology variation of the contour automatically and to stabilize global region information responses to local variations such as weak edges and noise [13–16]. However, the performance of the level set method is limited due to the computational cost of embedding the contour in higher dimensional space. In particular, for object tracking, it is difficult to predict drastic shape changes of the object because the algorithm does not incorporate motion dynamics between frames into the tracking frameworks.

4 Experimental Results and Discussion

The proposed experiment is carried out using 50 digital images of different types of *E. coli* particles of transmission electron microscopy. The experimentation is done using an Intel quadcore processor @ 2.03 GHz machine. Figure 4 presents the block diagram of proposed model.

The training and testing algorithms of proposed method are given below:

Algorithm 1: Training phase:

- Step 1. Acquire the input training set images (*E. coli* electron microscope image set).
- Step 2. Transform the present input image into grayscale image.
- Step 3. Accomplish preprocessing by using morphological procedures, namely, erosion, reconstruction and dilation.
- Step 4. Achieve segmentation of the image of step 3 using proposed SIS and particle filtering with Chan–Vese level set method and obtain resulting binary image.
- Step 5. Identify and eliminate the border touched cells to obtain binary image of ROI and then perform labelling the segmented binary image.
- Step 6. Calculate geometric shape features for each labelled segment (Major axis, Minor Axis, Area, Eccentricity, Perimeter, Length/Width ratio, Compactness) and store them.
- Step 7. Iterate the steps 1–6 for all the training images.
- Step 8. Calculate the minimum feature vector and maximum feature vector of *E. coli* particle and store them as knowledge base.

Algorithm 2: Testing phase:

- Step 1. Acquire the input: *E. coli* electron microscope image set.
- Step 2. Transform the present input image into grayscale image.

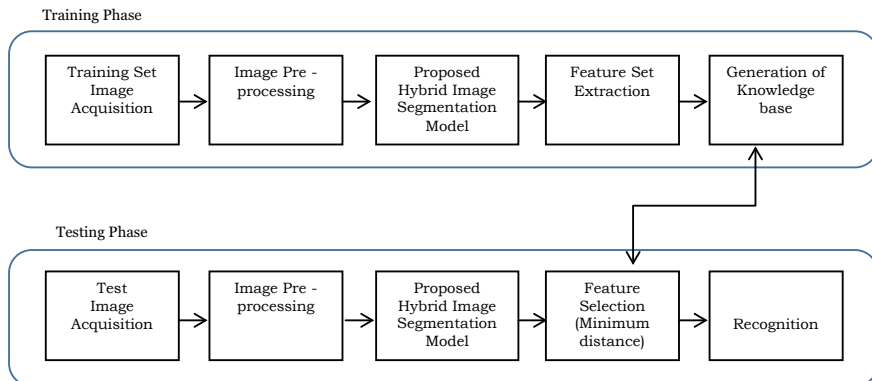


Fig. 4 Generic block diagram of the proposed method

- Step 3. Accomplish preprocessing using morphological procedures, namely, erosion, reconstruction and dilation.
- Step 4. Achieve segmentation of the image of step 3 using proposed SIS and particle filtering with Chan–Vese level set method and obtain resulting binary image.
- Step 5. Identify and eliminate the border touched cells to obtain binary image of ROI and then perform labelling the segmented binary image.
- Step 6. Calculate geometric shape features for each labelled segment (Major axis, Minor Axis, Area, Eccentricity, Perimeter, Length/Width ratio, Compactness) and store them.
- Step 7. Apply rule for classification of the *E. coli* particles: A segmented region is of *E. coli*, if its feature values lie in min-max range (3 Sigma classification rule).
- Step 8. Iterate the steps 7 and 8 for all labelled segments and output the identified *E. coli* particles.

Cross-validation and Resampling Methods:

For cross-validation and resampling model, the principal necessity for suggested method is to obtain a number of training data set and validation data set pairs from the acquired imageset X (remaining some part as the test set). If the dataset sample “ X ” is huge enough, then randomly divide it into K parts. After this, randomly subdivide every part into two and use one half for training and the other half for validation. So, given a dataset X , generate K training/validation set pairs, $\{T_i, V_i\}_{i=1}^K$ from this dataset. Figure 5 shows the sample training dataset (Figs. 6 and 7; Tables 1 and 2).

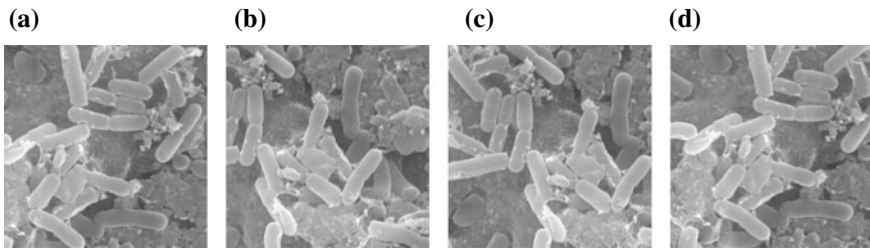


Fig. 5 Sample training images of *E. coli*

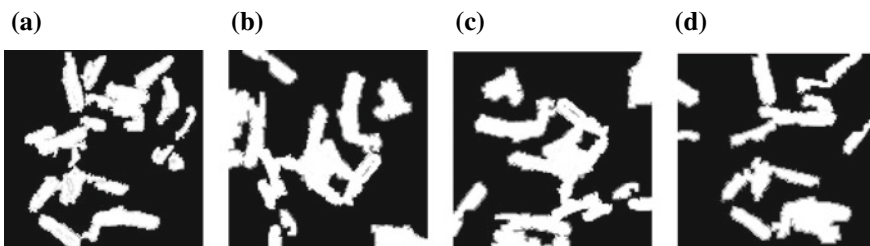


Fig. 6 Segmented outcome of region of interest

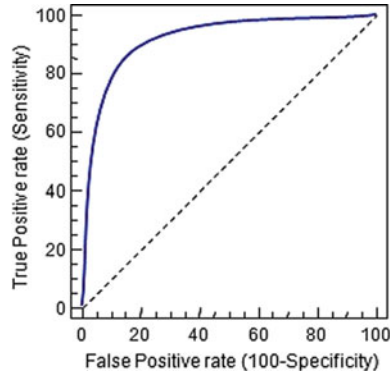


Fig. 7 Plot of Receiver Operating Characteristic (ROC) curve for the proposed model

Table 1 The feature vectors of the *E. coli* particles

<i>E. coli</i> features	Image pixel area	Region eccentricity	Particle perimeter	Particle circularity	Length/width ratio of the region
Particle 1	3165	0.52	367	0.41	1.17
Particle 2	3498	0.67	390	0.47	1.03
Particle 3	3434	0.70	326	0.50	1.09
Particle 4	3456	0.58	311	0.41	1.04
Particle 5	3684	0.59	375	0.39	1.04
Particle 6	3459	0.63	325	0.52	1.07
Particle 7	3698	0.68	300	0.38	1.07
Particle 8	3147	0.59	340	0.44	1.04
Particle 9	3021	0.66	339	0.49	1.17
Particle 10	3089	0.63	378	0.41	1.11

Table 2 The minimum and maximum geometric feature values of knowledgebase

Feature set	Image pixel area	Region eccentricity	Particle perimeter	Particle circularity	Length/width ratio of the region
Minimum	3021	0.52	300	0.38	1.03
Maximum	3698	0.7	390	0.52	1.17

5 Conclusion

Image processing and pattern recognition methods presently used in larger scale to aid medical experts. The current technologies such as machine learning also integrated to achieve better accurate results. In this research article, we have proposed a computer-aided *E. coli* particle segmentation and recognition based on improved method which comprises SIS, particle filter and Chan–Vese level set method. The investigational outcomes are matched with the manual outcomes gained by microbiological professionals. The projected model is further optimum and computationally less complex. It produces a classification rate of 95.50% for *E. coli* particles. The current research work can be enhanced further by improved preprocessing techniques, feature sets and classifiers, which will be taken up in our future work.

References

1. Caselles, V., Kimmel, R., & Sapiro, G. (1997). Geodesic Active Contours. *International Journal of Computer Vision*, 22(1), 61–79.
2. Mumford, D. (1989). Optimal approximation by piecewise smooth functions and associated variational problems. *Communications on Pure Applied Mathematics*, 42(5), 577–685.
3. Chan, T., & Vese, L. (2001). Active contours without edges. *IEEE Transactions on Image Processing*, 10(2), 266–277. <https://doi.org/10.1109/83.902291>.
4. Tsai, A., Yezzi, A., & Willsky, A. S. (2001). Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE Transactions on Image Processing*, 10(8), 1169–1186. <https://doi.org/10.1109/83.935033>.
5. Cremers, D., & Soatto, S. (2004). Motion competition: A variational approach to piecewise parametric motion segmentation. *International Journal of Computer Vision*, 62(3), 249–265.
6. Goldenberg, R., Kimmel, R., Rivlin, E., & Rudzsky, M. (2001). Fast geodesic active contours. *IEEE Transactions on Image Processing*, 10(10), 1467–1475. <https://doi.org/10.1109/83.951533>.
7. Niethammer, M., & Tannenbaum, A. (2004). Dynamic geodesic snakes for visual tracking. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004* (Vol. 1, pp. 660–667). <https://doi.org/10.1109/cvpr.2004.1315095>.
8. Paragios, N., & Deriche, R. (2002). Geodesic active regions: A new framework to deal with frame partition problems in computer vision. *Journal of Visual Communication and Image Representation*, 13(12), 249–268. <https://doi.org/10.1006/jvci.2001.0475>.
9. Avenel, C., Mémin, E., & Pérez, P. (2009). Tracking closed curves with non-linear stochastic filters. In X.-C. Tai, K. Mrken, M. Lysaker, & K.-A. Lie (Eds.), *Scale space and variational methods in computer vision*, no. 5567 in Lecture Notes in Computer Science (pp. 576–587). Berlin: Springer.
10. Lesage, D., Angelini, E. D., Bloch, I., & Funka-Lea, G. (2009). A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes. *Medical Image Analysis*, 13(6), 819–845. <https://doi.org/10.1016/j.media.2009.07.011>.
11. Image courtesy: from http://en.wikipedia.org/wiki/Image:EscherichiaColi_NIAID.jpg—*Escherichia coli*: Scanning electron micrograph of *Escherichia coli*, grown in culture and adhered to a cover slip. Credit: Rocky Mountain Laboratories, NIAID.
12. Blake, A., Curwen, R., & Zisserman, A. (1993). A framework for spatiotemporal control in the tracking of visual contours. *International Journal of Computer Vision*, 11(2), 127–145.

13. Doucet, A., Godsill, S., & Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3), 197–208.
14. Isard, M., & Blake, A. (1998). CONDENSATION—Conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1), 5–28.
15. Yilmaz, A., Li, X., & Shah, M. (2004). Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11), 1531–1536. <https://doi.org/10.1109/TPAMI.2004.96>.
16. Shao, J., Porikli, F., & Chellappa, R. (2007). Estimation of contour motion and deformation for nonrigid object tracking. *JOSA A*, 24(8), 2109–2121.
17. Pham, D. L., Xu, C., & Prince, J. L. (2000). A survey of current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, 2, 315–338.
18. Zaitoun, N. M., & Aqel, M. J. (2015). Survey on image segmentation techniques. *Procedia Computer Science*, 65, 797–806.
19. Iglesias, J. E., & Sabuncu, M. R. (2015, August). Multi-atlas segmentation of biomedical images: A survey. *Medical Image Analysis*, 24(1), 205–219.
20. Hamuda, E., Glavin, M., & Jones, E. (2016, July). A survey of image processing techniques for plant extraction and segmentation in the field. *Computers and Electronics in Agriculture*, 125, 184–199.
21. Khairuzzaman, A. K. M., & Chaudhury, S. (2017). Multilevel thresholding using grey wolf optimizer for image segmentation. *Expert Systems with Applications*, 86(15), 64–76.
22. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A. (2017, December). A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42, 60–88.
23. Mesejo, P., Ibáñez, Ó., Córdón, Ó., Cagnoni, S. (2016, July). A survey on image segmentation using metaheuristic-based deformable models: State of the art and critical analysis. *Applied Soft Computing*, 44, 1–29.
24. Sridevi, M., & Mala, C. (2012). A survey on monochrome image segmentation methods. *Procedia Technology*, 6, 548–555.
25. Comaniciu, D., Ramesh, V., Meer, P. (2000). Real-time tracking of non-rigid objects using mean shift. In *IEEE Conference on Computer Vision and Pattern Recognition, 2000. Proceedings* (Vol. 2, pp. 142–149). IEEE.
26. Sethian, J. A. (1999). *Level set methods and fast marching methods: Evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science* (Vol. 3). Cambridge University Press.
27. Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4), 321–331.
28. <https://www.cliffsnotes.com/study-guides/biology/microbiology/microscopy/staining-techniques>.

Automatic Bengali Document Categorization Based on Deep Convolution Nets



Md. Rajib Hossain and Mohammed Moshikul Hoque

Abstract Automatic document categorization has gained much attention by natural language processing (NLP) researches due to the enormous availability of text resources in digital form in recent years. It is the process of assigning a document into one or more categories that help the document manipulate and sort quickly. An efficient information processing system is required due to the rapid growth of Bengali text contents in digital form for searching, organizing, and retrieving tasks. In this paper, we proposed a framework for classifying Bengali text documents using deep convolution nets. The proposed framework consists of word embedding and document classifier models. Experiments with more than 1 million Bengali text documents reveals that the proposed system worthy of classifying documents with 94.96% accuracy.

Keywords Bengali language processing · Document categorization · Word embedding · Deep convolution neural nets

1 Introduction

Automatic document categorization is a challenging task in the field of NLP where a text document or a sequence of text documents automatically assigned into a set of predefined categories. Bengali language is spoken by about 245 million people in all over the world and is being considered the seventh most spoken language in the world [1]. Number of Bengali text documents in digital form have grown rapidly day by day in size and variety due to the increased usability of the Internet. It is very difficult to manage such huge amount of text documents for human expert manually

Md. R. Hossain · M. M. Hoque (✉)
Department of Computer Science & Engineering, Chittagong University of Engineering
& Technology (CUET), Chittagong 4349, Bangladesh
e-mail: moshiulh@yahoo.com

Md. R. Hossain
e-mail: rajcsecuet@gmail.com

© Springer Nature Singapore Pte Ltd. 2019
N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication
and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_43

that also consume a lot of time and cost of money. Therefore, an automatic document categorization system will be developed to handle a large amount of text data so that documents can be organized, manipulated, or sorted easily and quickly.

Although very few research activities have been conducted on Bangla language processing (BLP) such as syntax analysis, English to Bangla MT, Bangla OCR, and so on. Bengali text document categorization is also an important research issue that needs to be solved. There are many linguistic and statistical approaches that have been developed for automatic documents categorization of English and European languages. However, no usable and effective system is developed for classifying the Bangla texts still today. Bangla document categorization system may be used by security agency to identify the suspected streamed web data or spam detection, the daily newspapers to organize news by subject categories, the library to classify papers or books, the hospitals to categorize patient based on diagnosis reports, archiving, or clustering the government/nongovernment organization data, improve Bengali content searching, retrieving or mining specific web data, and so on.

In this paper, we proposed a framework for automatic Bengali documents categorization that works with word embedding model and deep convolution neural networks (DCNNs) [2, 3]. The proposed model will be able to overcome the traditional document classification shortcomings and also hardware cost. The word embedding algorithm such as Word2Vec extracts semantic feature for each word and represents as 1D vector. The semantic feature carries the actual meaning with respect to surrounding words and word order. Each document represents a 2D feature vector where the rows represent the word and columns represent the feature values. There are lots of hyperparameters in Word2Vec algorithm and we tune these parameters for Bengali corpus and achieved the better accuracy with respect to other language corpora. In the recent year, the convolution neural networks (CNNs) achieved the very good result for English and some other languages [4]. We design a DCNNs architecture where each hyperparameters will be well trained for Bengali large-scale data set and generate a classifier model. The model obtained the better accuracy for categorization of Bengali text documents.

2 Related Work

A significant amount of researches has been conducted on document categorization in English and European languages. In the recent year, the Word2Vec [5–7] and Glove [8] algorithms achieved the state-of-the-art word embedding result for English, French, Arabic, and Turkish languages. The CNN- and RNN-based documents classifier approaches achieved 84.00 and 85.60% accuracy from English text [9]. Conneau et al. [3] introduced very deep CNN and achieved 96–98% accuracy from different English data set classification. In hierarchical, CNN (HCNN)-based and decision-tree-based CNN also have been achieved higher accuracy for English text [10, 11]. Stochastic gradient descent (SGD) based classifier perform lower accuracy due to feature scaling and lack of huge hyperparameters tuning [12]. The character-level

CNN is performing slower embedding system and memory consuming [13]. There are very few researches have been conducted on Bengali text document classification. The TF-IDF-based features are the traditional technique which only depends on documents term frequency or BoW model [1, 12, 14]. Lexical feature only carries the limited number of information such as average sentence length, average number of words length, number of different words, and so on [15]. TF-IDF feature performs lower accuracy due to absence of semantic information. The TF-IDF feature not contained the word position and correlation of the word. The lexical and TF-IDF feature are not working properly for Bengali language due to its large inflectional diversity in verbs, tense, noun, etc. In our work, we use DCNNs for Bengali documents categorization. This approach showed better performance than the previous Bengali text classification methods due to the hyperparameter tuning and deep network training architecture.

3 Methodology

The proposed document categorization architecture consists of three main modules: text to feature extraction module, documents classifier training module, and documents classifier projection module.

3.1 Text to Feature Extraction Module

Word2Vec with skip-gram algorithm is used to train a Bengali word embedding model. In order to train the model, we tune the window size and embedding dimension. The tuned hyperparameter shows the better result for Bengali text classification purpose. The embedding model row numbers represent the number of distinct words and columns represent the feature dimension. First split the sentence to word list for each text document and for each word feature vector is extracted from embedding model. Therefore, for each document, a 2D feature vector is generated.

3.2 Documents Classifier Training Module

Figure 1 shows the overall architecture of Bengali text document classifier model. The DCNNs is consisting of input layer, convolution layer, rectified linear units (ReLU) layer, pooling layer, and fully connected (FC) layer. $S(R, C, F)$ represents the input and output tensor where R represents the number of rows, C for the number of columns, and F for the depth or the number of filters.

Input layer: The input document has sequentially passed each word through the look-up table ($W_{v,d}$) and generates a $n \times d$ representation vector, where n represents

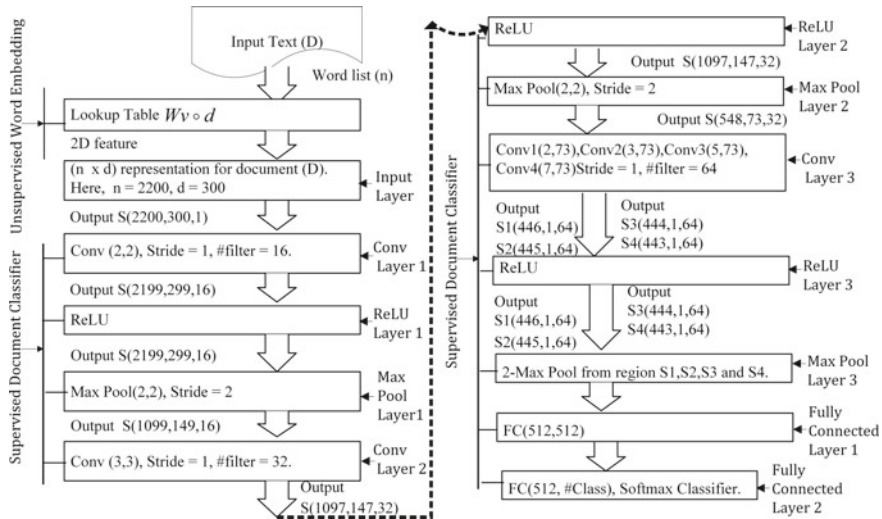


Fig. 1 DCNNs layered architecture

the number of word in document and d represents the feature dimension. In our case, n is fixed to 2200 and d to 300. If the number of word shorter than 2200 then a padding operation will be imposed. The 2D representation $S(2200, 300, 1)$ is fed to the input tensor that will be known by the input layer.

Convolution layer: The convolution layer extracts the local feature from the input tensor and trained the filter weights. The i th layer height and width are calculated by the following Eqs. (1)–(3).

$$D_i^l = D \quad (1)$$

$$H_i^l = H_{i-1}^{l-1} - D_h^l + 1 \quad (2)$$

$$W_i^l = W_{i-1}^{l-1} - D_w^l + 1 \quad (3)$$

Here, padding = 0 and stride = 1. H_i^l , W_i^l represents the output layer height and width, H_{i-1}^{l-1} , W_{i-1}^{l-1} represents the input layer height and width, respectively. The D denoted the number of filter in the current layer. D_h^l, D_w^l represents the filter height and width. Each of the convolution operation follow a Eq. (4).

$$S_{i^l j^l d^l}^l = \sum_{i=0}^{H_i^l} \sum_{j=0}^{W_i^l} \sum_{d^l=0}^{D_i^l} S_{i,j,d}^{l-1} \times K_{i,j,d}^l \quad (4)$$

where $K_{i,j,d}^l$ represents the kernel of each filter. In this architecture, convolution operation is imposed in three different layers. The first operation is applied at input tensor $S(2200, 300, 1)$ with 16 filters and kernel size (2, 2) with stride is to

one, and output tensor is to $S(2199, 299, 16)$. The second operation is applied with 32 filter and kernel size (3, 3) with stride is to one, and the output tensor is to $S(1097, 147, 32)$. The last operation is applied with four different kernel sizes such as Conv1(2, 73), Conv2(3, 73), Conv3(5, 73), and Conv4(7, 73) with zero padding and stride is to one only for heightwise and 64 filters. The output produces four tensors with $S1(446, 1, 64)$, $S2(445, 1, 64)$, $S3(444, 1, 64)$, and $S4(443, 1, 64)$.

Rectified Linear Units (ReLU) layer: The rectified linear units (ReLUs) are a special operation that combines nonlinearity and rectification layers in DCNNs. The input and output volumes remain same in this layer only changing the elementwise value.

$$S_i^l = \max(0, S_i^{l-1}) \quad (5)$$

Here, $\max(0, S_i^{l-1})$ denotes the elementwise operation. ReLU propagates the gradient efficiently, and therefore reduces the likelihood of a vanishing gradient problem which in turn reduces the time complexity. The ReLU sets negative values to zero, and therefore solved the cancellation problem.

Pooling layer: The pooling or feature reduction layer is responsible for reducing the volume of activation maps. They are used for reducing the computational requirements progressively through the network as well as minimizing the likelihood of overfitting. In the DCNNs architecture, the max pooling operation is imposed in three different layer. In the pooling layer, the input and output are calculated by Eqs. (6)–(7).

$$H^l = \frac{(H^{l-1} - K_h^l + 2P^l)}{T^l} + 1 \quad (6)$$

$$W^l = \frac{(W^{l-1} - K_w^l + 2P^l)}{T^l} + 1 \quad (7)$$

where H^l and W^l are the output tensor height and width of pooling layer. K_h^l , K_w^l , and T^l are the pooling layer pooling height, width, and stride size. In the DCNNs architecture, the padding parameter P is to zero and the first pooling layer input tensor $S(2199, 299, 16)$ with max pool(2, 2), stride is to two, and the output tensor size is to $S(1099, 149, 16)$. The second pooling layer input tensor is $S(1097, 147, 32)$ with max pool(2, 2), stride is to 2 and the output tensor size is $S(548, 73, 32)$. In the last pooling layer, the notation 2-max pool from each region means the pool 2 nonoverlapping feature from each feature map. The output tensor of the pooling layer having $2 * 4 * 64 = 512$ feature.

Fully connected (FC) layer: The fully connected or dense layer is a last layer of DCNNs. It follows the convolution, ReLU, and max pooling layer. The main goal of the layer is to design a flatten tensor which is input tensor activation map in three-dimensional volume. The transformation from input to output tensor of FC layer is $S(R, C, F) \rightarrow S(R * C * F, 1)$, where R , C , and F denote the input tensor rows, columns, and number of filters. In a FC layer, every node in the layer is connected to each other in the preceding layer. In Fig. 1, the first $FC(512, 512)$ means that the input tensor $S(512, 1)$ is connected to $S(512, 1)$. The second FC layer $F(512, \#Class)$

means that each of the class got a classification score using Softmax classifier. The value of the each node in FC layer is calculated by Eq. (8).

$$\theta_i^l = \sum_{j=1}^M W_{i,j}^l A_i^{l-1} \quad (8)$$

where $W_{i,j}^l$ represents the weight of current layer l and A_i^{l-1} represents the previous layer activation maps. The θ_i^l calculates the projected value.

$$A_i^l = F(\theta_i^l) \quad (9)$$

Here, A_i^l represents the current layer activation maps.

3.3 Documents Classifier Projection Module

In this module, unlabeled text data is taken as input and it is divided into word and if the input is less than 2200 words then padding is added. For each word is look-up by embedding model and generate a 2D feature vector where each row represents the word and corresponding column represents the feature vector. The DCNNs architecture trained a classifier model (pretrained model) which saves the layerwise different filter data. The DCNNs is initialized by the trained model and 2D feature is projected by the DCNNs architecture, and finally produces a score vector which means that by classwise expected score. The output of FC layer is projected by a weight matrix W with C distinct linear classification, and the predicted probability for the i th class given by Eq. (10).

$$P(\text{class} = i | X) = \frac{(e^{X^T W_i})}{\sum_{c=1}^C e^{X^T W_c}} \quad (10)$$

where X is a i th class feature vector and W is the trained weight matrix. For each unlabeled text data, the system provides 12 significant score, from this score we select the max value which is desired class value.

4 Experiments

We implemented DCNNs algorithm in Python with TensorFlow and ran the experiments on a Nvidia GeForce GTX 1070 GPU. This GPU has 8 GB of GPU RAM, which helps us for extending the networks and batch size. A GPU-based computer with 32 GB physical memory and Intel Core i7-7700K CPU with 256 GB SSD is

used for experiments. The DCNNs architecture has lots of hyperparameters. The proposed system is suited for better performance with the following parameters: number of batch size = 32, L2 regularization lambda = 0.0001, number of training epochs= 200, word embedding = Word2Vec, decay coefficient = 1.5, dropout keep probability = 0.50, save model after this many steps = 100, evaluate model on development set after this many steps = 100, and convolutions filter initialization method = Xavier initialization.

4.1 Corpus

Resource acquisition is one of the challenging hurdles to work with electronically low resource languages like Bengali. Due to the lack of available Bengali corpus for text categorization, we have built our own corpus and data in the corpus have extracted from the available online Bengali resources such as blogs and newspapers [16–19]. The unlabeled data is stored in the corpus for word embedding purposes. We have collected 836412 Bengali unique words. If a word is not found in the embedding table, then applying zero-value padding technique. Table 1 shows the summary of the dataset.

For the classification tasks, handcrafted labeled data are collected from Bengali online newspapers. Table 2 shows the statistics of labeled dataset. The number of training and testing documents are varied in each category with variable word length. The label of the training data is assigned by the human expert that help to achieve better accuracy. In the label corpus, if the number of words is lower than 2200 then zero padding is added to each of the input document.

4.2 Evaluation Measures

In order to evaluate the proposed Bengali document categorization system, we used the following measures: development/training phases loss versus iteration, develop-

Table 1 Embedding data summary

Number of documents	113082
Number of sentence	220000
Number of words	33407511
Number of unique words	836412
Embedding type	Word2Vec
Contextual window size	12
Word embedding dimension	300

Table 2 Summary of the categorical data

Category name	#Training documents	#Testing documents
Accident (AC)	6069	402
Art (AR)	1264	146
Crime (CR)	11,322	1312
Economics (EC)	4526	743
Education (ED)	5159	865
Entertainment (ET)	8558	1734
Environment (EV)	923	110
Health (HE)	2004	580
Opinion (OP)	6479	1248
Politics (PL)	24,136	1834
Science & Technology (ST)	3106	694
Sports (SP)	12,653	1039
Total	86,199	10,707

ment phase/training phases accuracy versus iteration. The testing phase statistical analysis is shown by precision, recall, F_1 -measure, and accuracy.

Training and development phase evaluation: The loss and accuracy of training and development imply the model beauty or better fitness. The loss and accuracy is the summation of the errors made for each example in training and development sets. In each training iteration, the loss is calculated by the following equation:

$$\text{Loss}_i = -W * X_i^T + \sum_{c=1}^C e^{W_c * X_i^T} \quad (11)$$

where the Loss_i means the i th iteration loss mean value and W represents the Softmax layer weight matrix, where rows and column represent the feature and category values, respectively. The X_i represents the i th example feature vector. The C represents the total category of our system. The training accuracy is calculated by Eq. (12).

$$\text{Acc}_i = \frac{P_i}{M_i} \quad (12)$$

Here, Acc_i , P_i , and M_i represent the i th iteration accuracy value, total number of correctly predicted category, and total number of sample data point in that iteration. The development loss and accuracy are also calculated using Eqs. (11) and (12).

Testing phase evaluation: In order to measure the overall performance of the proposed system, we used precision, recall, accuracy, and F_1 -measure [Eqs. (13)–(16)].

$$\text{Precision} = \frac{T_p}{T_p + F_p} \quad (13)$$

$$\mathbf{Recall} = \frac{T_p}{T_p + F_n} \quad (14)$$

$$\mathbf{Accuracy} = \frac{T_p + T_n}{T_p + F_p + T_n + F_n} \quad (15)$$

$$\mathbf{F_1\text{-measure}} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Recall} + \text{Precision}} \quad (16)$$

where T_p , T_n , F_p , and F_n represent the true positive, true negative, false positive, and false negative, respectively.

5 Results

In order to check the effect of size of training data on the performance, we plotted two learning curves: loss versus iterations and accuracy versus iterations. Moreover, we applied the ROC measure for the testing phase accuracy.

- **Development and training loss:** Figure 2 shows the impact of number of iterations on loss in development and training phases. In the development phase at iteration number 1 loss is more than 2.5 and loss is decreased with the increase of iteration. The decreasing rate of loss is stable at iteration number 20,000 (Fig. 2a). In that case, we trained on the training data only and test on the tested data. In the training phase, at iteration number 1 the loss is more than 35 and the loss is decreasing from the iteration number 5000 (Fig. 2b). This decreasing rate remains stable after

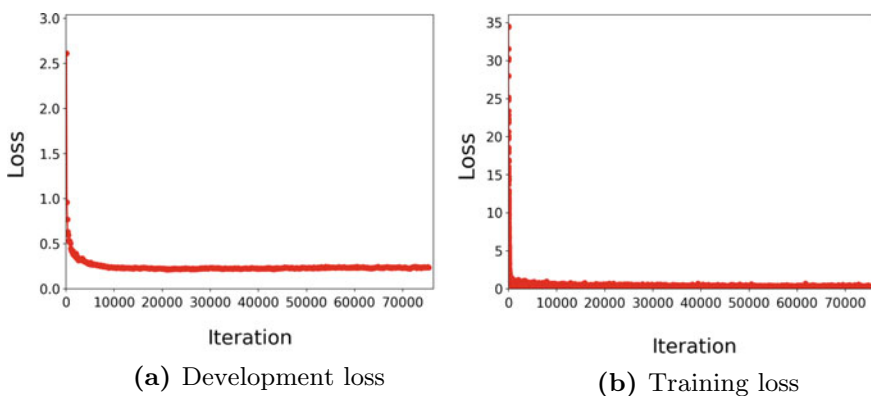


Fig. 2 Losses in development and training phases

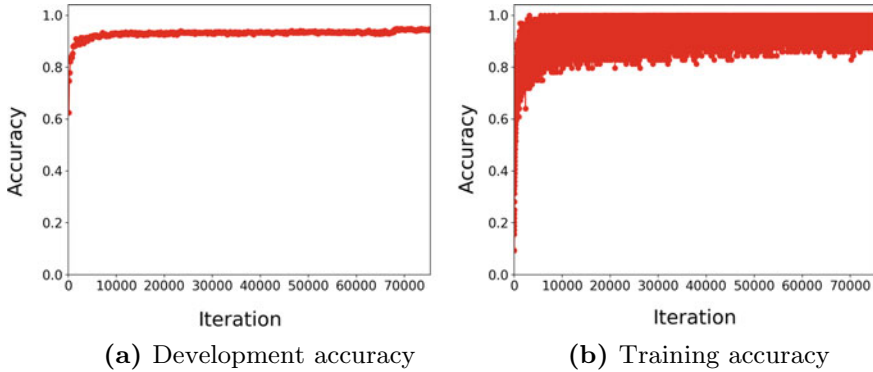


Fig. 3 Accuracy with iteration in development and training phases

10,000 iterations. In this case, we trained on a set consisting of both, training and development data, and test on the test data.

- **Impact of number of iteration on accuracy:** Figure 3 illustrates the impact of number of iterations on accuracy. Figure 3a shows that the development accuracy is increasing with respect to iteration number and varies from 87.00% (iteration no. 8000) to 93.00% (iteration no. 65000). However, the accuracy is almost stable from iteration number 68,000 and remains constant with accuracy about 94.50%. Figure 3b shows that the training accuracy is increasing with respect to the iteration numbers.

5.1 Testing Performance

Table 3 shows the precision, recall, and F_1 -measure of the proposed system. The maximum accuracy is achieved by the entertainment (ET) class and the minimum accuracy is achieved by the environment (EV) class.

5.2 Comparison with Existing Approaches

To evaluate the effectiveness, we compare the proposed system with existing approaches [1, 5]. Table 4 summarizes the comparison performance. This result shows that the proposed system outperforms the previous work in terms of accuracy.

- **ROC measures:** Figure 4 shows the ROC curve for multiclass Bengali documents categorization. This curve reveals that the maximum area under the curve covered by both classes SP and ST, whereas the minimum area covered by the EV class

Table 3 Summary of the analysis

Category name	Precision	Recall	F ₁ -score	Support
HE	0.90	0.88	0.89	580
AC	0.90	0.89	0.89	402
AR	0.81	0.83	0.82	146
CR	0.99	0.92	0.95	1312
EC	0.93	0.94	0.94	743
ED	0.98	0.93	0.95	865
ET	0.99	0.99	0.99	1734
EV	0.85	0.62	0.72	110
OP	0.96	0.96	0.96	1248
PL	0.93	0.98	0.96	1834
ST	0.93	0.98	0.95	694
SP	0.94	0.97	0.96	1039
Avg./total	0.95	0.95	0.95	10,707

Table 4 Performance comparison

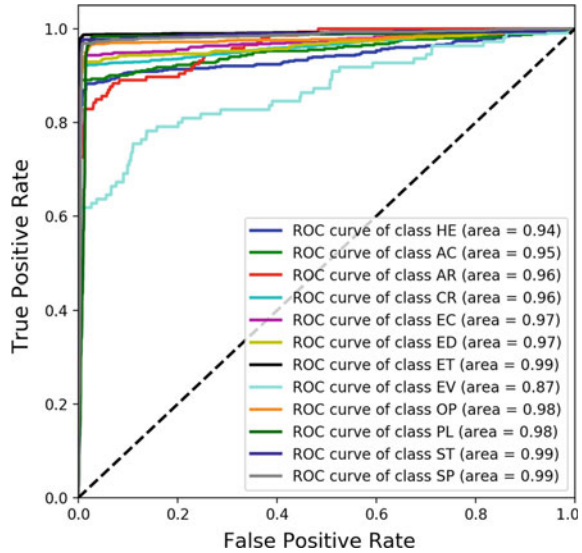
Method	#Training documents	#Testing documents	#Category	Accuracy (%)
TF-IDF + SVM [1]	1000	118	5	89.14
Word2Vec+K-NN+SVM [5]	19,750	4713	7	91.02
Proposed	86,199	10,707	12	94.96

(0.87). According to area under cover value, ten categories go to excellent class and only one class (EV) falls into good class and there is no category into bad class.

6 Conclusion

Text document classification is a well-studied problem for the highly resourced languages like English. However, it is a relatively new problem for an under-resourced language, especially in Bengali. Bengali text document categorization is a challenging task due to the lack amount of digitized text, and scarcity of available corpora. In this work, we introduce a new technique for Bengali document categorization system based on DCNNs. In this system, semantic features are extracted by Word2Vec algorithm and classifier is trained by DCNNs. We used 86,199 documents for training and 10,707 documents for testing and both sets have been handcrafted from online newspapers. The system obtains 94.96% accuracy for text document classification

Fig. 4 Receiver operating characteristic (ROC)



and shows the better performance compared to the existing techniques. For future research, we will extend our framework for other forms of text classification such as books, blogs, tweets, and online forum threads. Moreover, we will also include more classes with more data to improve the overall performance of the system.

References

1. Mandal, A. K., & Sen, R. (2014). Supervised learning methods for Bangla web document categorization. *International Journal of Artificial Intelligence and Applications (IJAIA)*, 5(5), 93–105.
2. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *Journal of CoRR*.
3. Conneau, A., Schwenk, H., & Cun, Y. L. (2017). Very deep convolutional networks for text classification. In *The 15th Conference of the European Chapter of the Association for Computational Linguistics* (Vol. 1, pp. 1107–1116), Valencia, Spain.
4. Xu, K., Feng, Y., Huang, S., & Zhao, D. (2015). Semantic relation classification via convolutional neural networks with simple negative sampling. In *Empirical Methods in Natural Language Processing* (pp. 536–540), Lisbon, Portugal.
5. Ahmad, A., & Amin, M. R. (2016). Bengali word embeddings and its application in solving document classification problem. In *19th International Conference on Computer and Information Technology* (pp. 425–430).
6. Johnson, R., & Zhang, T. (2017). Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)* (Vol. 1, pp. 562–570).
7. Tang, D., Qi, B., & Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. In *Empirical Methods in Natural Language Processing* (pp. 1422–1432), Lisbon, Portugal.

8. Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543).
9. Lee, J. Y., & Deroncourt, F. (2016). Sequential short-text classification with recurrent and convolutional neural networks. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)* (pp. 515–520).
10. Kowsari, K., Brown, D. E., Heidarysafa, M., Meimandi, K. J., & Barnes, L. E. (2017). Hierarchical deep learning for text classification. In *16th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 364–371).
11. Bahassine, S., Madani, A., & Kissi, M. (2017). Arabic text classification using new stemmer for feature selection and decision trees. *Journal of Engineering Science and Technology*, 12, 1475–1487.
12. Kabir, F., Siddique, S., Kotwal, M., & Huda, M. (2015, March). Bangla text document categorization using stochastic gradient descent (SGD) classifier. In 2015 International Conference on Cognitive Computing and Information Processing (CCIP) (pp. 1–4).
13. Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolution networks for text classification. In *NIPS'15 28th International Conference on Neural Information Processing Systems* (Vol. 1, pp. 649–657).
14. Krendzelak, M., & Jakab, F. (2015). Text categorization with machine learning and hierarchical structures. In *Proceedings of 13th International Conference on Emerging eLearning Technologies and Applications* (pp. 1–5).
15. Liebeskind, C., Kotlerman, L., & Dagan, I. (2015). Text Categorization from category name in an industry motivated scenario. *Journal of Language Resources and Evaluation*, 49(2), 227–261.
16. The Daily Prothom Alo, Online, <http://www.prothom-alo.com>.
17. The Daily Jugantor, Online, <https://www.jugantor.com>.
18. The Daily Ittefaq, Online, <http://www.ittefaq.com.bd>.
19. The Daily Manobkantha, Online, <http://www.manobkantha.com>.

Artist Recommendation System Using Hybrid Method: A Novel Approach



Ajay Dhruv, Aastha Kamath, Anuja Powar and Karan Gaikwad

Abstract Recommendation systems have a wide range of applications today in the digital world. The recommender system must be able to accurately predict the users' tastes as well as broaden their horizon about the available products. There are various dimensions in which recommendation systems are created and evaluated. Accuracy and diversity play an important role in the recommendation systems and a trade-off must be identified between the two parameters to suit the business requirements. The proposed system makes use of various recommendation approaches to give a wide range of recommendations to users. The recommendations are provided based on similarity of the selected artist, top artists in a genre, using a hybrid model and artists listened by users' friends. Some recommendations would include the most popular ones, and some would be randomly picked for a diverse range of recommendations.

Keywords Collaborative filtering · Diversity · Hybrid model · R · Recommender systems

1 Introduction

In today's world of Internet and large-scale growth of e-commerce business, recommender systems play an important role in the growth of the business. This is because if the website provides products as per the users' requirements, the customer is bound to be loyal to that company for a long period of time. Hence, effective recommendations play an important role [1, 2]. The main objective of a recommender system from the customer point of view is to provide suggestions to online users to make

A. Dhruv (✉) · A. Kamath · A. Powar · K. Gaikwad
Department of Information Technology, Vidyalkar Institute of Technology,
University of Mumbai, Mumbai 400037, India
e-mail: ajay.dhruv@vit.edu.in

A. Kamath
e-mail: aasthaskamath@gmail.com

A. Powar
e-mail: anujapowar@hotmail.com

© Springer Nature Singapore Pte Ltd. 2019
N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_44

better decisions from a variety of options available. The main goal from the business point is to increase the sales and profit [3]. Recommending popular items involves lower risk in terms of accuracy. However, it is not always advisable to recommend only the popular ones. Increasing the diversity of recommendations is essential in businesses. Increasing the diversity and accuracy is not in the same direction, thus making it important to make ideal decisions between the two parameters [4]. Appropriate recommendation model must be chosen so as to maximize the benefits and maintain customer loyalty by creating value-added relationships. The user contentment plays an important parameter in measuring the success of the recommendations provided [5]. A recommender system must consider diverse factors while providing recommendations. Some of them include vendor's target market, data sources, data availability, scalability, algorithms to be used, and interfaces to manage the recommendations and data access.

Recommender systems are broadly classified as content-based filtering, collaborative filtering (CF), hybrid approach, and knowledge-based recommender systems.

1.1 Content-Based Filtering

The content-based filtering method (Fig. 1) makes use of content and features of the item already purchased by the user to recommend items. Content-based filtering considers how do the items ia and ib relate to each other [6]. The similarity between two items can be calculated using various methods like Pearson, Jaccard similarity, and Cosine similarity. The problem with content-based filtering is overspecialization. Overspecialization is taking into account only those items that are very similar to each other and giving least importance to the interests of the users. Also, it offers only partial information, generally text information, and the visual and semantic information is a challenging task [7].

1.2 Collaborative Filtering

The collaborative filtering (CF) (Fig. 2) approach makes recommendations for a user with the help of collective preferences of other users [3, 6]. Collaborative filtering

Fig. 1 Content-based filtering

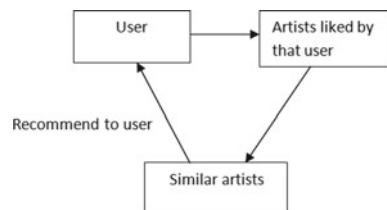
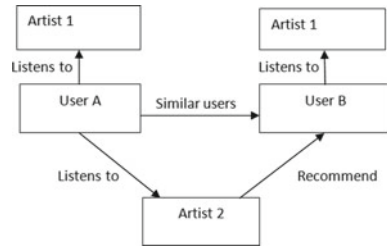


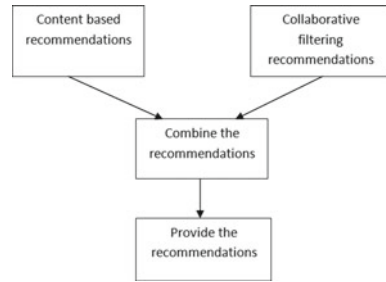
Fig. 2 Collaborative filtering



models are grouped into two types—neighborhood-based and model-based methods. Neighborhood-based methods include user-based collaborative filtering (UBCF) and item-based collaborative filtering (IBCF). The UBCF recommends to a user those items that are most preferred by similar users [1]. The IBCF recommends to a user the items that are most similar to the users’ purchases. Model-based approaches include Bayesian clustering, adaptive learning, linear classification, and neural networks [8]. The user feedback can either be explicit indications or implicit indications. Implicit feedback gathering does not require active user requirement. Feedback can be derived from monitoring users’ activities and click-throughs. Explicit feedback can be gathered by letting users rate the items. Knowing whether user like/disliked a particular item and text comments can be analyzed. CF, however, suffers from data sparsity problems, particularly when dealing with large datasets. The cold start problem occurs when a new user or item has just entered the system. Unless some information becomes available from reliable sources, it is difficult to find similar ones. New items cannot be recommended until some users’ rate it. To alleviate the data sparsity problem, many approaches have been proposed. Some of them are dimensionality reduction techniques, such as singular value decomposition (SVD), removing unrepresentative or insignificant users (users who have not given much ratings/reviews), or items to reduce the dimensionalities of the user–item matrix directly [9].

1.3 Hybrid

The hybrid system (Fig. 3) uses information from previous user–item interactions and contents of purchased items. There are seven types of hybrid methods—Weighted, switching, mixed, feature combination, cascade, feature augmentation, and meta-level [10].

Fig. 3 Hybrid model

1.4 Knowledge-Based

There might arise some conditions when collaborative and content-based filtering does not work. The knowledge-based filtering uses explicit knowledge about the users and the products [11]. Such systems require knowledge about the group of items and knowledge about the users [7].

2 Investigating Related Work

Table 1 shows the comparative study of the existing recommender systems.

3 Proposed System

The proposed system aims to implement a multi-featured approach to musical artist recommendation through the use of UBCF algorithm, similarity matrices, content-based filtering, and hybrid filtering. The proposed system will address the diversity and accuracy problems of recommender system which was missing in the earlier systems. Accuracy and diversity are not in the same direction; hence, the proposed system will implement a trade-off between these two quantities [4]. The flowchart (Fig. 4) of the proposed system is as shown.

4 Implementation

The system was implemented using the R language. The “Last.fm” dataset was used for implementation.

4.1 Data Acquisition and Cleaning

The artist dataset has been taken for research. The dataset has information about the artists, users, friends of users, and artists tagged by the users along with its timestamp. Additional attributes like age and frequency of user listening to a particular artist were added to improve the quality of the recommendations. The data cleaning is the process of filling missing values, smoothing the data, removing any outliers, and resolving any inconsistencies [15]. Thus, the dataset was cleaned and made suitable for analysis.

4.2 Loading the Data

The data was loaded to RStudio to build a recommender model.

Table 1 Comparison of recommender systems

S. No.	Ref. No.	Title of paper	Technology used	Advantages
1	[6]	Building a sporting goods recommendation system	Single value decomposition (SVD), with matrix factorization, ALS-WR algorithm	<ol style="list-style-type: none"> 1. Predicts next purchase of user 2. Considers the purchase frequency of items 3. Uses item-based recommendation to solve cold start problem
2	[10]	Recommender system for news articles using supervised learning	SVD algorithm	<ol style="list-style-type: none"> 1. The system considered various parameters like relevancy, readability, novelty, and time-on-page to recommend articles 2. Category of articles is considered to improve recommendations

(continued)

Table 1 (continued)

S. No.	Ref. No.	Title of paper	Technology used	Advantages
3	[12]	Research paper recommendation with topic analysis	Collaborative filtering algorithm	<ol style="list-style-type: none"> 1. Generate satisfactory recommendations with few ratings 2. Alleviates cold start problem
4	[7]	Development of a tourism recommender system	Artificial intelligence mechanism	<ol style="list-style-type: none"> 1. The system plans and combines itineraries with other cultural and leisure activities according to user's preferences 2. Considers factors like place of origin of user, travel group, date of trip, and activities to recommend places of interest
5	[8]	A personalized electronic movie recommendation system based on support vector machine and improved particle swarm optimization	Support vector machine (SVM) classification, improved particle swarm optimization (IPSO)	<ol style="list-style-type: none"> 1. Personalized recommendation 2. Evolution speed factor and aggregation degree factor used to optimize parameters of the model
6	[13]	A new recommender system for the interactive radio network Fmhost	Collaborative filtering algorithm	<ol style="list-style-type: none"> 1. Matrix factorization tech to increase scalability
7	[14]	An innovative tour recommendation system for tourists in Japan	Collaborative filtering, greedy algorithm	<ol style="list-style-type: none"> 1. Suggests optimal touring plans composed of various points of interest

4.3 Developing Recommender Engine

The RStudio aids development of recommender systems. It supports various algorithms like UBCF, IBCF, SVD, POPULAR, RANDOM, and Hybrid recommendations [16]. The dataset was tested using various methods on similarity. Analysis using different models was also performed.

Fig. 4 Flowchart of the proposed system

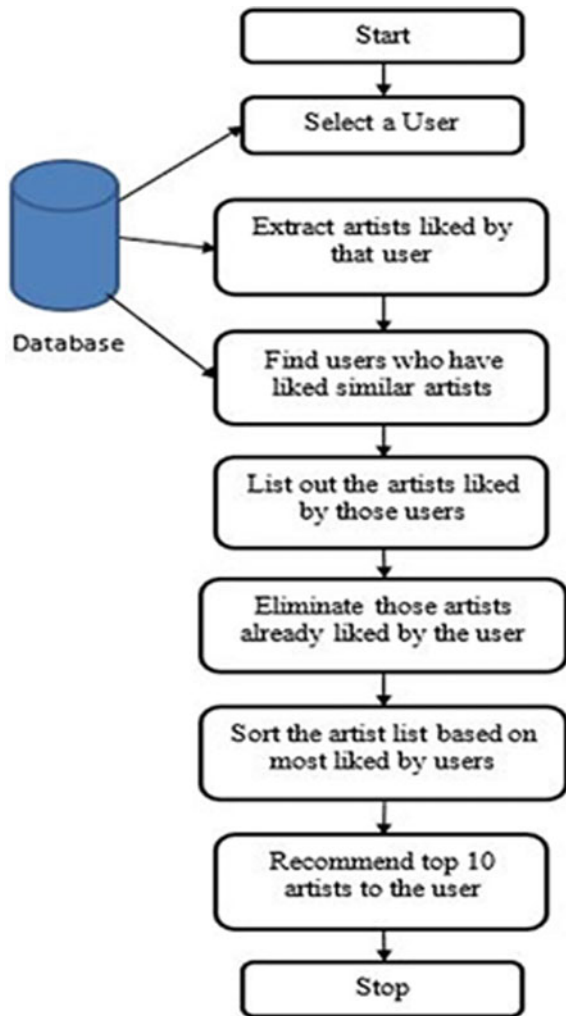


Figure 5 shows the comparison of ROC curves using three models—UBCF, RANDOM, and POPULAR. ROC analysis helps to pick the best model. The greater the area under the ROC curve, the better is the model performance. As seen in the graph, the UBCF outperforms the other models. Though UBCF performs better, due weightage has been given to RANDOM and POPULAR methods to include a diverse range of recommendations.

The recommender model provides recommendations by implementing various algorithms. Many factors are considered for recommendation. A user can get top 10 similar artists by selecting a particular artist (Fig. 6). The similarity was calculated using similarity() function, and cosine distance was used for measuring similarity (Fig. 7). The greater the value of the cosine distance, the greater the similarity between

Fig. 5 Comparison of various models

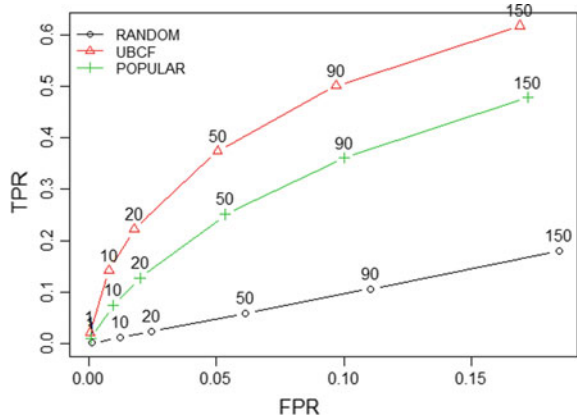


Fig. 6 Similar artists to a selected artist

Top 10 Artists Similar to Selected Artist

Select Artist:

Madonna

Christina Aguilera

Beyonce

Pink

Black Eyed Peas

Eminem

Prince

Shakira

Destiny's Child

Queen

two artists. The rows and columns represent the unique artists, and the cell value corresponds to the degree of similarity. The characteristics of the artist are considered to recommend to the user [17].

The user can get the top 5 recommendations on a particular genre (Fig. 8). The user tagged data is used to classify an artist in a particular genre. The rows represent artist IDs, and the columns represent the tag ID (Fig. 9). The cell value shows the frequency of artist tagged to a particular genre. As the data is sparse in the matrix, the matrix is binarized to give recommendations.

The user-based collaborative filtering (UBCF) model (Fig. 10) provides recommendations by finding users that are similar to the selected user. From such users, the top 20 artists are determined by the frequency of tagging of the artists. The artists that are previously listened by the artist are removed. From the remaining artists, the top seven are selected for recommendation. The remaining three are selected from

Fig. 7 Artist similarity matrix

▲	7	9	12	15	25	30	32
77	0.200	0.070	0.000	0.000	0.000	0.277	0.206
81	0.247	0.115	0.000	0.000	0.058	0.325	0.169
84	0.131	0.275	0.000	0.000	0.000	0.207	0.269
85	0.057	0.000	0.000	0.000	0.120	0.337	0.263
86	0.082	0.000	0.000	0.000	0.087	0.049	0.064
88	0.230	0.207	0.000	0.082	0.069	0.195	0.152
89	0.299	0.157	0.000	0.062	0.052	0.237	0.231
93	0.149	0.314	0.000	0.000	0.000	0.355	0.385
96	0.144	0.000	0.000	0.000	0.101	0.057	0.074

Fig. 8 Top artists in a genre

Top 5 Artists in Selected Genre

Select Genre:

Duran Duran

Madonna

Depeche Mode

Michael Jackson

The Cure

the remaining list of artists. These ten recommendations are shuffled and presented to the user.

The hybrid model uses the linear-weighted method of recommendation. A linear-weighted hybrid is composed of recommendation components $\kappa 1$ through κk , whose output is combined by computing a weighted sum. A linear-weighted hybrid of this style has a number of advantages like; the recommendations are specialized in a particular dimension of the data. Thus, the linear-weighted hybrid offers a way to construct algorithms that take all dimensions of a system into account without requiring mathematically complex and computationally intensive dimensionality reduction techniques, which are less extensible and flexible [18].

In the hybrid model (Fig. 11), the weights are assigned to POPULAR, RANDOM, and RERECOMMEND methods to provide recommendations. The POPULAR method recommends the trending artists. The RANDOM method selects a random list of artists. It is used to improve diversity of the recommender system. The RERECOMMEND method recommends artists from the user’s history. The weights

Fig. 9 Artist–genre matrix

▲	1	2	4	5	6	7	13
7	25	3	NA	NA	15	1	NA
9	NA	NA	NA	NA	NA	NA	NA
12	NA	NA	4	7	NA	NA	NA
15	1	NA	12	2	NA	NA	NA
25	7	NA	7	3	NA	7	NA
30	1	NA	NA	NA	NA	NA	NA
32	NA	NA	1	NA	NA	NA	NA
45	1	NA	NA	NA	NA	NA	NA
51	NA	NA	NA	NA	NA	NA	1
52	NA	NA	NA	NA	NA	NA	10
53	NA	NA	NA	NA	NA	NA	16
54	NA	NA	NA	NA	NA	NA	8
55	NA	NA	NA	NA	NA	NA	1
56	NA	NA	NA	NA	NA	NA	3

Select a User ID (2 - 2100)

7

Select A Recommendation Method:

- By Similar Artists (Top 5)
- By Genre (Top 5)
- 10 Artists Recommended by Similar Users
- Hybrid Recommender
- 10 Artists your Friends liked

Artist You Have Previously Listened To:

- Alanis Morissette
- Alicia Keys
- Ashanti
- Ashlee Simpson
- Ashley Tisdale
- Avril Lavigne

10 Artists You May Like- using UBCF

- Duran Duran
- Off the Sky
- The Killers
- Nine Inch Nails
- Elo da Corrente
- Lokua Kanza
- Maria Bethania
- Faith Evans
- Queensberry
- Movels Coloniais de Acaju

Fig. 10 Artist recommendation using UBCF

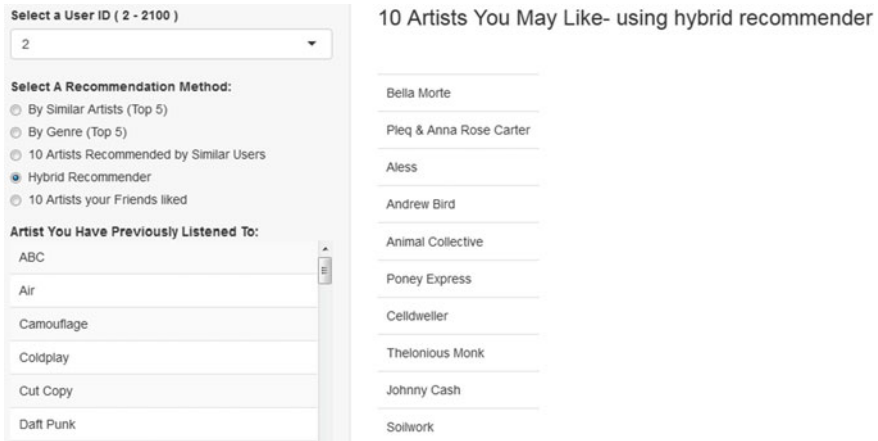


Fig. 11 Artists recommendation using hybrid model

assigned to POPULAR, RANDOM, and RERECOMMEND are 0.3, 0.6, and 0.1, respectively. Recommendations are also given considering artists tagged by friends of the users (Fig. 12). The friends are the ones with whom the user has confidence in and have similar likings as that of the user [19]. The friends are filtered considering their age. The artists that are tagged most frequently by those friends and not yet tagged by the user are recommended.

4.4 Model Evaluation

The most popular measure of effectiveness is accuracy metrics, which is used to evaluate the ratings a particular user would give to a particular item [3]. The true positives (TP) are the recommendations provided to the user and liked by the user. False positives (FP) are the recommendations provided but not liked by the user. The False Negatives (FN) are those which the system fails to recommend, though the user would like them. The true negatives (TN) are those which the user does not like and are not recommended [19].

The Top-N recommendation lists are typically evaluated in terms of their precision and recall. Precision measures the percentage of recommended products that are relevant, whereas recall measures the percentage of relevant products that are recommended [18, 20].

The accuracy of the recommender system is calculated as follows [8]:

$$\text{Accuracy} = \frac{\text{The correctly classified samples}}{\text{The total number of samples}} \tag{1}$$

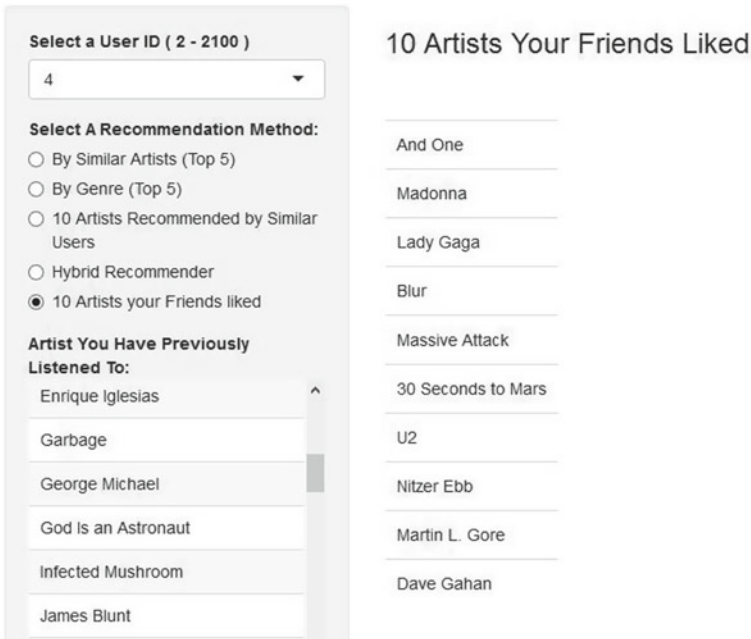


Fig. 12 Artist recommendation from friends’ tags

The performance of the UBCF model is shown in Table 2 and that of the hybrid model is shown in Table 3. From Formula (1), the accuracy of the system is calculated. The accuracy of the system using the UBCF method is 95%, and using the hybrid model is 69%.

5 Conclusion

In this paper, a recommender system is implemented using various recommendation approaches. The system also gives recommendations based on artist similarity, top artists in a genre, and artists listened by users’ friends. An interactive user interface is developed to display the recommendations. It is observed that the accuracy of the system decreases on increasing the randomness of the recommendations. An attempt has been made to provide a list of accurate and diverse range of recommendations to the user, using various algorithms.

Table 2 Performance of the system using the UBCF method

TP	FP	FN	TN	Precision	Recall	TPR	FPR
1.29144385	18.70855615	15.55080214	778.44919786	0.06457219	0.08461227	0.08461227	0.02345738

Table 3 Performance of the system using the hybrid method

TP	FP	FN	TN	Precision	Recall	TPR	FPR
1.1149733	8.8850267	24,9919786	770.0080214	0.1114973	0.0428570	0.0428570	0.0113972

6 Future Work

Possible future work with the recommender model could include assessing in an online environment whether or not the suggested “Artists You Might Enjoy” lists lead users to explore artists they have not listened to previously. Such an assessment could be done by tracking click-through rates for those lists. The system could include URLs of the artists which, upon clicking on the artist recommended, will directly lead the users on their respective websites. The system could be made dynamic by including user feedback for the recommendations provided and replacing those recommendations which the user did not like with new ones. The recommendations liked by the user could be used for recommending further items.

References

1. Bhumichitr, K., Channarukul, S., Saejiem, N., Jiamthapthaksin, R., & Nongpong, K. (2017, July). *Recommender Systems for university elective course recommendation*. Presented at the 14th International Joint Conference on Computer Science and Software Engineering (JCSSE), Nakhon Si Thammarat, Thailand. <https://doi.org/10.1109/jcsse.2017.8025933>.
2. Ekvall, N. (2012). *Movie recommendation system based on clustered low-rank approximation*. Master's Thesis, Department of Mathematics, Linköping University, Sweden.
3. Shakirova, E. (2017). *Collaborative filtering for music recommender system*. Presented at the IEEE Conference of Russian, Young Researchers in Electrical and Electronic Engineering (EIConRus), St. Petersburg, Russia. <https://doi.org/10.1109/elconrus.2017.7910613>.
4. Javari, A., & Jalili, M. (2014). A probabilistic model to resolve diversity–accuracy challenge of recommendation systems. In: *Knowledge and Information Systems*, London (pp. 609–627). <https://doi.org/10.1007/s10115-014-0779-2>.
5. Ge, M., & Persia, F. (2017). *Research challenges in multimedia recommender systems*. Presented at the IEEE 11th International Conference on Semantic Computing, San Diego, CA, USA. <https://doi.org/10.1109/icsc.2017.31>.
6. Flodman, M. (2015). *Building a sporting goods recommendation system*. Degree project in Computer Science, Second Level KTH Royal Institute of Technology 26.
7. Ciurana Simó, E. R. (2012). *Development of a Tourism recommender system*. Master of Science Thesis, Master in Artificial Intelligence, (UPC-URV-UB).
8. Wang, X., Luo, F., Qian, Y., & Ranzi, G. (2016). A personalized electronic movie recommendation system based on support vector machine and improved particle swarm optimization. *PLoS ONE*, 11, e0165868. <https://doi.org/10.1371/journal.pone.0165868>.
9. Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009, 1–19. <https://doi.org/10.1155/2009/421425>.
10. Chaturvedi, A. K. (2017). *Recommender system for news articles using supervised learning*. MISS Master Thesis, Department of Information and Communication Technologies—UPF, Barcelona, Spain.
11. Gorakala, S. K., & Usuelli, M. (2015). *Building a recommendation system with R: Learn the art of building robust and powerful recommendation engines using R*. UK: Packt Publishing.
12. Pan, C., & Li, W. (2010, June). *Research paper recommendation with topic analysis*. Presented at the International Conference on Computer Design and Applications, Beijing, China. <https://doi.org/10.1109/icdda.2010.5541170>.
13. Zaharchuk, V., Ignatov, D. I., & Konstantinov, A. (2012). *A new recommender system for the interactive radio network FMhost*. National Research University Higher School of Economic 12.

14. Le, T. Q., & Pishva, D. (2016, January). *An innovative tour recommendation system for tourists in Japan*. Presented at the 18th International Conference on Advanced Communication Technology (ICACT), Pyeongchang, South Korea. <https://doi.org/10.1109/icact.2015.7224843>.
15. Jiang, D., & Shang, W. (2017). Design and implementation of recommendation system of micro video's topic. In IEEEACIS 16th International Conference on Computer and Information Science ICIS 3. <https://doi.org/10.1109/icis.2017.7960040>.
16. Hahsler, M. (2015). Recommenderlab: A framework for developing and testing recommendation algorithms. CRAN, 40.
17. Dou, Y., Yang, H., & Deng, X. (2016, August). *A survey of collaborative filtering algorithms for social recommender systems*. Presented at the 12th International Conference on Semantics, Knowledge and Grids, Beijing, China. <https://doi.org/10.1109/skg.2016.014>.
18. Gemmell, J., Schimoler, T., Mobasher, B., & Burke, R. (2012). Resource recommendation in social annotation systems: A linear-weighted hybrid approach. *Journal of Computer and System Sciences*, 78, 1160–1174. <https://doi.org/10.1016/j.jcss.2011.10.006>.
19. Ramesh, B., & Reeba, R. (2017, April). *Secure recommendation system for E-commerce website*. Presented at the International Conference on circuits Power and Computing Technologies [ICCPCT], Kollam, India. <https://doi.org/10.1109/iccpct.2017.8074240>.
20. Paraschakis, D., Nilsson, B. J., & Hollander, J. (2015, December). *Comparative evaluation of Top-N recommenders in e-Commerce: An industrial perspective*. Presented at the IEEE 14th International Conference on Machine Learning and Applications, Miami, FL, USA. <https://doi.org/10.1109/icmla.2015.183>.

Anomaly Detection of DDOS Attacks Using Hadoop



Y. S. Kalai vani and P. Ranjana

Abstract A distributed denial of service is one of the major threats in the cyber network, and it causes the computers to have been flooded with the HTTP GET packet. As Http flood attacks used standard URL requests, it is quite challenging to differentiate from valid traffic. In an Http flood, the Http clients such as web browser interact with the application or server to send Http requests. The request can be either GET or POST. The aim of attack is when to compel the server to allocate as many resources as possible to serving the attacks, thus denying legitimate user's access to the server's resource. To handle this DDOS attack, the traditional intrusion detection system is not suitable to hold and find the huge amount of data in the network. Hadoop is a framework that allows processing and storing huge datasets. In Hadoop MapReduce is the programming model to process huge data stored in Hadoop. This paper explains how to detect the HTTP flood attack using KDD CUP 99' dataset with improvised the algorithm in MapReduce using anomaly detection strategy.

Keywords Hadoop · HTTP GET · MapReduce

1 Introduction

The cyberattack denial-of-service is a major threat in the Internet, and it causes many serious problems in the network. Distributed denial-of-service (DDOS) attacks come under the category of denial-of-service attacks. A DDOS is a cyberattack which is caused by executor using more than one IP address and flooding the request to the victim to attack the regular response. An HTTP GET attack is an application layer attack which consists of low and slow attacks, the target Apache, and Windows servers. This type of attack sends a large number of request which consists of unused

Y. S. Kalai vani (✉)

Department of Computer Applications, Sindhi College, Bangalore University, Bangalore 560024, India

e-mail: kalaiys@rediffmail.com

P. Ranjana

Department of CSE, Hindustan University, Padur, Chennai 600002, India

© Springer Nature Singapore Pte Ltd. 2019

N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,

https://doi.org/10.1007/978-981-13-5953-8_45

packets to the web server to slow down the server which are in an active state and it also shuts the server by exhausting the bandwidth limits of network.

Network intrusion detection system is a software tool which is used to detect any malicious behavior in the network. It consists of two-part signature-based detection which is used to detect the malicious information based on the signature, that is, known attacks are detected in the network. Anomaly-based intrusion detection is used to detect the anomalies in the network which is not known before, that is, usual behavior in the network. The HTTP flooding attack is detected by the anomaly intrusion detection system using the Hadoop environment. Hadoop is a platform which is used to store huge amount of data in the form of zeta bytes, so this Hadoop-based IDS is used to hold and detect huge volume of data in the network. The dataset used here is KDD CUP 99' [1] which is used to find the HTTP GET flooding rate using counter-based algorithm in MapReduce, using the parameter values which consists of some important classes such as normal, ICMP, TCP, UDP, Port scan HTTP, and IP. There are number of parameters used to check the anomalies in network such as protocol type, source bytes, destination bytes, HTTP, session rate, and counter IP. Proposed system follows the improvised counter-based algorithm which analyzes the packets and filter the packets based on the request types. If the request HTTP GET is more than the threshold, then that attack is filtered and emitted. Detection of anomalies in the network uses the preprocessing technique in the counter-based algorithm.

Hadoop provides the platform to achieve the efficient way of detecting HTTP GET flooding [2] attacks using the MapReduce scheme. It is used to calculate the number of packets per second in the network; if it exceeds the threshold, the network traffic flow is more. This impact causes decrease in speed of the computer. Counter-based algorithm is used to measure the parameters in the log file. If the protocol is HTTP, then attack is confirmed to HTTP flooding which exceeds the threshold and then the log files ignore anomalous attack.

2 Related Work

As the vulnerabilities are increased in the network in the form of cyberattacks, use powerful IDS to detect the anomalies in the network. Many research papers are given the information about the cyberattacks and detection methods. This research paper focuses on DDOS attacks and its different types. A DDOS is an attack derived from DOS. DOS attack is a malicious attempt by a single user or group of user to cause the victim, site, or server to deny service to its customers. A distributed denial-of-service is an attack in which multiple computer systems attack a target such as web server or network. Due to this attack sometimes, the web server may crash, not accepting any request from the user side. To handle DDOS attack, the traditional IDS is not suitable to hold the number of packets and to detect the attack. The network anomaly intrusion detection system in the Hadoop framework is used a tool to detect the HTTP GET flooding attack.

2.1 Categories of DDOS Attacks

Volume-based attack

It aims to saturate the bandwidth of the affected website, and its magnitude is measured in bits per second. It consists of user datagram packet (UDP) flood, Internet message protocol (ICMP), and spoofing of IP address. Volume-based attacks are measured by bits per second.

Protocol Attacks

Protocol is a set of constraints used to transmit the data in the form of packets. It comprises the attacks such as SYN FLOOD, fragmented packets, ping of death, Smurf DDOS, etc. SYN flood attack creates the attack by sending the SYN request which contains malicious code to attack the server. Protocols are measured in packets per second.

Application attacks

Application attacks are low and slow attacks which target the servers like Apache, Window, or OpenBSD vulnerabilities. This attack sends the request to server through HTTP GET/POST methods, which contains malicious code that creates the vulnerabilities in the server.

2.2 Methods to Detect the DDoS Attacks

This paper focuses on the application layer attack called HTTP GET flooding. This HTTP flooding attack can be detected by several approaches are used in intrusion detection scheme. Many algorithms are used to detect the intrusions in the network-based; on this category among all this, research paper focuses on MapReduce algorithm which is used to solve the problem of existing system drawbacks.

- Statistical method methods [1] are used to identify the detection by using entropy and Chi-square test by comparing the values of packet traffic coming from the clients.
- Support vector machine is a method which consists of a set of efficient data mining algorithm that comprises the attributes taken part in the network anomalous attack.
- Soft computing method is used to find the DDOS attack by specifying the neural network strategies to detect the HTTP Flood attacks.
- Hadoop-based method is a framework which is used to detect the HTTP GET [3] flood attack using the MapReduce algorithms.

2.3 HTTP GET Flood Attack

An HTTP GET flood attack is an application layer attack which damages the system information in the application level. This attack sent along with HTTP GET request which contains malicious code to damage the server. It is low and slow, but it creates a lot of damage in the TCP connection between client and user.

2.3.1 Network Attack Layer

HTTP flood attack is an application layer attack which sends the request from client to server after the TCP connection establishment. Attackers implementing their HTTP flood attack on web servers and its resources so the users unable to access the resources and server also. The following diagram explains about a single HTTP GET request attack, and how it slows down the server as it threshold is less than the traffic Rate.

Figure 1b explains how the flooding HTTP GET requests in a TCP connection that is multiple HTTP GET requests are coming from the client side for one single TCP connection. As the load is increased and it contains malicious info in the request, server cannot handle the request in specific format so it is unable to process the user request.

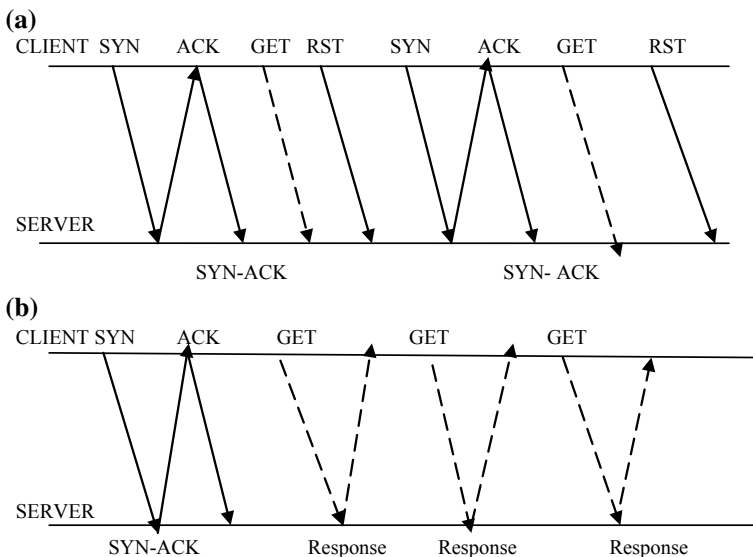


Fig. 1 a A single HTTP GET request attack in TCP connection. b A multiple HTTP GET request attack in TCP connection

2.4 *Intrusion Detection Techniques*

An intrusion detection is a software [4] application which monitors a network for malicious activity. Any type of malicious activities is collected and sent to the administrator; security issues are taken place to fix an alarm when the abnormal activities are occurred in the network.

Intrusion detection is classified into three types.

Host-Based Intrusion Detection System:

It is an intrusion detection system which monitors the internal part of the computer system and monitors the network packets in its network Interfaces.

Hybrid Intrusion Detection System:

Network-based intrusion detection system is a combination of host-based intrusion detection system and network-based intrusion detection system which monitors the computer's internal activities and network activities.

Network-Based Intrusion Detection System:

A network intrusion detection system is an intrusion detection system that attempts to discover unauthorized access to a computer network and detect the cyberthreats in the network.

A network intrusion is classified into two types:

1. Misuse detection (Signature-based) and
2. anomaly detection (profile-based).

Signature-based detection: This type of mechanism requires signature-based files, i.e., known patterns of attacks. If a pattern is matched, there is a possibility of an attack and then an alarm will be triggered. It has low false-positive alarm rate as matches are based on known patterns. Signature detection [1] fails to detect attacks which are variations in the known attacks.

Anomaly-based detection: This type analyzes computer and network activities and looks for an anomaly; if an anomaly is found, alarm is triggered. Anomaly is abnormal behavior of network or deviation from common rule for attacks. It has high false-positive alarm rate. The rise in use of technology has led to increase in amount of network traffic data. The traffic data is expected to be in the range of Zettabytes [2] and is significantly increased from past few years. This data having high Volume, Velocity, and Variety is often termed as Big Data. In this generation of Big Data, the IDS should be good enough to process huge volume of data in real time.

Unsupervised Anomaly detection Approach:

It is a type of anomaly detection approach which is the most flexible setup which does not require any label; moreover, there is no distinction between a training and test dataset. This idea behind an unsupervised anomaly detection algorithm scores the

data exclusively based on intrinsic properties of the dataset. Distances and densities are used to give estimation which is normal and what is an outlier or anomalies.

This research paper proposed a model of anomaly detection system using unsupervised data in KDD cup dataset. We have the test dataset which are not having class labels which are the example for the unsupervised dataset.



3 Proposed System Design and Implementation

3.1 Hadoop Frame Work

Hadoop is a framework which is used to process and store huge amount of data. KDD CUP99' is taken as dataset, and it is preprocessed by the Hadoop framework to avoid the duplicate records. Here, we have taken the dataset for DDOS attack which consists of a set of classes with its attributes. Anomaly-based intrusion detection technique is implemented to detect the abnormal activities. HTTP flood attack is caused due to many requests sent by the same user or from the different IP addresses which creates a traffic in the network and slows the process of the server. To handle the huge amount of IP address in the network, the Hadoop framework is taken as a proposed framework.

Hadoop is a framework which allows to store huge volume of datasets and to process the dataset. Hadoop framework is divided into two types such as processing and storage. MapReduce is a programming model which is used to process huge data stored in Hadoop. When the used install Hadoop in a cluster, the MapReduce will be provided as a service. In MapReduce, the user can write the programs to perform some computations in parallel and distributed fashion. MapReduce [3] is a programming framework and it takes the dataset as a huge volume in a distributed environment.

MapReduce is a programming framework that allows us to perform distributed and parallel processing on large datasets in a distributed environment. In this proposed system, we have taken KDD CUP 99' dataset for DDOS with specific classes and a set of attributes. MapReduce consists of two tasks, namely, Map and Reduce. First, the DDOS dataset is taken for sample and it is processed by the mapper. The output of the map work is key-value pairs of the dataset which does not contain the redundant data. The output of map job is the input of the reducer. The reducer aggregates the values, and applying the strategy gives the result as in the form of intermediate format.

Software configuration for this process is Apache Hadoop, and it is open-source software utilities that facilitate using a network of many computers to solve the problems of enormous amount of data and its computation. The core part of the Apache Hadoop [4] consists of a storage part which is called as Hadoop Distributed File System (HDFS), and processing model is considered as MapReduce.

Hadoop Modules:

The Apache Hadoop is a framework which consists of modules, namely, Hadoop Distributed File System and Hadoop YARN [5] which is responsible for managing computing resources in clusters and using them for scheduling user's applications; Hadoop MapReduce consists of the programming model for large-scale data processing. Hadoop framework is written in Java Programming Language and some code is written in C language.

3.2 *MapReduce*

MapReduce is a programming model and an associated implementation for processing and generating big datasets with a parallel, distributed algorithm on a cluster.

A MapReduce program is composed of map procedure which performs filtering and sorting [5], reduce method which performs a summary operation. MapReduce is composed of a map procedure which performs filtering and sorting so the detection algorithm which is used to separate the malicious activity from normal activity.

3.3 *Intrusion Detection in MapReduce*

HTTP GET flooding is considered as one of the serious HTTP flooding attacks. The attackers send the massive HTTP GET request to the victim server until the server and its resources are get exhausted. Several techniques are introduced to solve the problem such as threshold determination, pattern analysis, traffic analyzer, etc. HTTP flood attack is an application layer attack that causes damages in web applications and web servers. The requests are in the form of HTTP GET or POST method flooded [6, 7] to the target web server which is not handled by the web server. Requests coming from different IP addresses which causes the serious problem so the efficient algorithm is used to detect the intrusion in the network. The intrusions are in the form of anomalous behaviors and it should be classified into supervised and unsupervised data format.

To handle unsupervised data, the classification analysis is used and to handle supervised data the clustering algorithm is used (Fig. 2).

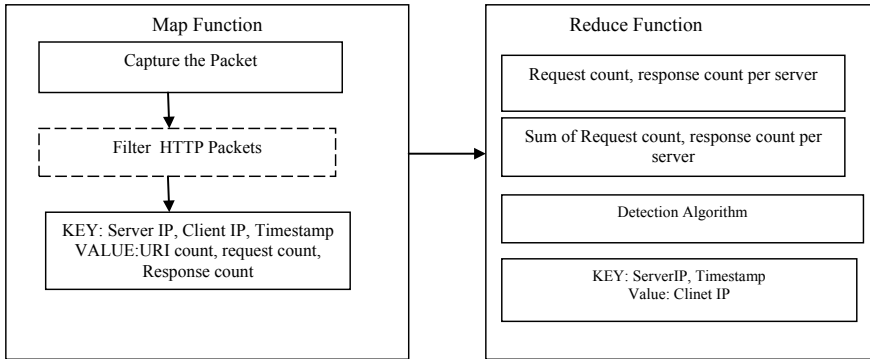


Fig. 2 MapReduce for HTTP GET flood attack detection

4 Proposed System and Design for DDOS

In this research work, the anomaly detection system which consists of KDD CUP 99’ used as a dataset. Anomaly detection is used to find any abnormal activities in the process. Hadoop is a platform which takes a data in a huge format and stores the data in the MapReduce [8]. It is open-source software which is used to detect any anomalies in the network. If servers are getting affected with the overloaded request, so server is unable to process other requests in the network (Fig. 3).

Algorithm 1: Packet Analysis Algorithm using Map Function

- Step 1: Read the dataset in KDD CUP 99’.
- Step 2: Preprocess the dataset for further processing.
- Step 3: Filter the selected feature in the KDD CUP 99’.
- Step 4: Take the test dataset.
- Step 5: Filter the HTTP GET dataset.
- Step 6: Tabulate the result including server IP and client IP address as key and value.

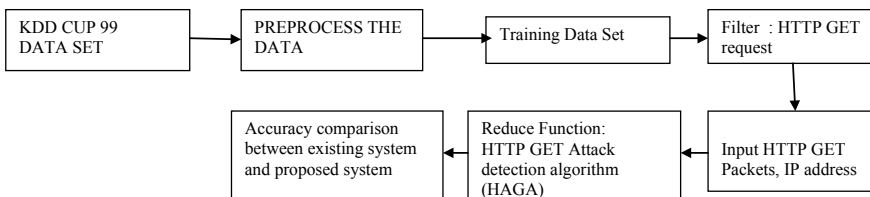


Fig. 3 Processing steps in proposed system

Algorithm 2: HTTP GET Attack Detection Algorithm (HAGA) using Reduce function

Input: c: KDD CUP 99' data set, <key,value> as
 < server IP, packet count >,
 < client IP, request Count >,
 < masked of timestamp, response count >

Output: emits the attacker's IP Address

Step 1: procedure Packet Analyzer using Map Reduce class

Step 2: Generate <key, valye> till the end of the file

Step 3: Aggregate <key, value> as
 <serverIP, packetcount>
 <clinetIP,requestcount>
 <maskedTimestamp, responsecount >

Step 4: calculate Detection Accuracy = $TP+FN/TP+FN+FP+TN$

Step 5 : If total request > threshold
 Detects the attacker's Address
 Else
 Continue.
 End if

Result:

In this research paper, the improvised algorithm for DDOS attack consists of a KDD CUP 99' dataset and attack is generated to server. Different types of attacks such as ICMP flooding, TCP flooding, UDP flooding, Smurf flooding, port scan, land flooding, HTTP flooding, session flooding, and IP flooding are generated and log files are created for every 30 min. To detect attack, attributes are derived from the log files and from the derived attributes dataset is created with nine different types of attack classes such as ICMP, TCP, UDP, and SMURF. Improvised counter-based algorithm for DDOS attack consists of dataset, and its attribute based on the HTTP GET attack detection is a simple method that counts the total traffic volume or the number of web page requests [5]. The algorithm is used to detect the false alarm rate in the network attack with the low false alarm rate. This algorithm needs three inputs parameters: time interval, threshold, and unbalance rate. The above algorithm is calculated based on the threshold value in the network. This algorithm calculates the false alarm rate base on the threshold rate and request, response rate. The execution time for the proposed algorithmic approach is around eighty seconds (80 s) which is comparatively less than the total time of the attack scenario in which the experiment was performed. So it can be concluded that this algorithmic approach is a near real-time one (Table 1).

Table 1 Table of detection accuracy

	False alarm		Detection data	
	Normal data	Detection data	Normal data	Detection data
KDD CUP 99'	0.06	0.02	0.01	0.933

5 Conclusion

This paper focuses on the detection of intrusion detection in network in the form of HTTP GET FLOOD attack which is detected by the Hadoop because the traditional IDS is not suitable for the detection of huge amount of data, so the Hadoop is a framework which processes the huge amount of data. To handle the detection in network, the tool SNORT is used to find the network traffic in the network and it is used to identify the flooding attack for the server and it detects the traffic in the network. So the improved snort is used to detect the HTTP FLOOD attack in the network in an efficient manner, and it follows the counter-based algorithm in MapReduce which is used to process the huge amount of data and is capable of detecting the attacks in the network.

References

1. Kaskar, J., Bhatt, R., & Shirsath, R. (2014). A system for detection of distributed denial of service (DDoS) attacks using KDD cup data set. *International Journal of Computer Science and Information Technologies (IJCSIT)*, 5(3), 2014, 3551–3555. Department of Information Technology, Dr. D. Y. Patil College of Engineering, Pune, India.
2. Department of CSE, TCE Madurai, India, centurykarthik@gmail.com, ananddrox@gmail.com, kannathalk@gmail.com 978-1-4673-0671-3/11/\$26.00©2011 IEEE.
3. Navale, G. S., Kasbekar, V., Ganjapatil, V., & Bugade, S. (2014). *Detecting and analyzing DDos attack using Map Reduce in Hadoop*. Sinhgad Institute of Technology and Science, Narhe, Pune 41 University of Pune, India.
4. Zhiguo Shi, J. A. (2015). *An intrusion detection system based on Hadoop*. IEEE.
5. Mazhar Rathore, M., & Paul, A. (2016). *Hadoop based real-time intrusion detection for high-speed networks*. IEEE.
6. Basappa Kodada, S. P. (2015). *Big-data analytics based security architecture to detect intrusions*. NJCIET.
7. Jangla, G. K., & Amne, D. A. (2015). Development of an intrusion detection system on big-data for detecting unknown attacks. *IJARCCCE*, 4.
8. Shaik Akbar, T. R. (2016). A hybrid Scheme based on big-data analytics using intrusion detection system. *IJST*.

Pedestrian Detection and Tracking: A Driver Assistance System



Shruti Maralappanavar, Nalini C. Iyer and Meena Maralappanavar

Abstract Detection and tracking of pedestrian is a challenging task due to variable appearances, wide range of poses, and irregular motion of pedestrian along with motion of tracking camera under complex outdoor environmental conditions. In this paper, we propose an algorithm for pedestrian detection and tracking using HOG descriptors and particle filtering technique. A robust algorithm for pedestrian detection is proposed which works under nonlinear motion and overcomes occlusions. The performance of the above algorithm is tested for outdoor environment using standard dataset. The particle filter has benefits of handling nonlinear motion and occlusions, and they concentrate consecutively on the higher density regions of the state space and it is simple to realize which provides a robust tracking environment. Performance comparison of particle with conventional Kalman is also presented for the above-said cases.

Keywords Histogram of oriented gradients (HOG) · Support vector machine (SVM) · Particle filter · Kalman filter

1 Introduction

Human beings have an amazing potential of detecting and tracking objects in the surrounding even if they are in motion or in stationary. This knowledge of human visual system allows them to prevent colliding with other pedestrians in crowded places. From a human beings point of view, it is a simple task to detect pedestrian from the surrounding objects such as trees, vehicles, etc. but from the machine vision outlook it is not an easy task as it involves dealing with many parameters

S. Maralappanavar · N. C. Iyer (✉) · M. Maralappanavar
KLE Technological University, Hubballi, India
e-mail: nalinic@bvb.edu

S. Maralappanavar
e-mail: mshruti32@gmail.com

M. Maralappanavar
e-mail: msm@bvb.edu

© Springer Nature Singapore Pte Ltd. 2019
N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_46

such as illumination, cluttered backgrounds, pose of pedestrian clothing, size, and shape. Thus, it is challenging for a machine vision system to implement pedestrian detection and tracking functionality owing to their variable appearance: Variation shape, size, color of clothes of the tracked pedestrian with accessories, umbrella, child in trolley and outdoor environment: environmental conditions like rain, fog, or complex and cluttered backgrounds with probable occlusion from surrounding objects such as trees, vehicles, buildings, poles, other pedestrians, etc.

2 Related Work

The important attributes for both detection and tracking problem include robustness, accuracy, and speed. Detection algorithm can be patch based or part based, and the tracking accuracy depends on how good is the detection algorithm. Object detection algorithm based on the formation of patches from image proposed by Prioletti et al. [1] is cumbersome because of the formation of more number of patches from image which determines the classification decision. Further, custom database needs to be manually interpreted making implementation difficult task.

Later, Aghajanian et al. [2] proposed pedestrian detection algorithm to overcome the creation of huge database, by modeling humans as flexible assemblies of parts. These parts are represented by co-occurrences of local features which captures the spatial layout of the part's appearance. But the disadvantage with this method was false detection and inaccuracy because of considering unwanted parts similar to humans as human detection. Accuracy level is quite average.

To improve the performance of detection with respect to earlier methods [1, 2], Dalal and Triggs [3] proposed localizing using Histograms of Oriented Gradients (HOG) method.

On the other hand, for tracking, prediction of the most probable object in the current frame based on inputs from the previous frame is given by Huang et al. [4] using Kalman filter which is restricted only to neighboring area of the location and fails for occlusion and nonlinear motion of the object. Later, pedestrian tracking algorithm to overcome occlusion was proposed by Owczarek et al. [5] using a particle filtering approach, where pedestrian is tracked based on the state information of particle.

In this paper, we discuss pedestrian detection using location coordinates and its changes in consecutive image frames for tracking by deriving HOG features for the detection [3] and performing particle filtering for tracking [5].

The algorithm proposed is tested for its performance with occlusion and nonlinear motion of pedestrian and is also validated with standard database.

The rest of the paper is as follows: Methodology in Sect. 3.1 which deals with pedestrian detection using Histogram of Oriented Gradients (HOG) features and training using Support Vector Machines (SVM) and particle filter details in Sect. 3.2. Realization and experimental results are discussed in Sect. 4 with conclusion in Sect. 5.

3 Methodology

Details of pedestrian detection and tracking algorithm are given in this section. To track an object in a video sequence, detecting of a pedestrian using HOG features [3] is performed and location information is obtained which is then followed by tracking using particle filtering [5]. Details of detection and tracking of a particle are briefed in part A and part B, respectively.

3.1 Pedestrian Detection

Pedestrian detection using HOG features extraction involves the operation such as cell formation, filtering, and block normalization as shown in Fig. 1 [3]. The details of operations are given below.

Extraction of HOG features

Cell formation, filtering, and obtaining 1D histogram:

- Dividing original image into cells of each 8×8 pixel.
- Applying Sobel operator, obtain magnitude and direction of gradient the internals of which clearly indicates the spreading of intensity gradients or edge directions.
- Creating 1D histogram of unsigned gradients obtained above, by distributing among nine bins which indicates edge orientations over the pixels of the cell. To overcome lighting and illumination effects, block normalization is done as follows.

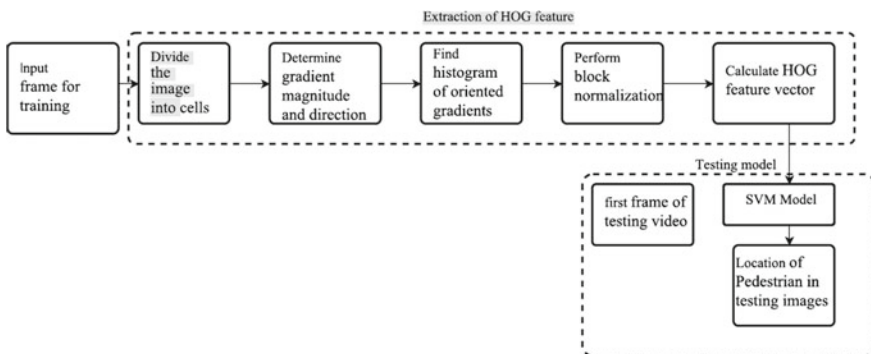


Fig. 1 Block diagram of pedestrian detection system

Block normalization

- Divide original image into block of each 16×16 pixel.
- Block is normalized with respect to histogram of the cells, leading to features to be extracted for an original image.

Testing Model

The extracted HOG features are used to obtain the model as follows:

- Original image as a test input to the trained model is given.
- Model gives the input about the presence or absence of pedestrian along with location.
- The location information is obtained from the detection algorithm which is used as an input to the tracking system as an initial state of input.

3.2 Particle Filter

Particle filtering is also named as the sequential Monte Carlo (SMC) [5], a simulation-based technique. Particle filter involves the operations such as initialization, particle prediction, assignment of weight to the particles, and resampling for an input frame of a given video after identifying and obtaining the location of the pedestrian.

In the particle filtering approach, distribution of the system state is approximated by a set of so-called particles and every particle is represented by a vector.

Algorithm can be summarized as follows [5]:

Initialization: Create a set of N -particles. Each particle has a state vector and an initial weight which corresponds to the target pedestrian's representation which is given as

$$S = \left[x, y, \frac{dx}{dt}, \frac{dy}{dt} \right]^T \quad (1)$$

$$\pi_0^{(n)} = \frac{1}{N} \quad (2)$$

where $n = 1, 2, \dots, N$,

where (x, y) are the coordinates of the bounding box, N is the number of particles, and dx/dt and dy/dt stands for the velocity in x -axis and y -axis.

Particle Prediction: The transition equations help in predicting new particle state. The state of each sample in every new frame is defined as function of driving vector, noise introduced to the state due to the measurement error of w_{t-1}

$$s_t^{(n)} = A s_{t-1}^{(n)} + w_{t-1}^{(n)} \quad (3)$$

where A is transition matrix which is defined by

$$A = \begin{bmatrix} 1 & 0 & dt & 0 & 0 \\ 0 & 1 & 0 & dt & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Assignment of weight to the particles: Each measurement z_t updates the weights of the particles.

$$\pi_t^{(n)} = \pi_{t-1}^{(n)} \cdot p\left(\frac{z_t}{s_t^{(n)}}\right) \quad (4)$$

$$\pi_t^{(n)} = \pi_{t-1}^{(n)} \cdot \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(d_t^{(n)})^2}{2\sigma^2}\right) \quad (5)$$

where $d_t^{(n)}$ represents the Hellinger distance between the hypothetical location and tracked pedestrian.

Perform weight normalization and sum up the weights to one.

Resampling: Ignore the particles which have negligible weights and again assign equal weights to the particles which have larger weights.

The distribution of particles after every operation is listed in Fig. 2. In the first frame of the video, particles are spread throughout the frame. The initial position of the pedestrian is obtained from the pedestrian detection system as shown in Fig. 2a.

Then, the particles are weighted based on the measurement update equation given in Eqs. (3), (4), (5), and thus the particles near the pedestrian will obtain higher weights based on conditional probability density. Discard the particles with negligible weights as they are of no use in the tracking process as shown in Fig. 2b. Resample the particles and assign equal weights to those particles which had higher weights in the previous iteration as depicted in Fig. 2c.

Once the pedestrian starts moving, again the weights of the particles are updated and particles with negligible weights are discarded. Further, resampling of particles is done. This process is done continuously until the pedestrian is tracked continuously.

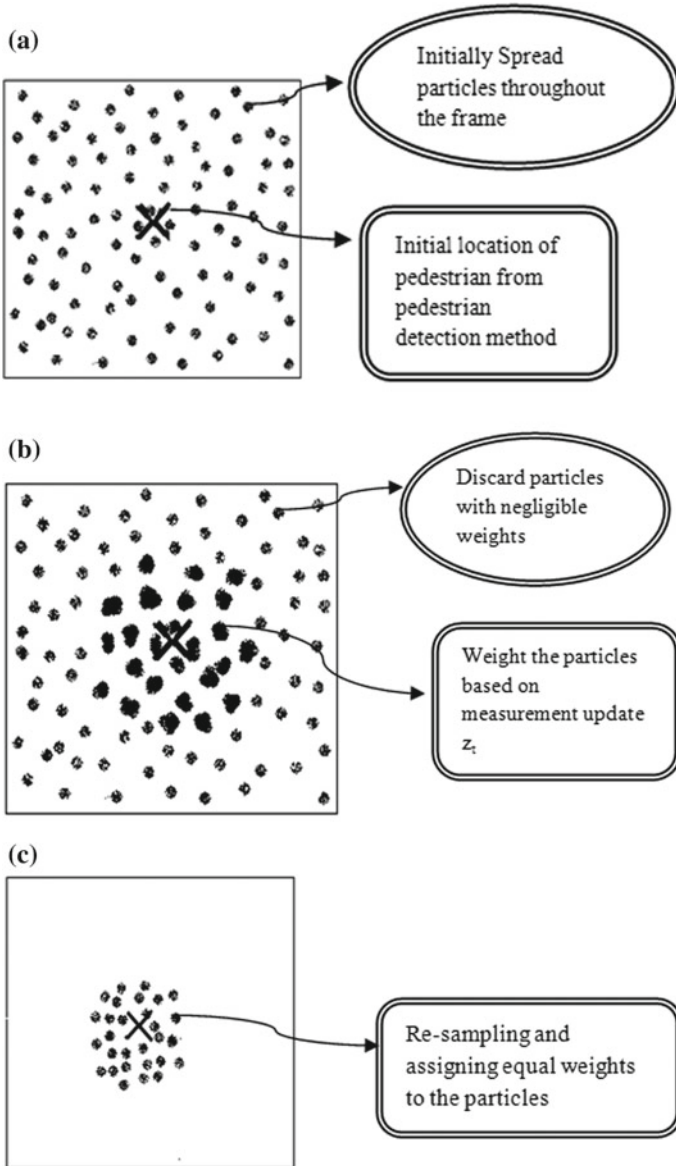


Fig. 2 Steps in particle filter algorithm

4 Realization and Experimental Results

The proposed algorithm of tracking of pedestrian using particle filtering is implemented in MATLAB and tested using CALTECH database. Two test videos having occlusion and nonlinear motion of pedestrian are taken as input. This is fed as input to Kalman and particle filter, and the results of this are shown in Figs. 3 and 4. The track speed is 50 frames/s by estimation. The results obtained for the two test conditions are explained below.

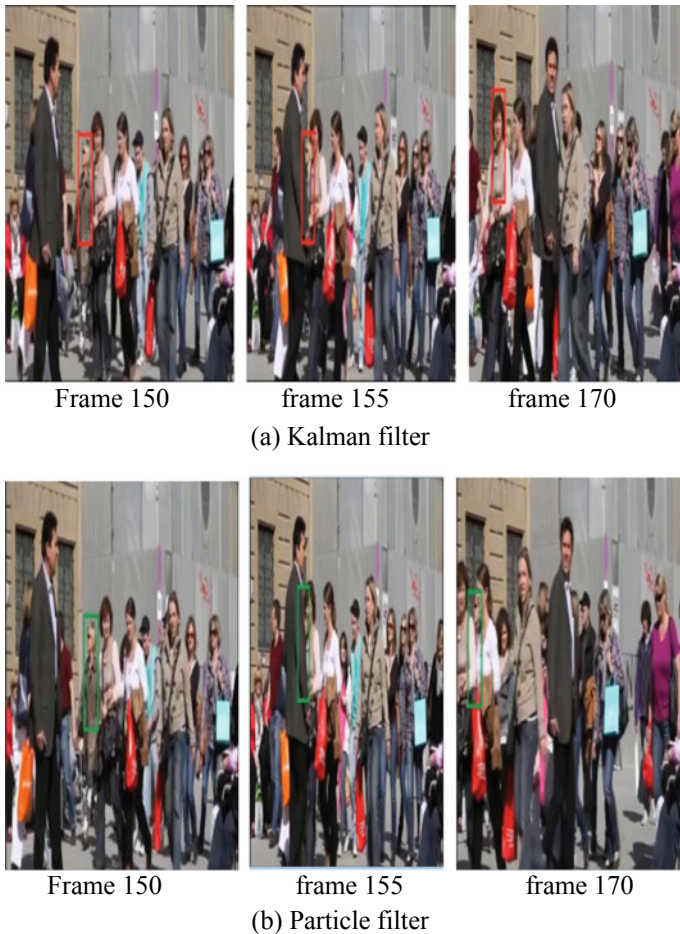


Fig. 3 Comparison of Kalman and particle filter

Case 1: Occlusion

For the first test video, with Kalman filter, tracking failed in frame 170 due to occlusion by the other pedestrian, whereas in the particle filter tracking the pedestrian who was tracked in frame 150 is still being tracked in frame 170 successfully as shown in Fig. 3a, b, respectively.

Case 2: Nonlinear motion

For the second test video, with Kalman filter, tracking of nonlinear motion of pedestrian in frame 185 failed as shown in Fig. 4a but particle filter overcomes nonlinear motion successfully as shown in frame 185 in Fig. 4a.

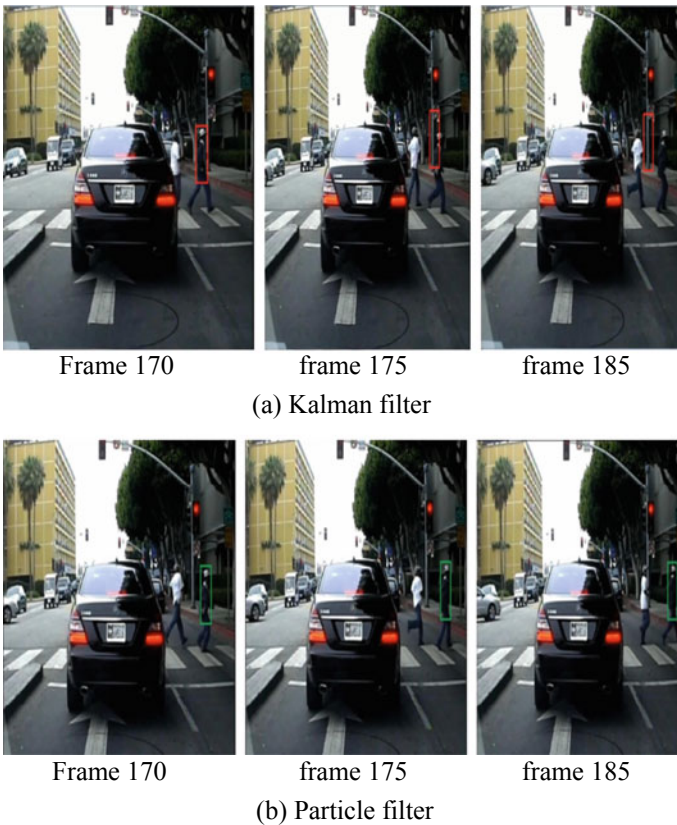


Fig. 4 Comparison of Kalman and particle filter

5 Conclusion

Pedestrian detection and tracking using HOG descriptors particle filter are implemented and tested for short video sequences from standard database. As compared to Kalman filter, it is able to overcome occlusion as well as nonlinear motion on the pedestrian. There is a tradeoff between accuracy and the processing time.

References

1. Prioletti, A. Part-based pedestrian detection and feature-based tracking for driver assistance: Real-time algorithm filtering approach. *Computer Science and Information Systems*.
2. Aghajanian, J., Warrell, J., Prince, S. J. D., Li, P., Rohn, J. L., & Baum, B. Patch-based within-object classification. In *12th International Conference on Computer Vision (ICCV)*.
3. Dalal, N., & Triggs, B. Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.
4. Huang, S., & Hong, J. *Moving object tracking system based on Camshift And Kalman Filter*.
5. Owczarek, M., Barański, P., & Strumiłło, P. *Pedestrian tracking in video sequences: A particle filtering approach*.
6. Fen, X., & Ming, G. (2010). Pedestrian tracking using particle filter algorithm. In *International Conference on Electrical and Control Engineering*.
7. Zhang, L., & Li, L. (2013). Improved pedestrian detection based on extended histogram of oriented gradients. In *2nd International Conference on Computer Science and Electronics Engineering, ICCSEE*.
8. Mittal, S., Prasad, T., Saurabh, S., Fan, X., & Shin, H. (2012). Pedestrian detection and tracking using deformable part models and Kalman filtering. In *ISOCC*.
9. Isard, M., Blake, A. CONDENSATION—Conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29.
10. Nkosi, M. P., Hancke, G. P., & dos Santos, R. M. A. *Autonomous pedestrian detection*. IEEE.

Exploiting Parallelism Available in Loops Using Abstract Syntax Tree



Anil Kumar and Hardeep Singh

Abstract Performance of a program depends on two factors: better hardware of the executing machine and exploiting parallelism for concurrent execution. Loops with multiple iterations provide efficient parallelism in an application, and are used to reduce overall execution time and also to increase performance. Abstract Syntax Tree (AST) can be used as an effective tool for exploiting parallelism at the compiler level. This definitely saves time and automates the decomposition of a parallel job that is to be executed in parallel framework. AST can be used to detect loops in the source code, therefore this approach can be used to design a new parallel computing framework where simple codes written for normal machines can be parallelized by the framework itself.

Keywords Parallel computing · Abstract Syntax Tree · Job decomposition · Parallelism · Parallel framework

1 Introduction

Source codes of software are becoming huge and complex. Compiling large codes is a time-consuming process. Parallel compilation of codes will help in reducing the time complexity. Parsing is the phase of a compiler which requires a significant amount of time for compilation. Many techniques have already been developed to extract the parallelism available in a parser. AST is proposed to be an effective way for exploiting parallelism at the compiler level via parsing and decomposing the code in this study. Studies of parallel applications so far suggest that achieving high performance with these applications is very significant. Not only do we find

A. Kumar (✉)

Department of Computer Engineering and Technology, Guru Nanak Dev University,
Amritsar 143005, India
e-mail: anil.gndu@gmail.com

H. Singh

Department of Computer Science, Guru Nanak Dev University, Amritsar 143005, India
e-mail: hardeep.dcse@gndu.ac.in

© Springer Nature Singapore Pte Ltd. 2019

N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_47

563

adequate parallelism in the program, but it is also important that we minimize the synchronization and communication overheads in the parallelized program. In fact, it is common to find parallel programs that are not able to run fast due to overheads of parallel execution. It is therefore required to increase the granularity to reduce the synchronization problems [1, 2]. We have studied two forms of parallelism based on granularity since these exploit parallelism available in loops which, further, is one of the techniques to improve the performance of the computer system.

To exploit loop parallelism at the instruction level, fine-grained parallel architectures are used. At runtime, various forms of available dependencies between operations in a program must be checked either with the help of compiler (which would be a static check) or with the help of hardware (to ensure that only independent operations are issued simultaneously and this is a kind of dynamic check). To extract maximum parallelism, the dependence-checking technique is used and not only it examines the basic blocks of the program but also goes beyond the boundaries of these blocks to find maximum independent operations from various iterations. On the other hand, coarse-grained architectures exploit available parallelism in loops by scheduling the entire set of iterations on separate processors. Parallelism can be exploited from two types of loops, those with cycles in their dependence graphs, and those with no cross-iteration dependencies. Both types can be executed on either fine-grained or coarse-grained parallel architectures but as a consequence, different performance judgments will be depicted by the two architectures. As the iterations of a loop can be executed several times, so they provide pronounced parallelisms available in an application. To exploit this parallelism, the suitable technique depends on the architecture of the executing parallel machine and the characteristics of each single loop. An optimum strategy for loop and machine architecture is thus required [3].

2 Forms of Parallelism

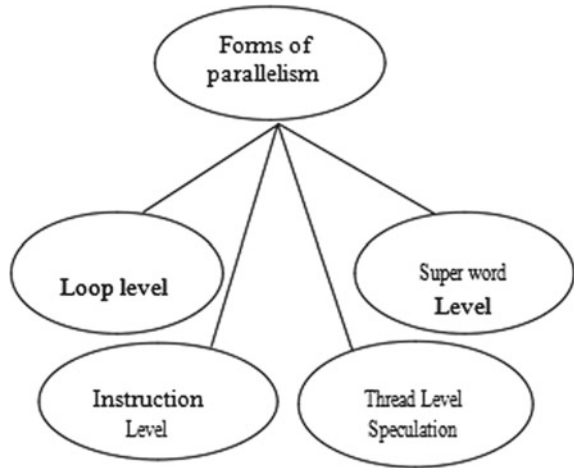
Figure 1 shows the various forms of parallelism which are further discussed in brief.

2.1 Loop Level Parallelism

Loop Level Parallelism is that form of parallelism, which is used to exploit parallelism among iterations of the loop. A simple example to understand the basic loop parallelism is [4]:

```
for (i = 1; i <= 500; i = i + 1)
    a[i] = a[i] + a[i + 1];
```

Loop Level Parallelism can be exploited by unrolling loops either dynamically, via branch prediction, or statically, via loop unrolling using compiler. The loops

Fig. 1 Forms of parallelism

with dependencies are executed using the do-across model based on coarse-grained architecture and software pipelining as the scheduling strategy based on fine-grained architectures. However, in these loops, explicit synchronization is not required [3]. The loops with no dependences are executed using do-all model based on coarse-grained architecture and software pipelining as the strategy based on fine-grained architecture. Due to less complexity and overheads, most of the frameworks use do-all loops for exploiting loop parallelism.

2.2 *Instruction Level Parallelism*

Instruction Level Parallelism (ILP) is a technique which simultaneously executes the sequential instructions from a program on multiple functional pipelining units [5]. For example,

1. $x = y + z$
2. $s = a - b$
3. $m = x * s$

Operations 1 and 2 are independent of each other, so they can be executed at the same time. But operation 3 depends on the outcome of operations 1 and 2, thus it can be calculated only after their completion [4]. ILP is helpful in exploiting Loop Level Parallelism [6, 7]. It has been observed that usually the runtime of the program gets reduced when we use the ILP approach because the execution time of considerable instructions gets superimposed over each other using this technique for parallelism. This approach is also very expensive. Machines with such designs increase the complexity in control logic and thus limit performance. Hence, ILP is not suitable for all conventional high-level language compilers [8].

2.3 Thread Level Speculation

Thread Level Parallelism is a form of parallel computing which uses the technique of distributing the execution of threads across different parallel processor nodes [9]. Focusing only on loops does not yield complete parallel potential of an application. Hence, to uncover the hidden Loop Level Parallelism and to strengthen parallelism in general purpose programs, Thread Level Speculation is required [10–12]. Thread level speculation hardware support, along with the use of several code transformations, has made it possible to expose the hidden Loop Level Parallelism in an application. In such architectures, the threads are extracted from sequential programs and are run in parallel. Each iteration of the loop can turn out to be a speculative thread. The degree of parallelism available in a loop is determined by dependences; more the independence between iterations more is the degree of parallelism. In addition to this, if we consider parallelism coverage, then choosing outer do-across over an inner do-all loop becomes more beneficial for parallelism coverage. Both degrees of parallelism, as well as parallelism coverage, are the main factors for loop speculation. The Procedure Level Speculation can also be used to improve the available parallelism for overcoming the fact that procedures are less popular than loops as a target of parallelization. It is also studied that simultaneous multithreading can convert the thread level parallelism to Instruction Level Parallelism because mostly superscalar processors exploit Instruction Level Parallelism by executing pipelining.

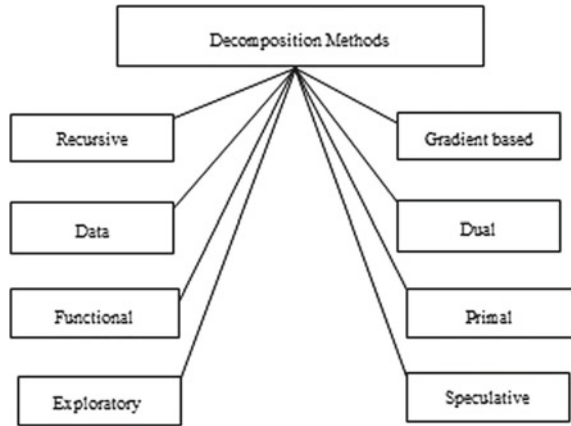
2.4 Superword Level Parallelism

Superword Level Parallelism (SLP) is a basic block vectorization technique based on loop unrolling. It can be a better replacement for all of the above as has been studied that it can exploit loop parallelism both across loop iterations, as well as within the basic blocks [13]. It is closely related to ILP. However, studies have shown that there is a need to recover the basic blocks that have not been vectorized sufficiently by this technique.

3 Decomposition Methods

Decomposition techniques serve as the best methods to make efficient use of parallel applications. These techniques divide a computation into a local part. This part does not require any inter-process intervention during its generation, and it mainly involves communication between neighboring and distant processors [14, 15]. When a problem is decomposed into a large number of tasks, it is called Fine-grained decomposition; otherwise, it is called coarse-grained decomposition. However, the nature of the problem decides which one is better, as the large number of tasks ensures better

Fig. 2 Decomposition techniques



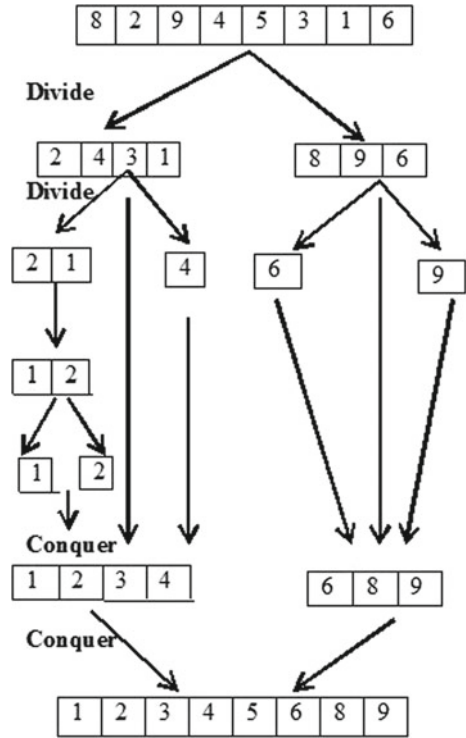
scalability with more concurrency and hence more utilization of resources. Lesser number of tasks usually includes less communication overheads. Figure 2 enlists the decomposition techniques which are as follows:

- Recursive Decomposition
- Data Decomposition
- Functional Decomposition
- Exploratory Decomposition
- Speculative Decomposition
- Primal Decomposition
- Dual Decomposition
- Gradient-Based Decomposition.

3.1 Recursive Decomposition

It is generally suited to problems that are solved using the Divide and Conquer strategy. A given problem is first decomposed into a set of subproblems. These subproblems are recursively decomposed further, until a desired granularity is reached [15]. The results of the subproblems are merged together when needed. Usually, smaller tasks are independent of one another, so they can be executed in parallel. So, good scalability can be achieved by sorting algorithms in parallel often use this approach for decomposition. Quad-tree, Oct-trees, and K-trees are the few approaches based on recursive decomposition for parallelization [16]. Figure 3 shows the task dependency graph of Quicksort algorithm based on recursive decomposition. The problem is solved using Divide and Conquer technique.

Fig. 3 Quicksort task dependency graph based on recursive decomposition [15]



3.2 Data Decomposition

Data Decomposition technique is used when data structures carrying bulk of similar data need to be processed. The data groups form tasks. These can be input data, output data, or even intermediate data. The decompositions of the computation into tasks are done by using the Owner Computes Rule [8]. All data forms can be decomposed.

In Fig. 4, the input–output matrices are first partitioned into sub-matrices then these are further decomposed into tasks. The tasks are independent of each other and thus, it is easily solved and scales well [15, 17].

3.3 Functional Decomposition

The technique of decomposing functions into tasks is known as Functional Decomposition. These tasks are executed concurrently on different data by processes which use the pipeline approach. Instead of its performance simplicity, it does not scale well because enough functions are not available for splitting. As an example, Fig. 5 shows the flow of data in counting word problem using pipelining.

(a)

$$\begin{pmatrix} X_{1,1} & X_{1,2} \\ X_{2,1} & X_{2,2} \end{pmatrix} \cdot \begin{pmatrix} Y_{1,1} & Y_{1,2} \\ Y_{2,1} & Y_{2,2} \end{pmatrix}$$

↓

$$\begin{pmatrix} Z_{1,1} & Z_{1,2} \\ Z_{2,1} & Z_{2,2} \end{pmatrix}$$

(b)

Task1: $Z_{1,1} = X_{1,1}Y_{1,1} + X_{1,2}Y_{2,1}$
 Task2: $Z_{1,2} = X_{1,1}Y_{1,2} + X_{1,2}Y_{2,2}$
 Task3: $Z_{2,1} = X_{2,1}Y_{1,1} + X_{2,2}Y_{2,1}$
 Task4: $Z_{2,2} = X_{2,1}Y_{1,2} + X_{2,2}Y_{2,2}$

Fig. 4 **a** The input–output sub-matrices are formed. **b** Matrices are decomposed into tasks

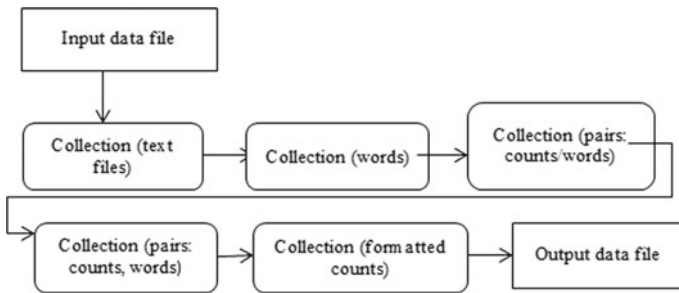
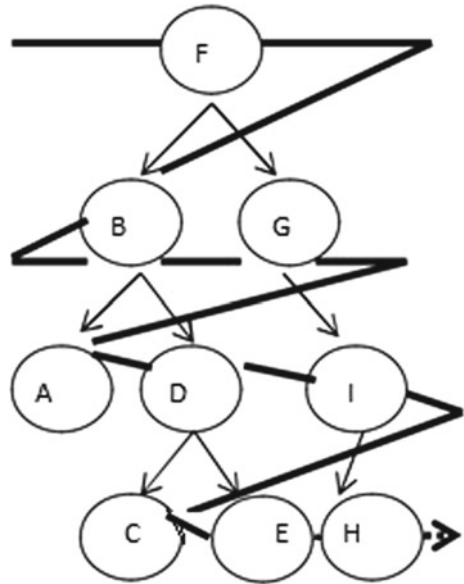


Fig. 5 Pipeline data flow in Word Count [5]

3.4 Exploratory Decomposition

Exploratory Decomposition is a technique which first searches the predefined space, and then partitions it into tasks to process them concurrently. It is a special case decomposition which is generally not applicable [15, 18]. An example is breadth-first search in trees as shown in Fig. 6. Traverse order are: F, B, G, A, D, I, C, E, H.

Fig. 6 Breadth first traversal of tree



3.5 Speculative Decomposition

Sometimes, there are many functions available in a program and only one of those needs to be carried depending upon execution of some condition, such as switch statement. All such functions are turned into tasks and carried out before the evaluation of the condition. This is called Speculative Decomposition [15, 19]. As soon as the condition is evaluated, the result of only one task is used, and the rest are discarded. But this decomposition does not make good use of resources, and thus is not popular.

The different decomposition methods described so far can be combined together to form hybrid decompositions which usually overcome the limitations of the comprising decompositions and often propose good solutions.

3.6 Primal Decomposition

Primal Decomposition is the optimization technique which solves the structured optimization problems in mathematics. It uses the master-slave approach for decomposition. The master problem manipulates the primal variables (variables comprising linear objective function) and subproblems are solved in parallel [20].

Table 1 Summarized table of various decomposition methods

Decomposition methods	Granularity	Scalability	Communication overheads	Time/space complexity
Recursive	Coarse grain	High	Low	Low
Data	Fine grain	High	High	High
Functional	Fine grain	Low	High	Low
Exploratory	Coarse grain	Low	High	High
Speculative	Coarse grain	Low	High	Low
Primal	Coarse grain	High	High	High
Dual	Fine grain	High	High	High
Gradient based	Fine grain	High	Medium	Medium

3.7 Dual Decomposition

Dual Decomposition is an optimization technique which uses sub-gradient algorithm for the master [20]. It is obtained from Lagrangian Formation [32]. It solves the dual subproblems in parallel and updates the dual variables (variables comprising linear objective function) iteratively. The dual problem solution defines lower bound for the primal solution.

3.8 Gradient-Based Decomposition

The Decomposition Optimization Algorithm, also known as Path-following Gradient-Based Decomposition Method, decomposes the problem into smaller sub-problems by using Dual Decomposition [21]. It is a combination of three techniques, namely Dual Decomposition [20], Smoothing [22], and Lagrangian Relaxation Method [23].

Table 1 shows the summary of decomposition techniques and their evaluation based on parameters such as Granularity, Scalability, Communication Overheads, and Time/Space Complexity.

4 Code Parsing Using Ast and Decomposition

Abstract Syntax Tree is used to represent the syntactic objects used by the systems that manipulate the programs, formulas, rules, etc. It is the outcome of the syntax analysis phase of the compiler of the program [24, 25]. Matching, substitution, and unification

are its main operations. The AST matching simply notifies the user about the changes which the global variables and functions in the program undergo. Basically, it is considered that these changes prove to be very useful from the point of software evolution because it is considered to be helpful for updating software. High order Abstract Syntax Trees is the variations of the basic Abstract Syntax Trees. For formal software development, there is need of a language generic environment; for that purpose high order AST has been designed. From parallel computing point of view, every parallelizer consists of various compiler passes. Each compiler uses AST as a data structure to represent the structure of the source code. It is studied [3] that the parallelism in do-all loops is useful in reducing the execution time of the program. AST can be used to detect the loops in the source code. The concrete structure of the program is a part of the definition of the language which is basically defined by the context-free language, whereas the Abstract Syntax Tree is a part of the implementation of the program and is defined by the tree structure [26, 27]. Thus, it can be said that the essential syntactic information cannot be given by the concrete structures and parse trees, which sometimes contains useless information. Thus, the Abstract Syntax Trees came up as a solution. Further, it has been studied that high-order syntax trees were not able to fulfill requirements such as accounts of structural index and induction, and recursive equation for abstract syntax. Abstract syntax with variable binding presents the algebraic (categorical) view [26] of syntax, which to some extent serves as one of the satisfactory solutions to fulfill the abovementioned requirements. Abstract syntax is the initial such model, with the algebra structure obtained as the solution to a recursive equation and substitution defined by an associated structural recursion.

4.1 Methodology

AST is parsing technique which can make the relevant information available at compiler level by parsing the source code of the program to exploit loop parallelism. Algorithm 1 is the explanation of the flowchart shown in Fig. 1. This procedure explains how the parser generates the AST.

Algorithm 1: Create AST

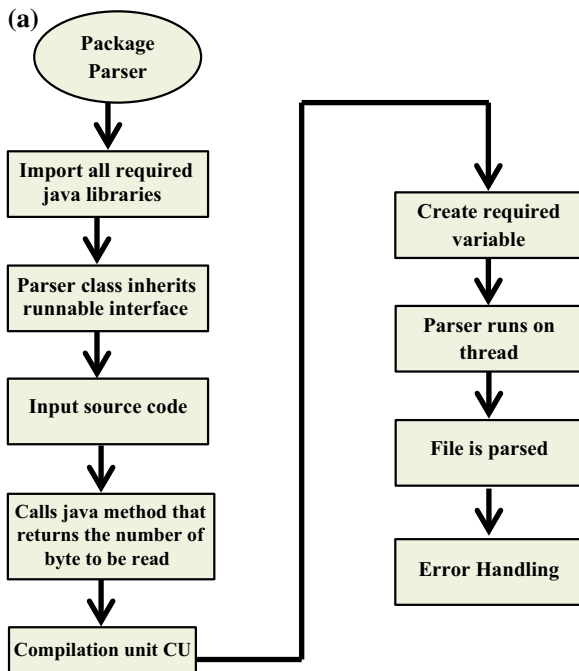
1. LOAD Parser
2. INPUT source code
3. READ number of bytes of source code
4. COMPILE source code
5. RUN Parser
6. GENERATE parsed code.

7. HANDLE errors.
8. CREATE variables, data structures for parsed code
9. CREATE database
10. ALLOCATE unique counter variable to each database table
11. GENERATE AST
12. INCREMENT counter variable on visiting node of AST

The detailed description of flowchart stages is shown below:

- **Loading parser:** Parser is imported from Java package “Package parser” and the loader links all the required Java libraries for the parser.
- **Input source code:** Source code is given as an input to the parser. Since Java is a strictly typed language so the source code should strictly follow the naming and typing rules. If possible, different names should be used for each element in coding.
- **Reading number of bytes of source code:** Parser calls the Java method that returns the number of bytes of the source code to be read by the parser.
- **Compiling code:** Source code is compiled using compilation unit CU and the variables are created.
- **Run parser:** Parser runs as a different thread by implementing Java’s runnable interface.
- **Generating parsed code:** Source code is parsed.

Fig. 7 a Source code parsing. b Generate Abstract Syntax Tree



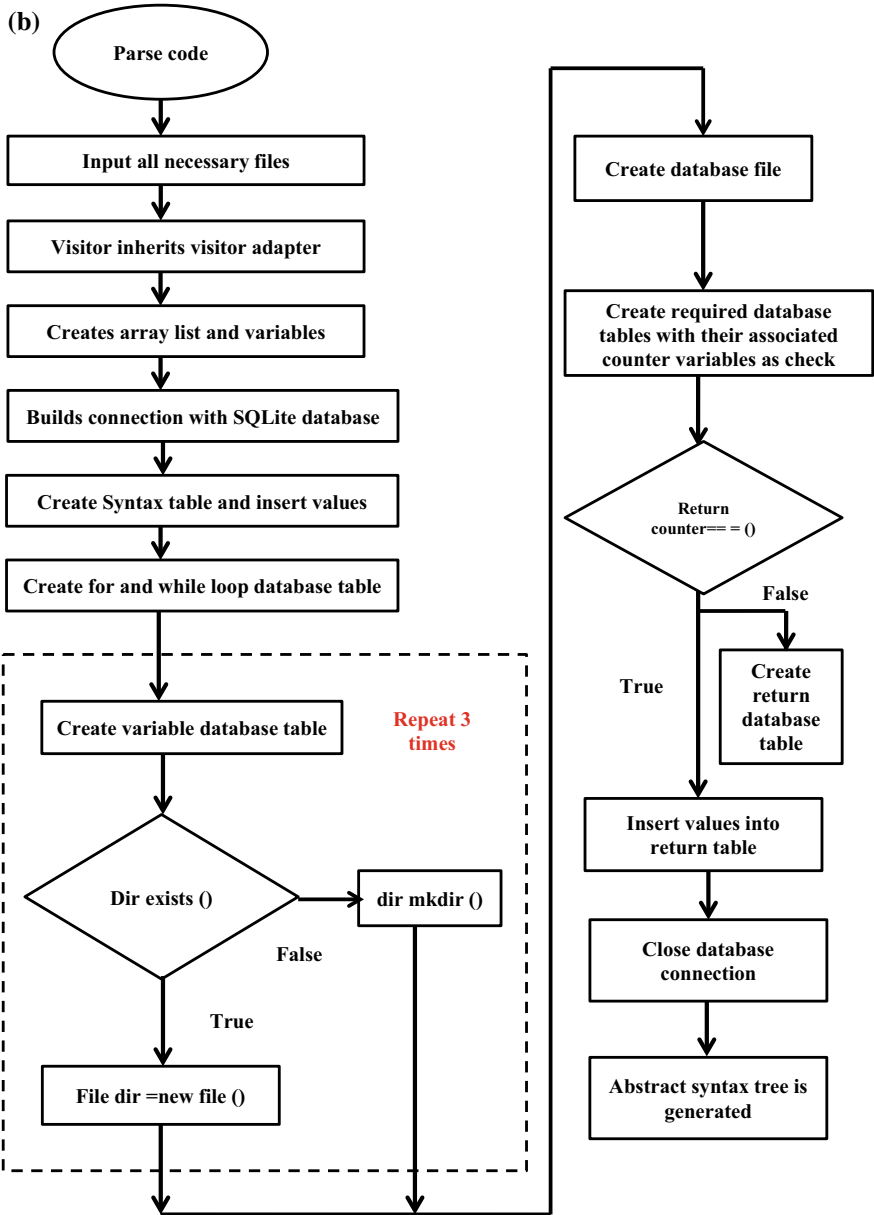


Fig. 7 (continued)

- **Handling errors:** Errors are handled with various Java handling techniques.
- **Creating data items:** Data structures like array lists, and variables are created for the parsed code.
- **Create database:** Database is created when the parser builds connection with SQLite database module.
- **Allocating counter variable:** A unique counter variable is allocated to each database table of loops, variables, etc.
- **Generating AST:** Abstract Syntax Tree is generated.
- **Incrementing counter variables:** Each counter variable is incremented in database for each useful visit to the node of AST. The different module in Java is created and named as visitor which inherits Visitor Adapter and imports all required Java libraries and packages.

The flowcharts (Fig. 7) explain how the Abstract Syntax Tree is generated. This AST will help to detect the loops available in the source code and enables the user to choose which among them are to be targeted to achieve parallelism. Moreover, the parsed information can be converted back to source code using the unparsing technique and this helps in modifying the code for parallelism.

5 Conclusion

With the support of Abstract Syntax Tree, the source code of a parallel job can be parsed and relevant information about different structures used in the code can be revealed at the compiler level. This information is useful for any parallel framework to parallelize a sequential algorithm with for-loops. It is clear from the literature survey that the loop parallelism for do-all loops seems to be the most practical and common within the programs. So, this type of parallelism is considered as the target to be exploited for parallelization. AST facilitates to decompose the problem at the compiler level, which can reduce the complexity at design and coding of parallel jobs using special parallel programming tools. This further reduces the need to know more parallel language constructs to parallelize the given job. This approach of using AST output at the compiler level can be used to design new parallel computing frameworks where simple codes, written for normal machine, can be parallelized by the framework itself.

References

1. Parhami, B. (2002). *Introduction to parallelism. Introduction to parallel processing: Algorithms and architectures* (pp. 3–23).
2. https://en.wikipedia.org/wiki/Instruction-level_parallelism.
3. Lilja, D. J. (1994). Exploiting the parallelism available in loops. *Computer*, 27(2), 13–26.
4. https://simple.wikipedia.org/wiki/Task_parallelism.

5. Kumar, R., & Singh, P. K. (2014). An approach for compiler optimization to exploit instruction level parallelism. In *Advanced Computing, Networking and Informatics* (Vol. 2, pp. 509–516). Cham: Springer.
6. Rau, B. R., & Fisher, J. A. (2003). *Instruction-level parallelism*.
7. Lo, J. L., Emer, J. S., Levy, H. M., Stamm, R. L., Tullsen, D. M., & Eggers, S. J. (1997). Converting thread-level parallelism to instruction-level parallelism via simultaneous multithreading. *ACM Transactions on Computer Systems (TOCS)*, 15(3), 322–354.
8. <http://www2.phys.canterbury.ac.nz/dept/docs/manuals/FORTRAN>.
9. <https://cloud.google.com/dataflow/examples/wordcount-example>.
10. Zhong, H., Mehrara, M., Lieberman, S., & Mahlke, S. (2008, February). Uncovering hidden loop level parallelism in sequential applications. In *IEEE 14th International Symposium on High Performance Computer Architecture, 2008. HPCA 2008* (pp. 290–301). IEEE.
11. Tullsen, D. M., Eggers, S. J., & Levy, H. M. (1995, June). Simultaneous multithreading: Maximizing on-chip parallelism. In *22nd Annual International Symposium on Computer Architecture, 1995. Proceedings* (pp. 392–403). IEEE.
12. Wall, D. W. (1991). *Limits of instruction-level parallelism* (Vol. 19, No. 2, pp. 176–188). ACM.
13. Larsen, S., & Amarasinghe, S. (2000). *Exploiting superword level parallelism with multimedia instruction sets* (Vol. 35, No. 5, pp. 145–156). ACM.
14. Quarteroni, A., & Valli, A. (1996). *Domain decomposition methods for partial differential equations*.
15. Johnson, T. A., Eigenmann, R., & Vijaykumar, T. N. (2007, March). Speculative thread decomposition through empirical optimization. In *Proceedings of the 12th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (pp. 205–214). ACM.
16. Jackins, C. L., & Tanimoto, S. L. (1983). Quad-trees, Oct-trees, and K-trees: A generalized approach to recursive decomposition of euclidean space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5, 533–539.
17. Stone, R. B., & Wood, K. L. (2000). Development of a functional basis for design. *Journal of Mechanical Design*, 122(4), 359–370.
18. Arabnia, H. R. (1990). A parallel algorithm for the arbitrary rotation of digitized images using process-and-data-decomposition approach. *Journal of Parallel and Distributed Computing*, 10(2), 188–192.
19. Jojic, V., Gould, S., & Koller, D. (2010, June). Accelerated dual decomposition for MAP inference. In *ICML* (pp. 503–510).
20. Devaurs, D., Siméon, T., & Cortés, J. (2013). Parallelizing RRT on large-scale distributed-memory architectures. *IEEE Transactions on Robotics*, 29(2), 571–579.
21. Neamtiu, I., Foster, J. S., & Hicks, M. (2005). Understanding source code evolution using abstract syntax tree matching. *ACM SIGSOFT Software Engineering Notes*, 30(4), 1–5.
22. Dinh, Q. T., Necoara, I., & Diehl, M. (2014). Path-following gradient-based decomposition algorithms for separable convex optimization. *Journal of Global Optimization*, 59(1), 59–80.
23. https://web.stanford.edu/class/ee364b/lectures/decomposition_slides.pdf linear algebra with applications 7.7-8 (2000), 687–714.
24. Pfenning, F., & Elliott, C. (1988, June). Higher-order abstract syntax. In *ACM SIGPLAN Notices* (Vol. 23, No. 7, pp. 199–208). ACM.
25. Fiore, M., Plotkin, G., & Turi, D. (1999). Abstract syntax and variable binding. In *14th Symposium on Logic in Computer Science, 1999. Proceedings* (pp. 193–202). IEEE.
26. Wile, D. S. (1997, May). Abstract syntax from concrete syntax. In *Proceedings of the 19th International Conference on Software Engineering* (pp. 472–480). ACM.
27. Lim, A. W., & Lam, M. S. (1997, January). Maximizing parallelism and minimizing synchronization with affine transforms. In *Proceedings of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* (pp. 201–214). ACM.

A Study on Cooperation and Navigation Planning for Multi-robot Using Intelligent Water Drops Algorithm



D. Chandrasekhar Rao and Manas Ranjan Kabat

Abstract The shortcomings of the existing multi-robot navigation planning methods in an unknown environment are studied extensively. An efficient approach for resolving these issues is addressed in the current study using a novel nature inspired-Intelligent Water Drops (IWD) algorithm [1]. The robustness of the proposed method for multi-robot navigation in an unknown environment is validated through V-Rep simulator. The performance of the algorithm is verified through simulation outcomes in terms of total path travelled, total path deviation, number of turns, and execution time. Further, the efficiency of the proposed method is compared with the existing state of the art to verify its potency. The simulation outcomes reveals that the proposed method takes in an improvement of 9.93%, 35.26%, 33.33%, and 13.04% in terms of the total path travelled, total path deviation, number of turns, and total execution time for all the robots to arrive their target, respectively, as compared to the existing state of the art. Moreover, the current study confirms the superiority of the proposed approach as compared to the existing state of the art in terms of generating optimal and safe navigation path for individual robots.

Keywords Multi-robot · Navigation planning · Waypoints · Intelligent water drops · Static environment

1 Introduction

In recent years, the application of robotics has been evolved as the most cost-effective approach in many fields [2–8]. However, navigation planning is one of the key concerns in most of the applications employed with mobile robots. In navigation planning

D. Chandrasekhar Rao (✉)

Department of Information Technology, VSSUT, Burla, Sambalpur 768017, Odisha, India
e-mail: dcrao_it@vssut.ac.in

M. R. Kabat

Department of Computer Science & Engineering, VSSUT, Burla, Sambalpur 768017, Odisha, India
e-mail: manas_kabat@yahoo.com

© Springer Nature Singapore Pte Ltd. 2019

N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_48

577

problem, the main objective of the mobile robot is to find an optimal and safe path while navigating from its source to destination. Thus, the efficiency of the mobile robot needs to be enhanced such that it can reach the target through a shortest path with minimal time and energy utilization. Various research findings have been addressed in this domain for improving the autonomy and decision accuracy of the mobile robot to resolve these issues.

The working environment of mobile robots may be regarded as static or dynamic. In a static environment, the obstacle and target positions are stationary, however, these may be dynamic in the dynamic environment. Further, the navigation planning of a mobile robot in an environment can be classified as global or local. In global planning, the mobile robot has prior knowledge about the environment and decides the collision-free path in advance. In local planning, the mobile robot generates collision-free path on the go with no prior knowledge about the environment. Local planning is sometimes referred to as online planning. Global and local planning is most suitable for static environment and dynamic environment. However, both of these planning may be combined to improve the efficiency of the mobile robot in a complex environment [9].

In this study, we proposed a novel approach for multi-robot navigation planning in a static environment using nature-based IWD algorithm. The proposed algorithm is efficient to generate a safe and an optimal path for the individual robot with minimum time and energy utilization. The effectiveness of the proposed algorithm is further verified with the existing state of the art [10].

The rest of this paper is structured as follows. In Sect. 2, the related work for multi-robot navigation has been elaborated. The problem formulation, operating principle of the basic IWD algorithm and its implementation in multi-robot navigation are discussed in Sect. 3. Section 4 briefs the simulation setup and performance evaluation of the proposed method. Finally, the work is concluded in Sect. 5.

2 Related Work

Navigation planning is the primary concern in most of the mobile robot applications. The approaches employed in resolving the robot navigation problem can be broadly categorized as traditional or heuristics [11]. The state of the art in this field is discussed below.

The traditional approaches comprise of Cell decomposition [12], the probabilistic Roadmap approach [13], Voronoi Diagram [14], and Artificial Potential Field [15] are employed to solve the Mobile Robot Navigation Problem (MRNP). However, these approaches are trapped in local minima, incompetence in the presence of large obstacle, and computational cost is more. Recently, with the development of heuristic approaches, it is embedded in many applications to resolve the issues associated with MRNP. These approaches have a good potential for global exploration, however, each approach has its own limitation. This boosts the researcher to employ heuristic approaches for addressing mobile robot navigation planning. Many heuris-

tic approaches such as Fuzzy Logic (FL), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Gravitational Search Algorithm (GSA), Differential Evolution (DE), and hybridization of these approaches have employed to solve MRNP.

FL is extensively employed to resolve the navigation planning problem of mobile robots, which utilizes the predicted behaviour of humans to deal with uncertainties. The FL incorporates the characteristics of human perception for resolving real-world problems. The authors in [16] proposed a Fuzzy Logic Controller (FLC) for MRNP to deal with an environment consisting of static obstacles. The rule set for FLC is constructed using the distance of obstacles from right, left, and front side of mobile robots. Simulation has been carried out to prove the effectiveness of the proposed FLC in an environment consisting of thousands of robots and obstacles. In [10], the authors proposed Obstacle avoidance FLC (OA-FLC) and Obstacle avoidance FLC (OA-FLC) for escaping from obstacles and maintaining the heading direction of robots toward goal in the static as well as the dynamic environment respectively. However, the proposed method has not guaranteed optimal path always. Though FL is efficient for simulating the decision making capability of human being, but it is incompetent in the selection of appropriate membership functions for finding optimal path. GA is another widely used method based on the natural selection procedure in the field of robot navigation. Ozkan et al. [17] proposed multi-robot navigation using a hierarchical-oriented genetic algorithm (HOGA) to optimize the time required by robots rather than the path travelled by robots. In [18], navigation planning based on GA is proposed in a dynamic environment. However, GA is inefficient in controlling population difference on a grid-like environment and may occur in an early convergence. Besides GA, PSO is another population-based method, employed in MRNP. PSO [19] and its improvement [20] are proposed for generating the optimal trajectory of robots in an unknown environment. Further, GSA based on the gravitational law of masses is employed to address the MRNP for generating the optimal path. GSA [21] and its improvement [22] are proposed to find the optimal trajectory by avoiding the static and dynamic obstacles in an unknown environment. However, the convergence rate of PSO and GSA are slow in an iterative process and may be trapped at local minima. Rao et al. [23] proposed DE algorithm for multi-robot navigation in a clutter environment. Similar to GA, DE uses crossover, trial, and selection operators to generate a safe path for mobile robots in an unknown environment. However, it does not guarantee an optimal path. Furthermore, hybridization of many heuristic approaches [24–26] has been proposed for addressing MRNP.

The above-discussed works generate collision-free path for robots in different environments using classical and soft computing approaches. Though, the proposed methods able to generate safe path for mobile robots but does not guarantee optimal path always in terms of shortest path, time and energy utilization for navigating the robot from source to destination. Furthermore, in a large problem space, most of the algorithms trapped in local minima, oscillate in the presence of large obstacles, and also fail to maintain a good balance between intensification and diversification. These factors motivated us to propose a novel approach to overcome the above-discussed shortcomings. Thus, a novel approach by utilizing the concept of IWD algorithm is proposed for multi-robot navigation in a static environment. The mobile robots

simulate the behaviour of IWD to find optimal and safe path in the world map. The proposed IWD algorithm is simple to implement, capable for global search and needs tuning of few control parameters in a static environment. The main highlights of this paper are as follows:

- i. The target function is formulated by considering the weighted sum of each individual objective function.
- ii. The proposed method is able to escape from static obstacles as well as teammates (other robots considered as dynamic obstacles) present in the environment.
- iii. The proposed method performance is verified through V-Rep simulator.
- iv. Further, the potency of the proposed method has been validated through the existing state of the art.

3 Proposed Multi-robot Navigation Planning

The configuration space for multi-robot navigation problem is consisting of multiple autonomous mobile robots and static obstacles. The primary issue is to identify an optimal, safe, and smooth path for individual mobile robots in the configuration space from their respective start to target position.

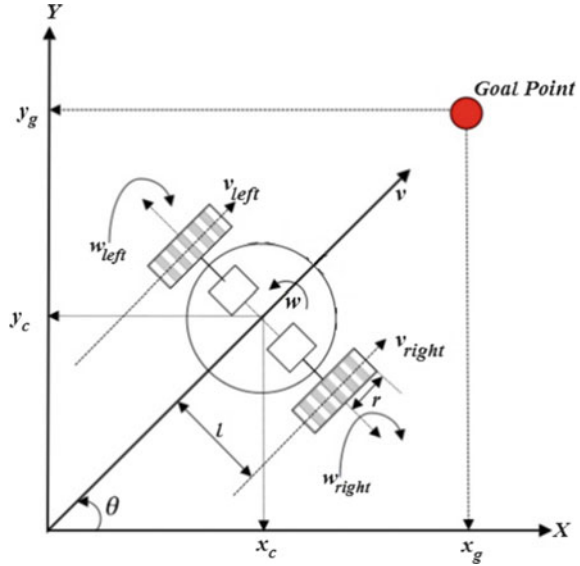
3.1 Problem Formulation

The problem space for the multi-robot navigation is considered as a two-dimensional world map. The environment consists of static obstacles of different shapes and two-wheeled multiple mobile robots. Every robot in the environment has predefined start and the goal position. The objective of the problem is to find an optimal and safe path for the individual robot from their respective start position to the goal position. In the initial setup, every robot's orientation is toward the goal position. The structure of a two-wheeled mobile robot is shown in Fig. 1.

The current position of a two-wheeled mobile robot in two-dimensional coordinate system is denoted by (x, y) and orientation of the robot with respect to abscissa is θ . All the robots in the environment are of identical structure with radius r . The angular and translational motion of the robot is denoted by ω and v , respectively. In order to navigate safely, every mobile robot generates successive waypoints by avoiding obstacles in their way. Let $\{p_1, p_2, \dots, p_m\}$ be the “ m ” waypoints generated by the mobile robot while navigating from source to goal. Thus, the total path travelled by an individual robot can be expressed as follows:

$$P_i = \sum_{j=0}^m \text{dist}(p_i^j, p_i^{j+1}) \quad (1)$$

Fig. 1 Structure of a two-wheeled mobile robot



where $\text{dist}(p_i^j, p_i^{j+1})$ represents the Euclidian distance between two successive way-points j and $j + 1$, p_i^0 and p_i^{m+1} is the start and the goal position of the i th robot, respectively. Finally, the objective function for minimizing the path cost for all the robots in the environment can be expressed as follows:

$$F_{\text{path}} = \min \left\{ \sum_{i=1}^n P_i \right\} \tag{2}$$

where P_i denotes the path travelled by i th robot. Further, to obtain collision-free path, the robot needs to maintain a minimum distance from obstacles as well as with its teammates during navigation in the environment. In order to escape from obstacles, a linear function can be formulated for evaluating the safety factor. It can be expressed as follows:

$$\begin{aligned} & \text{OBS}_{\text{avoid}}^i \\ &= \max_{1 \leq j \leq l, i \neq j} \left\{ \left\{ \begin{array}{ll} 0 & \text{dist}(\text{Pos}(R_i), \text{Pos}(\text{Obs}_j)) > \lambda_{\text{obs}} \\ \frac{1}{\text{dist}(\text{Pos}(R_i), \text{Pos}(\text{Obs}_j))} - \frac{1}{\lambda_{\text{obs}}} & \text{otherwise} \end{array} \right. \right\} \end{aligned} \tag{3}$$

where $\text{Pos}(R_i)$ is the position of i th robot, l is the number of obstacles surrounding R_i , $\text{Pos}(\text{Obs}_j)$ is the position of j th obstacles for $j = \{1, 2, \dots, l\}$, λ_{obs} is the constant safety factor between the obstacles and the corresponding mobile robot. However, the value of λ_{obs} can be chosen based on the influence radius of the obstacles. Here, the teammates (other robots) are considered as the dynamic obstacle for the current

robot. In order to maintain a safe path, the robot should maintain the minimum safety distance from the nearest obstacles. Thus, the objective function for safe path for multi-robot can be expressed as follows:

$$F_{\text{obs_avoid}} = \min_{1 \leq i \leq n} \{ \text{OBS}_{\text{avoid}}^i \} \quad (4)$$

Finally, the above-discussed objective functions for multi-robot navigation can be formulated into a single objective function. Thus, the target function can be viewed as a weighted sum of individual objective functions and can be expressed as follows:

$$F_{\text{fitness}} = \alpha_1 \cdot F_{\text{path}} + \alpha_2 \cdot F_{\text{obs_avoid}} \quad (5)$$

where α_1 and α_2 are the nonzero positive weight factor for total path travelled and safety factor for obstacle avoidance, respectively. In the present work the value of α_1 and α_2 are set to 0.6 and 0.4, respectively.

3.2 IWD Algorithm

IWD [1] is a novel population-based nature inspired optimization algorithm that simulates the behaviour of water drops in the river bed. IWD is able to provide a promising solution to various optimization problems [17]. The water drops of a river play a major role in forming the river path from source to destination. The water drops face many obstacles in its way and turn intelligently to avoid the obstacles. The observation made from the natural water drops that it forms an optimal river bed from source to destination in presence of obstacles. The two important properties of water drops such as the velocity and the soil accumulation govern the IWD to find an optimal path. The selection of subsequent sub path by IWD toward its destination depends on the soil concentration of the path. The relationship between these two properties is the driving force for the IWD to form an optimal path between the source and destination. The soil concentration on the path is inversely proportional to the velocity of IWD. Also, the amount of soil carried by IWD is an inverse relation with the time taken by IWD to traverse the sub path. The above-discussed property of IWD reveals the similarity between the paths finding behaviour of IWD with mobile robot navigation in the presence of obstacles. These factors motivated us to simulate the behaviour of IWD into a mobile robot to find an optimal path.

3.2.1 Implementation of IWD for Navigation Planning of Mobile Robots

The navigation path of a mobile robot can be represented through a graph $G = \{V, E\}$. Where, V and E represents the number of waypoints and the sub path between the two consecutive waypoints, respectively. Here, the IWD will be treated as the mobile robot. In the river bed, each IWD decides its next waypoint on the basis of the amount of soil in the sub path. Similar to IWD, the mobile robot chose its subsequent waypoint based on the obstacle distance from it. The distance of an obstacle is inversely proportional to the soil amount present in the path. As the concentration of soil is inversely proportional to the velocity of IWD, the mobile robot will reduce its velocity if it is nearer to the obstacles. Thus, the proposed IWD algorithm for multi-robot navigation can be formulated as follows.

- (a) Static parameters Initialization: Initialize the number of IWD (N_{IWD}), maximum number of iteration ($iter_{max}$) velocity updating parameters and soil updating parameters.
- (b) Dynamic parameters initialization: Initialize the set of visited waypoints by IWD ($V_{wp} = \phi$), initial soil in the path ($init_{soil}^{path} = 100$), initial velocity of IWD ($init_{vel} = 10$) and initial soil in IWD ($init_{soil}^{IWD} = 0$). The fitness of global best solution $F_{fitness}(IWD_{gbest})$ is set to $-\infty$.
- (c) The initial position of mobile robots is set as the initial position of IWD. Update V_{wp} with the initial position of IWD.
- (d) For each IWD, repeat (i) to (iv)
 - i. Compute the next waypoint for individual IWD using the following expression:

$$P_{i,j}^{IWD} = \begin{cases} \frac{f(\text{soil}(i,j))}{\sum_{1 \leq k \leq N_{IWD}, j \neq k, k \notin \text{Pos}(\text{Obs})} f(\text{soil}(i,k))} & \text{if } j \notin \text{Pos}(\text{Obs}) \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where

$$f(\text{soil}(i, j)) = \frac{1}{\varepsilon_{soil} + g(\text{soil}(i, j))} \quad (7)$$

and

$$g(\text{soil}(i, j)) = \begin{cases} \text{soil}(i, j) & \text{if } \min_{l \notin V_{wp}} \{\text{soil}(i, l)\} \leq 0 \\ \text{soil}(i, j) - \min_{l \notin V_{wp}} \{\text{soil}(i, l)\} & \text{otherwise} \end{cases} \quad (8)$$

where $\varepsilon_{\text{soil}} = 0.01$ is a small positive constant to avoid the constraint division by zero, $\text{soil}(i, j)$ is the concentration of soil between the two adjacent waypoints, the functions $f()$ and $g()$ is used to maintain a positive value to $\text{soil}(i, j)$. The new set of j th waypoints will be updated into V_{wp} .

- ii. For each IWD, the velocity is updated to move from i th waypoint to j th waypoint using following expression:

$$\text{Vel}_{\text{IWD}}(t+1) = \text{Vel}_{\text{IWD}}(t) + \frac{\alpha_v}{\mu_v + \lambda_v \cdot \text{soil}^2(i, j)} \quad (9)$$

where $\alpha_v = 1$, $\mu_v = 0.01$ and $\lambda_v = 1$ are the constant updating parameters for IWD velocity.

- iii. After updating the velocity of IWD, the amount of soil taken by the respective IWD during its journey on the sub path from i th waypoint to j th waypoint can be expressed as follows:

$$\delta\text{soil}(i, j) = \frac{\alpha_s}{\mu_s + \lambda_s \cdot \text{Time}(i, j; \text{Vel}_{\text{IWD}})} \quad (10)$$

and

$$\text{Time}(i, j; \text{Vel}_{\text{IWD}}) = \frac{\text{dist}(\text{Pos}(i), \text{Pos}(j))}{\max\{\varepsilon_{\text{vel}}, \text{Vel}_{\text{IWD}}\}} \quad (11)$$

where $\alpha_s = 1$, $\mu_s = 0.01$ and $\lambda_s = 1$ are the constant soil updating parameters for soil accumulation by IWD, $\text{Time}(i, j; \text{Vel}_{\text{IWD}})$ is the time taken by the IWD with a velocity of Vel_{IWD} to move from i to j th waypoint and $\varepsilon_{\text{vel}} = 0.001$ is the small positive constant to prevent the negative or zero value of Vel_{IWD} .

- iv. The soil amount in the sub path and accumulated by IWD then needs to be updated using the following expression:

$$\text{soil}(i, j) = (1 - \gamma) \cdot \text{soil}(i, j) - \gamma \cdot \delta\text{soil}(i, j) \quad (12)$$

$$\text{IWD}_{\text{soil}} = \text{IWD}_{\text{soil}} + \delta\text{soil}(i, j) \quad (13)$$

where γ is the small positive constant selected from (0, 1) and IWD_{soil} is the concentration of soil in the IWD.

- (e) After finding the solution for all IWD, evaluate the iteration best solution using the following expression

$$\text{IWD}_{\text{ibest}} = \min\{F_{\text{fitness}}(T_{\text{IWD}})\} \quad (14)$$

where IWD_{ibest} is the best solution in the current iteration based on the evaluation of each IWD (T_{IWD}) using the objective function stated in Eq. (5). Further, the sub path corresponding to the iteration best solution is updated using the following expression

$$soil(i, j) = \eta_s \cdot soil(i, j) + \eta_{IWD} \cdot \kappa(V_{wp}) \cdot IWD_{soil}^{ibest} \quad \forall (i, j) \in IWD_{ibest} \quad (15)$$

where IWD_{soil}^{ibest} is the soil accumulated by the iteration best IWD, $\kappa(V_{wp}) = 1/(|V_{wp}| - 1)$, η_s and η_{IWD} are the constants. After updating the soil concentration in the respective sub path, the global best solution for the current iteration can be updated using the following expression

$$IWD_{gbest} = \begin{cases} IWD_{gbest} & \text{if } F_{fitness}(IWD_{gbest}) \geq F_{fitness}(IWD_{ibest}) \\ IWD_{ibest} & \text{otherwise} \end{cases} \quad (16)$$

- (f) Repeat Step (a) to Step (e), until ($iter \neq iter_{max}$) and each IWD not reached the destination.

4 Simulation Result and Performance Evaluation

The multi-robot environment with obstacles is configured through V-Rep 3.4.0 [27]. The environment consists of four two-wheeled Pioneer p3dx mobile robots and eight different shapes of obstacle. The initial and target position of all the robot is predefined in the environment. The detailed specification of Pioneer p3dx can be found in [28]. The simulation program is carried out in a Laptop PC with Core i3 processor and 4 GB of RAM. The IWD algorithm is implemented using MATLAB 2016a and interfaced with V-Rep for controlling the robots. The size of the world map for simulation is $5 \times 5 \text{ m}^2$. The environment is consisting of four Pioneer p3dx mobile robot and seven obstacles. The Isometric view and top view of the environment is shown in Fig. 2a, b, respectively. The current position of mobile robots in Fig. 2b is the start positions of the robots and the sphere with different colour code represents the respective target position of the robots. Initially, all the robots heading direction is toward the target position.

The maximum speed of the robot is set to 20 cm/s. The simulation for multi-robot navigation was carried out by employing the IWD algorithm for 30 runs and the best performance of the algorithm is presented in Fig. 3. The intermediate configuration of robots during simulation is shown in Fig. 3a. The final configuration of robots where all the robots arrived at their predefined target positions is shown in Fig. 3b.

The performance of the IWD algorithm is evaluated using the simulation outcomes. The parameters such as total path travelled, path deviation, number of turns and execution time are considered for performance evaluation. The effectiveness of

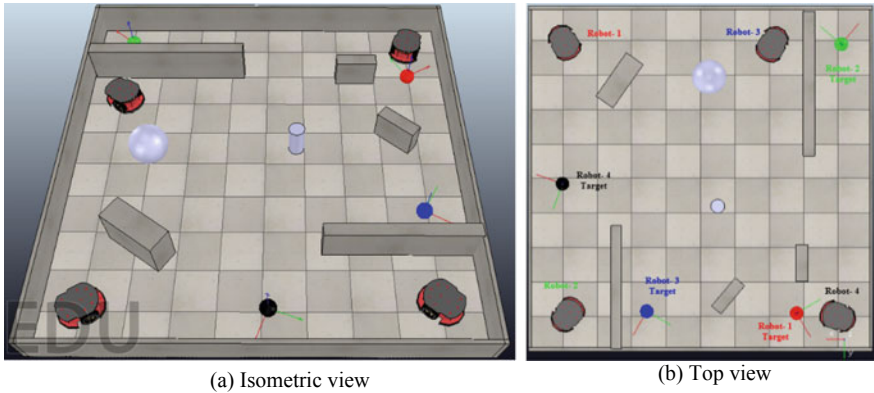


Fig. 2 Initial configuration of simulation environment

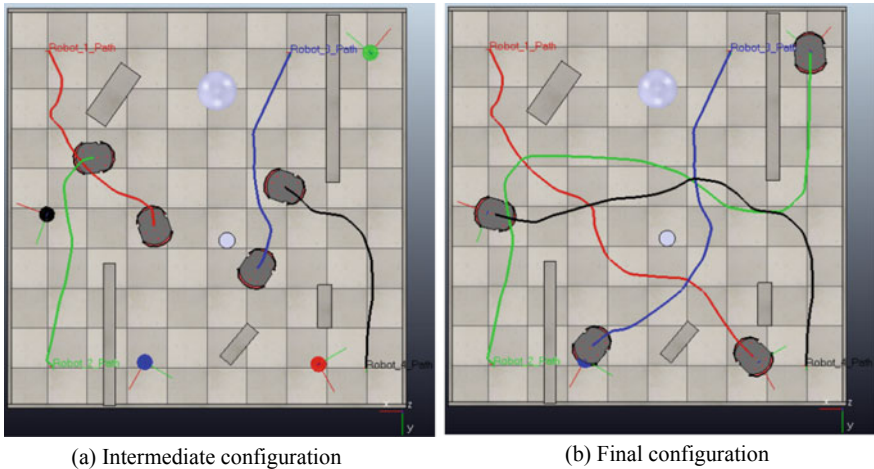


Fig. 3 Simulation output using IWD algorithm

IWD algorithm is further verified using the existing state of the art [10]. To verify the effectiveness of IWD algorithm, the state of the art [10] is implemented using Matlab and interfaced with V-rep using the same environment. The simulation is executed for 30 times for the state of the art [10] and the simulation outcomes are noted. The best performance out of 30 runs employing the state of the art [10] is shown in Fig. 4. The relative performance of IWD algorithm and the existing state of the art [10] has been verified using simulation outcomes for the above-discussed parameters. The simulation outcome of both the algorithms for performance evaluation is noted by taking the average of 30 runs and presented in Table 1. Further, the improvement of the proposed IWD algorithm over the state of the art [10] for multi-robot navigation problem is compared and presented in Table 2.

Table 1 Simulation outcome of IWD algorithm and existing state of the art

Robot	Position		Actual path (in cm)	Path travelled (in cm)		Path deviation (in cm)		No. of turns		Time (in s)	
	Initial	Target		IWD	[10]	IWD	[10]	IWD	[10]	IWD	[10]
Robot-1	(198, -197)	(-139, 195)	517	568	628	51	111	12	14	30.75	35.90
Robot-2	(193, 197)	(-204, -195)	558	827	864	269	306	12	19	45.74	52.60
Robot-3	(-104, -195)	(78, 193)	429	458	557	29	128	7	17	27.30	31.25
Robot-4	(-199, 200)	(200, 8)	443	587	660	144	217	13	16	33.70	37.50

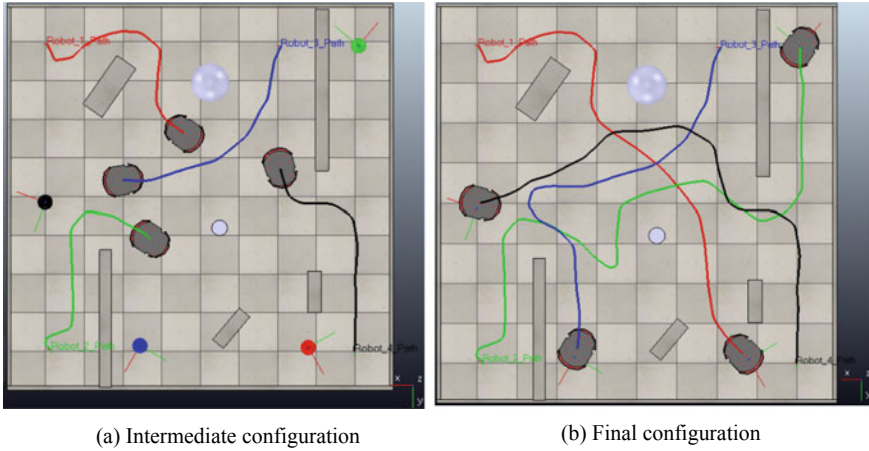


Fig. 4 Simulation output using the state of the art [10]

Table 2 Comparative analysis between IWD algorithm and the state of the art [10]

Algorithm	Total path travelled (in cm)	Total path deviation (in cm)	Total number of turns	Total execution time (in s)
IWD	2440	493	44	45.74
FLC [10]	2709	763	66	52.60
% of improvement	9.93	35.26	33.33	13.04

The simulation outcomes presented in Table 1 reveals that the individual robot performance employing IWD algorithm is superior to the existing state of the art [10]. Further, the relative percentage of improvement of the IWD algorithm as compared to the state of the art [10] in terms of total path travelled, total path deviation, total number of turns, and total execution time is 9.93, 35.26, 33.33, and 13.04, respectively.

5 Conclusion

In this a paper, a novel approach for solving multi-robot navigation problem using IWD algorithm is addressed. The objective function comprises of weighted sum of the distance to travel with respect to the target and obstacle avoidance is formulated for the problem. The subsequent waypoints for the mobile robot from its current position to target position are generated through the algorithm by evaluating the objective function. The proposed algorithm effectiveness is validated through simulation. Further, the potency of the proposed method is verified with the existing state of the art

through simulation. The simulation outcomes for multi-robot navigation reveal that the proposed algorithm outperforms the existing state of the art [10]. However, the performance of the algorithm is not verified in the presence of dynamic obstacles.

References

1. Shah-Hosseini, H. (2008). Intelligent water drops algorithm: A new optimization method for solving the multiple knapsack problem. *International Journal of Intelligent Computing and Cybernetics*, 1(2), 193–212.
2. Lottermoser, A., Berger, C., Braunreuther, S., & Reinhart, G. (2017). Method of usability for mobile robotics in a manufacturing environment. *Procedia CIRP*, 62, 594–599.
3. Stoyanov, T., Mojtahedzadeh, R., Andreasson, H., & Lilienthal, A. J. (2013). Comparative evaluation of range sensor accuracy for indoor mobile robotics and automated logistics applications. *Robotics and Autonomous Systems*, 61(10), 1094–1105.
4. Stephens, K. D., Jr. (2017, February 21). *Space exploration with human proxy robots*. U.S. Patent No. 9,573,276.
5. Sokolov, S., Zhilenkov, A., Nyrkov, A., & Chernyi, S. (2017). The use robotics for underwater research complex objects. In *Computational intelligence in data mining* (pp. 421–427). Singapore: Springer.
6. Bayat, B., Crasta, N., Crespi, A., Pascoal, A. M., & Ijspeert, A. (2017). Environmental monitoring using autonomous vehicles: A survey of recent searching techniques. *Current Opinion in Biotechnology*, 45, 76–84.
7. Krishna, K. R. (2017). *Push button agriculture: Robotics, drones, satellite-guided soil and crop management*. New York: CRC Press.
8. Bakhshpour, M., Ghadi, M. J., & Namdari, F. (2017). Swarm robotics search & rescue: A novel artificial intelligence-inspired optimization approach. *Applied Soft Computing*, 57, 708–726.
9. Zhang, H., Butzke, J., & Likhachev, M. (2012). Combining global and local planning with guarantees on completeness. In *International Conference on Robotics and Automation* (pp. 4500–4506).
10. Zhao, R., & Lee, H. K. (2017). Fuzzy-based path planning for multiple mobile robots in unknown dynamic environment. *Journal of Electrical Engineering & Technology*, 12(2), 918–925.
11. Masehian, E., & Sedighzadeh, D. (2007). Classic and heuristic approaches in robot motion planning—a chronological review. *World Academy of Science, Engineering and Technology*, 29(1), 101–106.
12. Lingelbach, F. (2004). Path planning using probabilistic cell decomposition. In *IEEE International Conference on Robotics and Automation* (Vol. 1, pp. 467–472).
13. Geraerts, R., & Overmars, M. H. (2004). A comparative study of probabilistic roadmap planners. In *Algorithmic foundations of robotics V* (pp. 43–57).
14. Bhattacharya, P., & Gavrilova, M. L. (2008). Roadmap-based path planning—using the Voronoi diagram for a clearance-based shortest path. *IEEE Robotics and Automation Magazine*, 15(2), 58–66.
15. Kim, M. H., Heo, J. H., Wei, Y., & Lee, M. C. (2011). A path planning algorithm using artificial potential field based on probability map. In *8th International Conference on Ubiquitous Robots and Ambient Intelligence* (pp. 41–43).
16. Pradhan, S. K., Parhi, D. R., & Panda, A. K. (2009). Fuzzy logic techniques for navigation of several mobile robots. *Applied Soft Computing*, 9(1), 290–304.
17. Kamkar, I., Akbarzadeh-T, M. R., & Yaghoobi, M. (2010, October). Intelligent water drops a new optimization algorithm for solving the vehicle routing problem. In *2010 IEEE International Conference on Systems Man and Cybernetics (SMC)* (pp. 4142–4146). IEEE.

18. Elhoseny, M., Shehab, A., & Yuan, X. (2017). Optimizing robot path in dynamic environments using Genetic Algorithm and Bezier Curve. *Journal of Intelligent & Fuzzy Systems*, 33(4), 2305–2316. <https://doi.org/10.3233/jifs-17348>.
19. Dadgar, M., Jafari, S., & Hamzeh, A. (2016). A PSO-based multi-robot cooperation method for target searching in unknown environments. *Neurocomputing*, 177, 62–74.
20. Ayari, A., & Bouamama, S. (2017). A new multiple robot path planning algorithm: dynamic distributed particle swarm optimization. *Robotics and Biomimetics*, 4(1), 1–15.
21. Purcaru, C., Precup, R. E., Iercan, D., Fedorovici, L. O., David, R. C., & Dragan, F. (2013). Optimal robot path planning using gravitational search algorithm. *International Journal of Artificial Intelligence*, 10, 1–20.
22. Das, P. K., Behera, H. S., Jena, P. K., & Panigrahi, B. K. (2016). Multi-robot path planning in a dynamic environment using improved gravitational search algorithm. *Journal of Electrical Systems and Information Technology*, 3(2), 295–313.
23. Rao, D. C., Pani, S., Kabat, M. R., & Das, P. K. (2017, August). Cooperation of multi-robots for obstacle avoidance in clutter environment using differential evolutionary algorithm. In *2017 International Conference on Information, Communication, Instrumentation and Control (ICICIC)* (pp. 1–6). IEEE.
24. Das, P. K., Behera, H. S., Jena, P. K., & Panigrahi, B. K. (2017). An intelligent multi-robot path planning in a dynamic environment using improved gravitational search algorithm. *International Journal of Automation and Computing*, 1–13.
25. Abaee Shoushtary, M., Hoseini Nasab, H., & Fakhrzad, M. B. (2014). Team robot motion planning in dynamics environments using a new hybrid algorithm (honey bee mating optimization-tabu list). *Chinese Journal of Engineering*, 2014.
26. Rakshit, P., Konar, A., Bhowmik, P., Goswami, I., Das, S., Jain, L. C., et al. (2013). Realization of an adaptive memetic algorithm using differential evolution and q-learning: A case study in multirobot path planning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(4), 814–831.
27. Rohmer, E., Singh, S. P., & Freese, M. (2013, November). V-REP: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 1321–1326). IEEE.
28. Adept Technology, Inc Mobile robots, “Pioneer p3-dx specifications.” www.mobilerobots.com/Libraries/Downloads/Pioneer3DXP3DX-RevA.sflb.ashx. Accessed May 28, 2018.

Customer's Activity Recognition in Smart Retail Environment Using AltBeacon



M. Lakshmi, Alolika Panja, Naini and Shakti Mishra

Abstract SRMS (Smart Retail Management System) is a project based on IoT (Internet of Things), which is an upcoming technology that deserves the attention of the industry. IoT provides unique identifiers to objects and people and transfers data over a network without any interaction of human to human or human to computer, for example tracking your activity level at real-time basis. In this project, we will be empowering retail stores with the power of IoT. In this project, we will be sending tailored offer schemes to the customer of the store whenever he/she is in the radius of a beacon of a particular shelf in store. This process, in turn, will help the customers to get specific offers for happier customer experience and will aid managers of the stores to better analyse the trends of the customers and create appropriate deals to attain higher profits with proper management of inventory.

Keywords Virtual beacons (AltBeacon) · Smart Retail Management System · Consumer behaviour analytics · Location tracking · Push notification · BLE (Bluetooth Low Energy) devices

1 Introduction

India is progressing towards a Digital India with the regularly increasing technically sound customers, there should be an effort from retail industry to move along with their customers and come with solutions that not only save their time but also make their shopping experience quite user-friendly.

M. Lakshmi (✉) · A. Panja · Naini · S. Mishra
Department of Information Science and Engineering, Nitte Meenakshi Institute of Technology,
Bengaluru 560064, India
e-mail: lakshmi.m@nmit.ac.in

A. Panja
e-mail: alolikapanja95@gmail.com

Naini
e-mail: naini.pandey18@gmail.com

S. Mishra
e-mail: shakti.mishra@nmit.ac.in

There are some serious challenges being faced by customers in the retail industry [1], some of which are as follows:

1. Struggling along with the economy: With the increase in the cost of commodities, it is difficult for the retailers to communicate the high cost of the products to the customers.
2. To overcome this challenge the retailers should come up with more deals and offers to showcase for their customer's benefit.
3. Staging stores: With the increasing demand of online shopping brick and mortar stores need to make their place more attractive so that customers would want to visit the stores and spend more time just as they would in destinations like Starbucks, CCD, etc.

With the SRMS, we tried to overcome this challenge by taking their feedback into consideration and adding more features to our app that would help the customers to have an enjoyable experience while they shop.

4. Analysing data: Retailers do not have much information about their customers which makes analysing their behaviour a challenging task.

In the SRMS, the preferences of every individual customer are taken into account during the registration procedure, which will help them to find more deals and offers which is specifically relevant to them only. The system also does real-time analysis of their location to provide the offers present in that particular section of the store.

5. Figuring out the potential of mobile devices: Shoppers browse similar products online while they are shopping at the stores to compare the prices of similar products. Hence the retailers must find a way to determine how to best take the advantage of this technology to attract shoppers.

There is a dedicated smartphone which belongs to each and every individual which they can use to enhance their shopping experience using the SRMS to get personalized notifications.

6. Standing in long queues to complete the payment procedure: Shoppers usually need to stand in long queues to complete their payment procedure.

With the SRMS the customers can directly scan the products before adding it to the cart and complete the payment procedure digitally using their debit/credit cards.

Our main aim was to create an IOT-based retail environment for an improved user retailer interaction for enhancing the marketing techniques and customers experience. SRMS is a refreshing wave that can come as a saviour for this industry by not only enhancing the user experience of the customers but also in attaining high gains for the industry by proper analysis of its data. SRMS uses beacon technology which is a low-energy device. SRMS is based on this process as it uses this connection between the customer's mobile device and beacons located on different shelves of the store. The data collected in this process, in turn, will help the managers to get an idea of customer trends, popularity of products, and other store dynamics. Implementation of this system can reduce the losses to stores, mismanagement in stores, and better

handling of inventory of stores. This can be a big step toward creating a Digital India and also boosting other initiatives like Make in India, etc.

In this paper, we have discussed the design, working and function of Smart Retail Management System and Conclusion. The rest of this paper is outlined as follows: Sect. 2 gives the literature survey, Sect. 3 presents the proposed methodology and the algorithm used to construct the SRMS, Sect. 4 explains the implementation and the data flow of the system with code snippets and Sect. 5 gives the conclusion and outlines of future work.

2 Related Work

2.1 Customer Behaviour Analysis

Customer behaviour analysis includes that it identifies the customer and their buying behaviour patterns.

In the retail industry, the analysis of this information is very important and useful as it helps to find solutions to a lot of marketing, problems and also makes the customer's shopping experience enjoyable. India has made a late entry into organized retail management techniques. There has been a lot of change over the past few years after adopting new ideas to track consumer behaviour.

In a recent survey carried out by Accenture, it was recorded that 72% of 258 North American business leaders said they should spend more on analytics. Amy Lin, a mother of two in suburban New York City said that it was easy to shop online rather than physically visiting the stores. The supermarket needs to come up with more attractive ideas to make people want to be there [1].

In 2014, Infinite Dial conducted a study which estimated that 61% among 160 million Americans, aged 12 or more, owned a smartphone [2]. It was predicted by Business Insider intelligence, which is a research service that an annual growth of 287% with 45 lakhs of beacons by 2019 out of which 35 lakhs of beacons is used in the retail environment. SWIRL is a mobile marketing firm. It reported that when 67% of the shoppers in retail industry received an in store alert; 81% of them opened the alert message and 79% made a purchase related to the message [3] (Fig. 1).

2.2 Beacon Technology

1. Bluetooth Low-Energy (BLE)

BLE or Bluetooth Low Energy is a wireless network technology which is used in many applications such as retail, health care, home entertainment, and security-based industries. When compared to Bluetooth technology, BLE provides a similar communication range with lower power consumption and minimal cost.

According to the survey carried out by SWIRL, beacon marketing campaigns are influencing shoppers behaviour: 73% of shoppers said that use of beacons in retail management increased their likelihood to purchase during their store visit. 61% said they prefer shopping at stores that delivered mobile content and offers while they shop. 61% said they would visit a store with beacon marketing campaign more often. 60% said they would buy more as a result of receiving beacon-triggered marketing messages.

Fig. 1 Case study carried out by SWIRL

Table 1 Comparison between Low energy and Bluetooth

S. No.	Technical specification	Bluetooth technology	Bluetooth Low-Energy technology
1	Distance/range	100 m	>100 m
2	Active slave	7	Not defined, implementation dependent
3	Minimum total time to send data	100 ms	3 ms
4	Power consumption	1 W as the reference	0.01–0.50 W (depending on use case)
5	Application throughput	0.7–2.1 Mbit/s	0.27 Mbit/s
6	Network topology	Point-to-point/scatternet	Point-to-point/star
7.	Primary use cases	Mobile phones, PCs, headsets, automotive	Mobile phones, gaming, fitness, medical, electronics and automotive

Table 1 depicts a comparison between Bluetooth Low Energy and Bluetooth devices [4].

2. BLE devices and Beacon

A BLE device communicates via its service and characteristics. A service specified for BLE device may have one or more characteristics. Each service and characteristics are represented by a Universal Unique ID (UUID). A Beacon device is broadcasting BLE device. It is a new technology with an indoor positioning system or local positioning system. This technology is based on Bluetooth low energy (BLE). At the hardware level, BLE devices are broadcasting data and on the other side, at the software level, beacons are messages sent by broadcasting devices that are then detected and processed by receiver devices. UUID serves as beacon identifier, while transmitting the data from beacon, which is one-way discovery mechanism, i.e., beacons devices are not aware of phones. Beacon has

the benefit to consume less power and is low cost when compared to other indoor positioning systems [5].

Hardware perspective of beacon

Beacons are small devices which are as small as a palm-sized object attached to the walls. These devices have tiny Bluetooth radio transmitter which continuously broadcasts radio signals. The signals are transmitted at regular intervals is a combination of letters and numbers.

Beacons mainly consist of three components: CPU, Radio, and Batteries. These devices use small lithium chip batteries or can be connected via powers like USB plugs. Beacons are available in different shapes and colours and may include additional features like temperature sensors, accelerometers, etc.

Software prospective of beacon

Beacons transmit a unique id number that communicates listening devices like smartphones to which beacon it is adjacent to. The concept of beacons is often misconceived. They are used for broadcasting a signal and not for tracking the objects.

Beacon software specifications [5] are as follows:

- Durability—Most beacon has a battery life of more or less 1.5–2 years. Some might range from 6 to 7 months. Beacon with can last over 60 months if they have energy-saving capabilities since Bluetooth is incredibly energy efficient.
- Interval—It depends on a specific scenario. It can be set according to ones need (ms-millisecond).
- TxPower—The transmission power of a beacon can range from 13 ft 1.4 inches to 164–295 ft depending on the specific use.
- Price—Beacons can cost as cheap as \$5 but the average cost of beacon may range from \$15 to \$25.
- Devices are not connected to beacons, they just receive radio waves broadcasted by a beacon and calculate distance (by RSSI). So, there is no limit.
- The maximum range of a beacon is around 70 m.

3. *Different types of beacons*

There are various kind of beacons available in the market:

- iBeacons (Apple)
- Eddystone (Google)
- AltBeacons (Radius Network).

The comparative study [6] is given in Table 2.

Table 2 Comparative study of different beacon devices

S. No.	Parameters	iBeacon	Eddystone	AltBeacons
1	Technology	In December 2013, Apple announced iBeacon. Built into Apple’s iOS 7 and later versions of mobile operating system that allows iPhones and iPads to constantly scan for Bluetooth devices nearby. Beacons use BLE which is part of Bluetooth 4.0 specification	In July 2015 Google announced Eddystone. Google’s Eddystone is a beacon protocol for open source beacons, which can be manufactured by any business at affordable costs	In July 2014 Radius Networks announced AltBeacon. It is an open source designed to overcome the issue of protocols favouring one vendor over the other
2	Compatibility	Android and iOS compatible but native only to iOS	It is cross platform and thus is compatible with any platform that supports BLE beacons	Compatible with other mobile operating platforms
3	Profile	It is proprietary software thus the specification is controlled by Apple	It is an open source. The specification is published openly on GitHub	It is an open source. The specification is published openly on GitHub
4	Ease of use	It is simple to implement	It is flexible but requires more complicated coding when it comes to integration	It is more flexible with customized source codes and has the ability to have different Manufacturer IDs and different Beacon codes
5	Broadcasted packets	Each beacon broadcasts information which is identified as a packet which has a unique id number comprising of three parts—UUID, major and minor	Eddystone broadcasts three different packets—(i) a unique id number (Eddystone uid), (ii) a URL address (Eddystone URL), (iii) sensor telemetry(Eddystone TLM)	Each beacons broadcast information with four parameters—UUID, major, minor and Tx power

2.3 *Benefits of the Retailers*

Using beacon devices in the retail shop enhances the experience of the user. The benefits of retailers are as follows [7]:

- Beacon technology can pinpoint location more accurately than other top contenders GPS, WIFI and NFC.
- It helps the retailers in blending the digital and physical realms.
- It also helps the retailers in directing customers to their purchase locations.
- It helps in maintaining consumer attention within the retail environment for a long time than usual.
- It enables retailers to create more tailored experiences for customers, thereby developing a good relationship with customers.
- It acts as an advertising tool which will enhance their marketing techniques.

2.4 *AltBeacons*

AltBeacon is a protocol specification that defines a message format for the beacon in the same vicinity. The messages are transmitted by devices for the purpose of signalling their proximity to nearby receivers. The contents of the transmitted message contain information that the receiving device can use to identify the beacon and to compute its relative distance to the beacon. The receiving device may use this distance as a condition to execute services and implement actions that are relevant to being in the vicinity to the transmitting beacon [8].

- AltBeacon is easier to use as it has greater transparency than other beacons to what a beacon transmits and how the data obtained can be used by Windows, Android, and other operating systems.
- It helps the customers as well as retailers to shift towards a shared platform for non-iOS devices.
- In the future, AltBeacons might empower developers to take true advantage of what a BLE beacon can do.
- AltBeacons can confidently compete against other beacons to increase the use of beacon adoption for both iOS as well as Android devices.

3 **Proposed Parallel Computing Frameworks**

3.1 *Methodology*

Figure 2 depicts how communication is established between the different components of SRMS. Each beacon represents different sections of the stores and has a unique

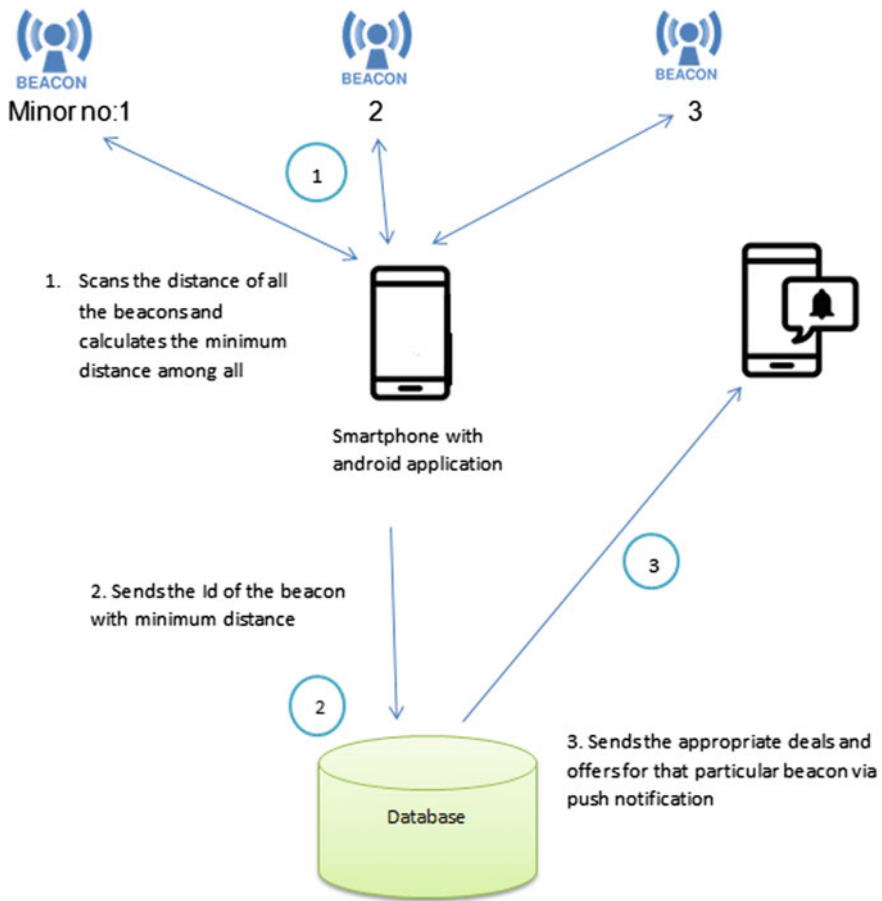
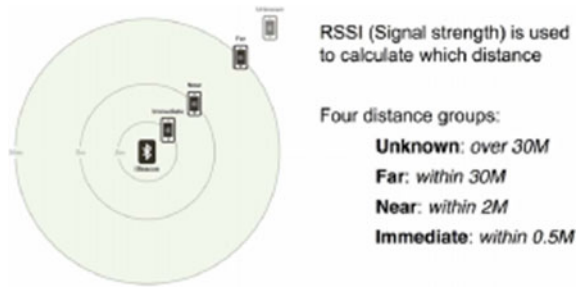


Fig. 2 Working of SRMS

minor number which helps us to identify a particular section. The first step in the (Fig. 2) depicts when the smartphone with the installed application enters the range the smartphone picks up the signal transmitted by the different beacons and finds the beacon closest to the smartphone. In the second step, the smartphone sends the beacon ID of the closest beacon to the database. In the third step, the database returns the deals and offers associated with that particular beacon ID. The deals and offers then appear on the smartphone as push notifications.

Fig. 3 Distance calculation



3.2 Algorithm

The proposed algorithm for implementing this methodology has been represented in Fig. 3. The algorithm works as follows:

- Step 1: Initialize the system by considering becond id = 0.
- Step 2: Compute the distance between all the beacons.
- Step 3: The minimum distance is set to highest.
- Step 4: Identify the minimal distance among all the distances. This beacon serves as a current beacon.
- Step 5: If the current beacon has not been used prior, activate the current beacon. Start pushing the notification.
- Step 6: If the beacon has already served in the past, enable all the push notification sent earlier.

3.3 Distance Calculation

Figure 3 displays the following four distance groups: Unknown, Far, Near and Immediate. These states define the distance of user from the beacon. In the figure, Unknown state signifies that distance cannot be calculated as the user is more than 30 m away from the beacon. Far state defines a range between 2 and 30 m. The near state ranges from 2 to 0.5 m. The distance of the object from the beacon can be easily calculated by this method. The last state is immediate, which is within the range of 0.5 m.

The distance calculation uses the strength of the signal from the beacon (Received Signal Strength Indication, or RSSI) and it gives the distance in metres. As the signal strength increased, the accuracy also increases (Algorithm 1).

```

Step1: Start.
Step2: Initialize System.
      BID(Beacon_id)='0';
Step3: Scan the distance of all the beacons using getDistance().
      AltBeacon.getDistance();
Step4: Sets minimum distance to the highest.
      minDist=Double.MAX_VALUE;
Step5: Calculates the minimum distance of all distances.
      for (allBeaconsInRange)
      if(current_beacon.getDistance())<minDist)
      minDis=current_beacon.getDistance

Step6: Checks if the current_beacon_id is not equal to the previous beacon_id.
      if(current_b_id !=BID)
Step7: If the condition is true it will activate the current beacon .
      isActive=false;
      activate(current_b_id)
Step8: If the condition is false it will show notification and sets isActive as true.
      showNotification();
      isActive=true;
Step9: Stop.

```

Algorithm 1 Beacon communication and push notification

4 Experimental Setup and Performance Analysis

4.1 Data Flow

A data flow diagram gives the diagrammatic representation of the flow of the data through the system. It is the first step in creating an overview of the system. The flow diagram (Fig. 4) shows the data flow of the application and the interaction between the different interfaces of the application.

In order to avail the benefits of the application, the user first needs to register and then login successfully that will bring them to the homepage. The homepage consists of a navigation drawer with the following options:

1. myAccount in which the user's details will be stored.
2. Scan and Buy which can be used by the user to scan the product directly via their smartphone. This will add the products into the cart where they can proceed with the online payment procedure.
3. Deals and Offers will showcase all the products available in the stores with discounts.
4. Logout option to log out of the application.

4.2 Code Snippets

Here are some of the code snippets, which explain certain functionalities of the application:

1. *Scanning and finding beacon with minimum distance*

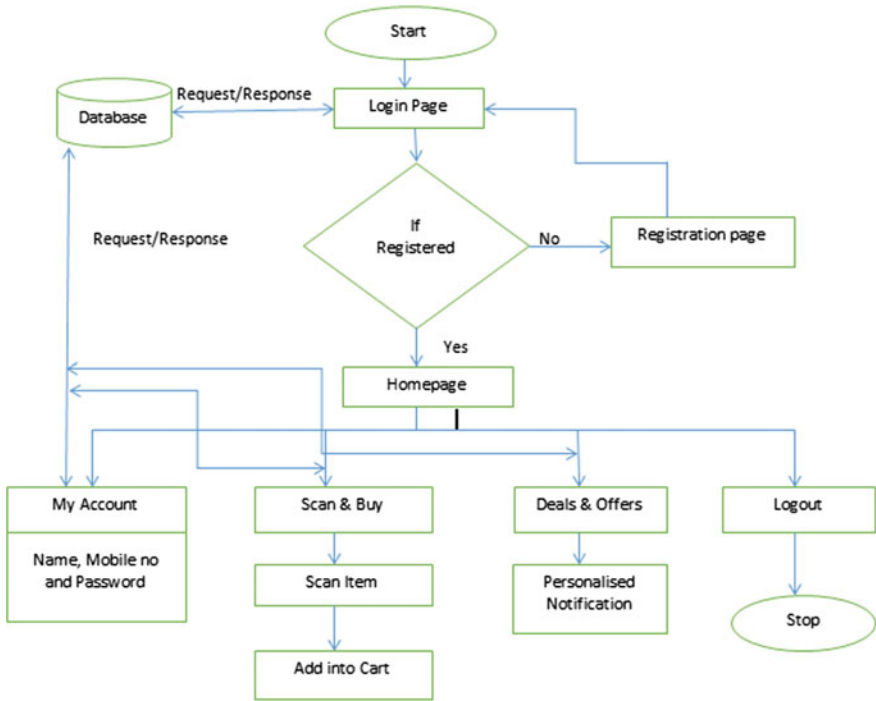


Fig. 4 Data flow of the application

```

public void onBeaconServiceConnect() {
    final Region region = new Region("myBeacons", null, null, null);
    beaconManager.setMonitorNotifier(new MonitorNotifier() {
        @Override
        public void didEnterRegion(Region region) {
            try {
                Log.d(BEACON_TAG, "didEnterRegion");
                beaconManager.startRangingBeaconsInRegion(region);
            } catch (RemoteException e) {
                e.printStackTrace();
            }
        }

        @Override
        public void didExitRegion(Region region) {
            try {
                Log.d(BEACON_TAG, "didExitRegion");
                beaconManager.stopRangingBeaconsInRegion(region);
            } catch (RemoteException e) {
                e.printStackTrace();
            }
        }

        @Override
        public void didDetermineStateForRegion(int i, Region region) {

        }
    });
}
    
```

2. *Send Notification*

```

void sendNotification(String newMsg) {
    NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(Home.this);
    mBuilder.setSmallIcon(R.mipmap.ic_launcher);
    mBuilder.setContentTitle("Shop Easy");
    mBuilder.setContentText(newMsg);
    mBuilder.setAutoCancel(true);
    mBuilder.setDefaults(-1);
    Intent notificationIntent = new Intent(this, Home.class);
    notificationIntent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    PendingIntent contentIntent = PendingIntent.getActivity(this, 0, notificationIntent,
        PendingIntent.FLAG_UPDATE_CURRENT);
    mBuilder.setContentIntent(contentIntent);
    NotificationManager manager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
    manager.notify(001, mBuilder.build());
}
}

```

3. *Scan and Buy*

```

@Override
protected void onResume() {
    super.onResume();
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        if (checkPermission()) {
            if (scannerView == null) {
                scannerView = new ZXingScannerView(this);
                setContentView(scannerView);
            }
            scannerView.setResultHandler(this);
            scannerView.startCamera();
        }
    }
}
}

```

4. *Result*

```

@Override
public void handleResult(Result result) {
    final String scanResult = result.getText();
    Get_Result conn = new Get_Result(this); //Get the product details related to bar_code of product
    conn.delegate = ScanAndBuy.this;
    String query = "select * from product where product_id = '' + scanResult + ''";
    conn.execute(URLS.Fetch_Product_URL, query);
}
}

```

The graph in Fig. 5 shows the power consumption of the Beacons with time it spends near the smartphone which contains the application. The more time it spends near the smartphone the more power it consumes and vice versa.

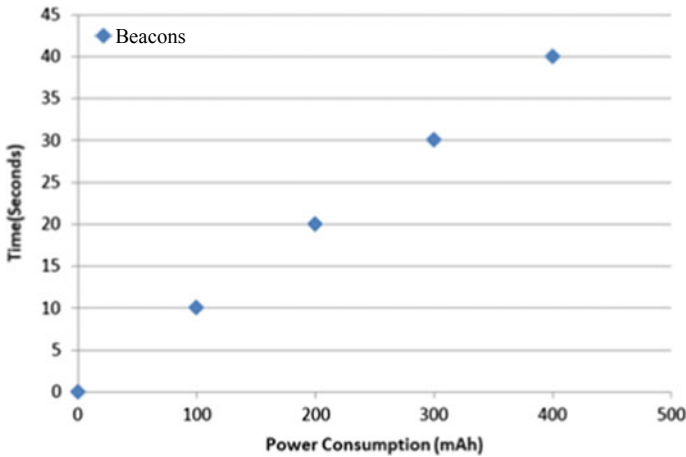


Fig. 5 Power consumption versus time graph

5 Conclusion

In this paper, we presented the design, working and the features of our Smart Retail Management System. Besides, this project successfully demonstrates a new role of Bluetooth low-energy device and the use of AltBeacons in the Retail Environment. SRMS will help the customers make the shopping experience more pleasurable and efficient by providing them with personalized notifications and an option to avoid long queues in the supermarket by enabling them to scan the product directly and make their instant digital payments.

Future Work

In the future, if all the studies are favourable we would like to implement the idea in a real-world scenario and add-on more features to attract more customers to the store, like keeping track of purchase history of individual customers which will give the retailers an idea about the likes and dislikes of the customers and accordingly provide personalized customer service. In future, we would also like to provide the customers with an indoor mapping system, which will guide them in and around the store. We will also try to implement this application on a hybrid platform to support smartphones with other operating systems.

References

1. Di, M. (2012, February 16). *Challenges facing today's retail industry*. Amecian Express Company.
2. Tim, H., & Tom, W. (2014, October 20). *The mobile commerce revolution and the current state of mobile*. Pearson Articles.

3. Moody, M. (2015, Spring). Analysis of promising beacon technology for consumers. *Elon Journal of Undergraduate Research in Communications*, 6(1).
4. Bluetooth Low Energy (Online). Available https://en.wikipedia.org/wiki/Bluetooth_Low_Energy#Hardware. Accessed March 21, 2018.
5. What is a beacon (Online). Available <https://kontakt.io/beacon-basics/what-is-a-beacon/>. Accessed March 22, 2018.
6. Mallik, N. *The key differences between iBeacon, Eddystone & AltBeacon* (Online). Available <https://www.quora.com/What-are-the-key-differences-between-iBeacon-Eddystone-AltBeacon>. Accessed March 20, 2018.
7. William, J. (2014, October). *What's the biggest influencer in consumer purchase decisions?* Small Business Trends.
8. Altbeacon Protocol Specification 1.0 (Online). Available <https://github.com/AltBeacon/spec>. Accessed April 9, 2018.
9. Zaim, D., & Bellafkih, M. (2016). Bluetooth Low Energy (BLE) based geomarketing system. In *2016 IEEE, INPT*, Madinat al irfane, Rabat, Morocco.
10. Kohne, M., & Sieck, J. (2014). Location-based services with iBeacon technology. In *2014 Second International Conference on Artificial Intelligence, Modelling and Simulation*.
11. Alt-Beacon (Online). Available www.altbeacon.org. Accessed March 21, 2018.

An Active Mixer Design For Down Conversion in 180 nm CMOS Technology for RFIC Applications



B. H. Shraddha and Nalini C. Iyer

Abstract This paper introduces a component of the radio frequency transceiver called the mixer. Mixers are found in almost all the communication systems at the front end. Radio frequency mixing is a key process within the RF technology and RF design. It is a nonlinear process that involves one signal level affecting the other signal levels at the output side instantaneously. The mixer design has the following design parameters Conversion gain, Linearity, Noise Figure, and port isolation. It is important to have better isolation between the ports as it is the measure of leakage or feedthrough from one port to another. Poor isolation leads to mixing of unwanted dripped signal with desired output signal creating inter-modulation products and adding distortion. The proposed Gilbert mixer is intended to produce IF frequency range of 1 MHz in UMC180 nm CMOS technology with a conversion gain of 8 dB, Noise figure of >10 dB, RF frequency range 5.001 GHz, reverse isolation >15 dB, and a stability factor of 1 at a low operating voltage of 1.8 V using a double-balanced topology. The mixer being designed provides a better isolation factor between the ports with less power dissipation of <10 mW.

Keywords Gilbert mixer · Conversion gain · Noise figure · Port isolation · UMC technology

1 Introduction

Radio frequency circuitry is gaining importance in recent years. Radio frequency communication has taken great advances from mobile phones to base stations; communication industry has been revolutionizing the way the entire globe transmits and receives information with growing demand. With this increase in the need and

B. H. Shraddha · N. C. Iyer (✉)
Department of ENC, BVBCET College of Engineering and Technology,
Hubballi 580031, India
e-mail: nalinic@bvb.edu

B. H. Shraddha
e-mail: shraddha_h@bvb.edu

© Springer Nature Singapore Pte Ltd. 2019
N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_50

605

demand of communication industry, there is ample scope of expanding and creating more reliable and efficient components. In order to realize the goals, there is a need for increasing the frequency range software and hardware design [1].

Mixers are an important component in any of the RFIC process applications. The processes in communication technology operate at least at ultra high frequencies and higher than that. Amplifying a high-frequency signal is not preferable as they induce a variety of parasitic into the design. The main purpose of making mixer an important component in the receiver front end is that it translates a signal from one frequency to another where the signal can be amplified or processed more effectively. Figure 1 shows the basic architecture of the mixer that is fundamentally a multiplier. The mixing unit has two input ports, i.e., radio frequency signal (RF) and local oscillator signal (LO) and one output port, i.e., intermediate frequency (IF) signal, respectively. The intermediate frequency signal contains two frequency components, one component depicts the down conversion of original signal and the other is up conversion compared to the frequency of the original signal. This work is intended to implement a down conversion mixer. Figure 1 makes use of low-pass filter which passes the down converted signal as an output.

The heterodyning process in the mixer is discussed in the following case, if both the input signals are:

$$a = A \cos(\omega_1)t \tag{1}$$

$$b = B \cos(\omega_2)t \tag{2}$$

The mixer output will yield

$$\begin{aligned} a \cdot b &= A \cos(\omega_1)t \cdot B \cos(\omega_2)t \\ &= \frac{AB}{2} \cos(\omega_1 - \omega_2)t + \frac{AB}{2} \cos(\omega_1 + \omega_2)t \end{aligned} \tag{3}$$

Equation 3 shows that the mixer output consists of the difference and summation of both input frequencies. Equations 1 and 2 are used to demonstrate the heterodyning process using alternating signals. The unwanted function can be filtered out. In the mixer design, there are four parameters that act as mixer specifications. They are conversion gain (CG), Linearity, Noise Figure (NF), and port isolation. The CG is

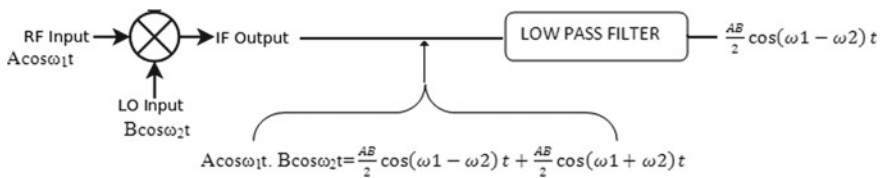


Fig. 1 Basic block of mixer

a ratio between the output signal and the input signal usually in the measure in decibels (dB) or milli-decibels (dBm). The linearity of the mixer is defined as how well the mixer reacts to the mixing of frequencies and ideal law of superposition in the ideal case explained in the above text. NF is a ratio of the signal-to-noise ratio (SNR) at the IF output and the SNR at the RF input port. Finally, the port isolation parameter shows how much leakage of signal occurs between two ports. Besides, the frequency of interest, there also exists the image frequency, interferers known as spurious response that has to be eliminated using filters during the system design [2].

Mixers are broadly classified into three types based on the number of differential ports. They are:

- Unbalanced mixer
- Single-balanced mixer
- Double-balanced mixer.

The design and implementation of unbalanced mixer cell include all three ports in a single-ended configuration. Single-balanced mixer cell has one port with differential configuration, whereas the other two are still in single-ended configuration [3]. The double-balanced mixer cells are implemented with all three ports in differential configuration. Table 1 shows the output components of three different types of mixers.

It is evident from Table 1 that the double-balanced mixer cell is the better choice as it does not provide any leakages from the input ports to output ports. There are multiple mixer designs and topologies that are well suited to meet the requirements of the system. The most used and popular mixer design is the Gilbert mixer cell design. Unlike the mixing circuits which make use of nonlinear components such as diodes, the Gilbert mixer cell uses linear time-invariant circuits to achieve the multiplication of time domain signals, and hence obtaining frequency shift in frequency domain. Gilbert cell mixer design provides balanced operation and there is a clearer signal expected at the output [4]. The work proposed by the author Yang et al. [5] on a low-voltage low-power and highly linear down conversion mixer operating at 900 MHz, though having a high linearity as an advantage this design provides a very low conversion gain of -6.5 dB. The down conversion mixer proposed by Faitah et al. [6] works with improved conversion gain but exhibits a higher noise figure. A down

Table 1 Output components of three types of mixers

	Unbalanced	Single balanced	Double balanced
Desired products	$\frac{1}{2} A \cos(w1)t.$ $sgn(\cos(w2)t)$	$A \cos(w1)t.$ $sgn(\cos(w2)t)$	$A \cos(w1)t.$ $sgn(\cos(w2)t)$
LO feedthrough	$\frac{1}{2} Vrf sgn(\cos(w2)t)$	$Vrf sgn(\cos(w2)t)$	-
RF feedthrough	$\frac{1}{2}(Vrf +$ $Asgn(\cos(w2)t)$	-	-

conversion mixer proposed by Caverly et al. [7] uses Gilbert cell configuration for wireless applications in 900 MHz band which has a very low conversion gain of 2.7 dB at 10 MHz intermediate frequency. The author Klumperink et al. [8] have proposed up conversion mixer using 0.18 μm technology with conversion gain of 10 dB but it has a drawback of providing a large noise figure of 24 dB. Darabi et al. (2005) proposed an up conversion mixer with low noise figure of 11.8 dB but the drawback is reduced conversion gain of 1 dB which is very low. The design of a direct conversion mixer included an improvement in the conversion gain of 6.6 dB but had a drawback of increased noise figure of 21 dB [9]. Hence, this paper promotes a design of balanced Gilbert mixer cell having a better performance in terms of conversion gain, noise figure, port-to-port isolation, linearity, and power consumption for various wireless applications. The brief of the paper is as follows, Sect. 2 discusses the proposed mixer design, comparison between the conventional Gilbert mixer cell and designed topology, Sect. 3 discusses the performance analysis and simulation results of the implemented mixer cell, and lastly, the conclusion is discussed in Sect. 4.

2 Proposed Mixer Design

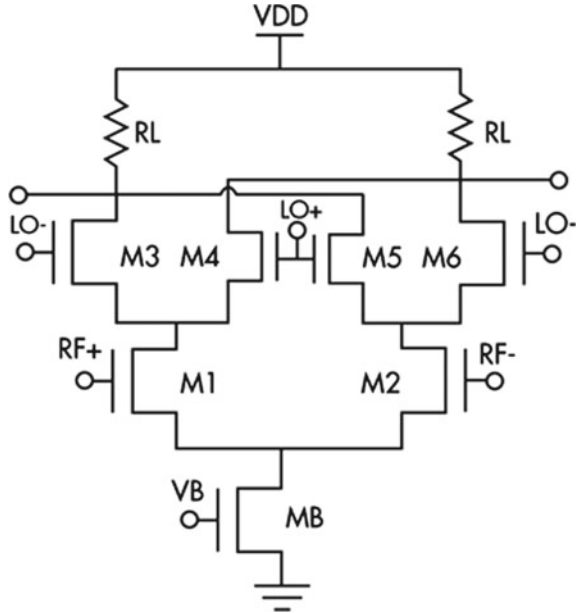
This section discusses designed mixer cell and its comparison with the conventional topology of Gilbert cell.

2.1 *Conventional Double-Balanced Gilbert Cell*

It consists of four unturned devices interconnected by multiple hybrids, transformers, or baluns. Figure 2 shows the usual topology of double-balanced Gilbert cell mixer [10, 11].

The mixer consists of two stages one stage is the switching stage and the other is the transconductance stage. In the double-balanced architecture, there are two transistors at the bottom referring to the transconductance stage and on the other hand, the switching stage has two pairs of transistors. The LO signal is preferably a square wave which switches the transistors M3, M4, M5, and M6 on or off depending on the magnitude of the LO signal represented as LO+ and LO– shown in Fig. 2. Each time when switching occurs, the current through the load moves through the switching transistors and multiplies with the LO signal and converts the current to the voltage. The output is differentially measured and all mixer parameters like gain are tabulated.

Fig. 2 Double-balanced Gilbert mixer cell



2.2 Mixer Design Parameters

Referring to Fig. 2 and solving for the output voltage, we get

$$V_{IF} = I_{IF} * R = \frac{4}{\pi} gm V_{RF} R [\cos(w_{RF} - w_{LO})t], \tag{4}$$

Hence, the conversion gain is given by

$$CG = \frac{2}{\pi} * gm * R \tag{5}$$

From Eq. 5, it is clear that the intermediate frequency output signal of a double-balanced Gilbert mixer cell contains frequency components other than LO frequency signal. Hence, this topology offers better isolation between the ports. The linearity in terms of IIP3 is given by

$$IIP3 = \sqrt{\frac{32}{3}} \frac{I_{SS}}{C_{ox} \frac{W1}{L1}} \tag{6}$$

The noise figure is given by

$$NF = \frac{\pi^2}{4} \left(1 + \frac{2\gamma}{gm \cdot Rs \cdot Rf} + \frac{2}{gm \cdot Rf \cdot Rs \cdot Rload} \right) \tag{7}$$

Port-to-port isolations are given by

$$LO_{IF}isolation = dBm(mix(HB . V_{out2}, (0, 1))) - LO_{Power} \quad (8)$$

$$RF_{IF}isolation = dBm(mix(HB . V_{out2}, (1, 0))) - RF_{Power} \quad (9)$$

$$RF_{LO}isolation = dBm(mix(HB . lo,0, (1, 0))) - RF_{Power} \quad (10)$$

Equation 4 is used to derive the gain of the mixer cell. Equations 8, 9, and 10 compute the dBm of the mixed outputs at desired frequencies and they are being subtracted from the respective input ports. Based on the above equations, the aspect ratios of the MOSFET's are designed. The mixer cell designed is aimed to meet the following specifications as shown in Table 2.

The tool used to design and implement the proposed mixer cell is CADENCE tool. This tool consists of managing libraries, schematic editing, symbol creation, test bench verification, analysis of simulation results, complete physical design and comparison of pre- and post-simulation results. Figure 3 shows the circuit topology of a conventional double-balanced Gilbert mixer cell. The circuit topology when simulated for the different parameters of the mixer cell, the values obtained were not able to meet up the set specifications. The circuit topology shown in Fig. 3 were modified to meet up the required specifications and hence degeneration resistors are added to the transconductance stage. The enhanced circuit topology of double-balanced Gilbert mixer cell is shown in Fig. 4.

Table 2 Mixer cell specifications

S. No.	Specifications	Value
1	Conversion gain	0 dB
2	Noise figure	Adequate
3	IP1 dB	Optimum
4	IIP3	Optimum
5	RF frequency range	5.0–5.001 GHz
6	IF frequency	1 MHz
7	Reverse isolation	>15 dB
8	Stability factor	1
9	Power consumption	<10 mW
10	Technology	UMC180
11	Supply voltage	1.8 V

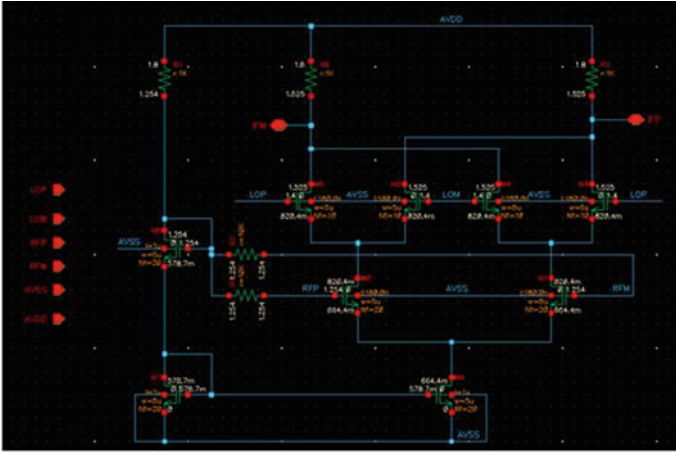


Fig. 3 Circuit topology

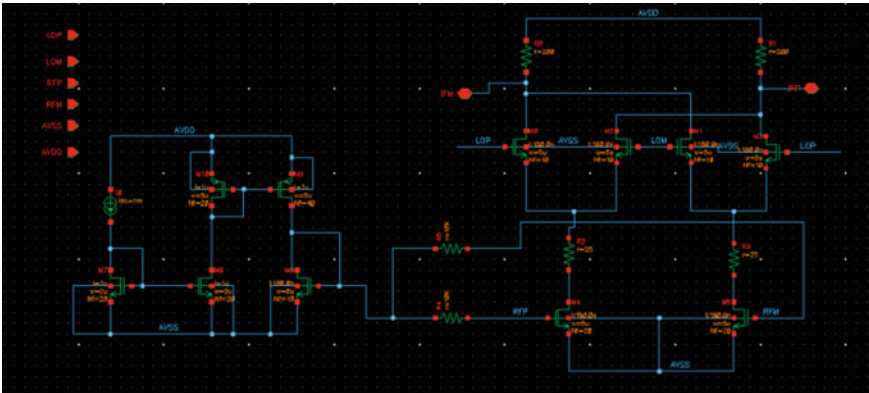


Fig. 4 Enhanced circuit topology

3 Performance Analysis and Simulation Results

The test bench for simulating different parameters is shown in Fig. 5. The proposed mixer design is simulated using UMC180 nm technology PDK files and results are validated on Cadence Virtuoso.

The gain of the mixer evaluates the mixer’s frequency conversion action. Hence it is termed as conversion gain. It is defined as the ratio of power delivered to the load to the power available at the input port. The conversion gain of the mixer was simulated by sweeping the signal amplitude of LO input and conducting the PAC and PSS analysis. The plot in Fig. 6 reveals that the maximum conversion gain offered by the mixer is 8 dB at the LO power of 8 all other parameters such as port isolation,

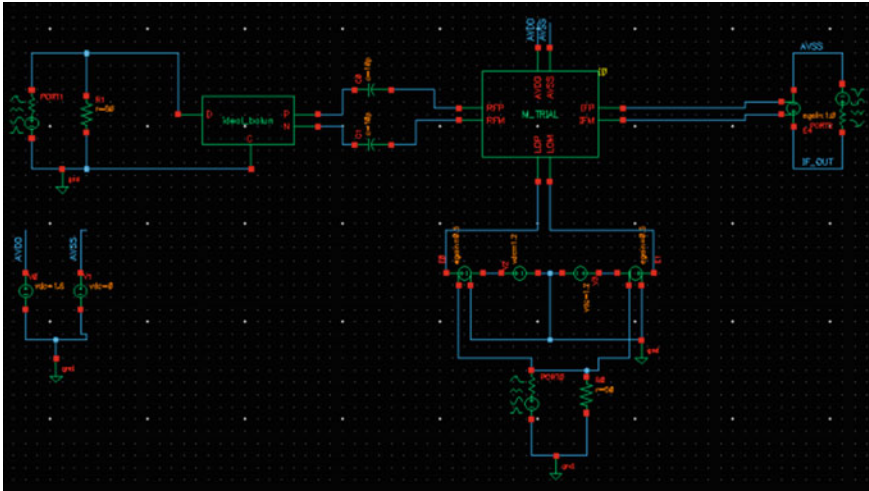


Fig. 5 Test bench of the enhanced double-balanced Gilbert mixer cell

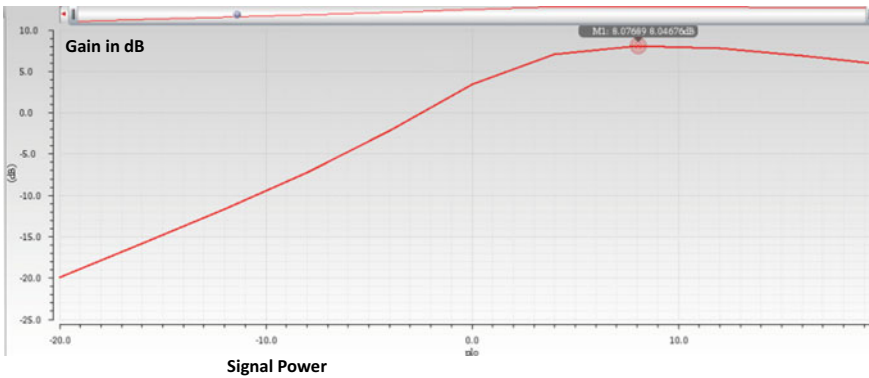


Fig. 6 Voltage conversion gain versus LO signal power

noise figure, and power consumption are not considered. The stability of the gain over the variation in the RF frequency is been simulated using PAC analysis. As seen in Fig. 7 though there is a variation in the intermediate frequency from 0 to 10 MHz, the gain is almost constant with a small variation of 8.05 dB at the higher side to 8.048 dB at the lower side. The voltage conversion gain versus RF frequency using PXF analysis has been simulated to check the variation in the gain of the mixer. Figure 8 reveals that though there is variation in RF frequency from 0 to 10 MHz the gain is constant with 8.054 dB at the higher side to 8.051 dB at the lower side.

As there are three different ports in the mixer cell, there exists feedthrough leakages between the ports. The different feedthroughs are measured in order to look into the isolation of the mixer. Figures 9, 10, 11, and 12 show the RF to LO, RF to IF, LO to

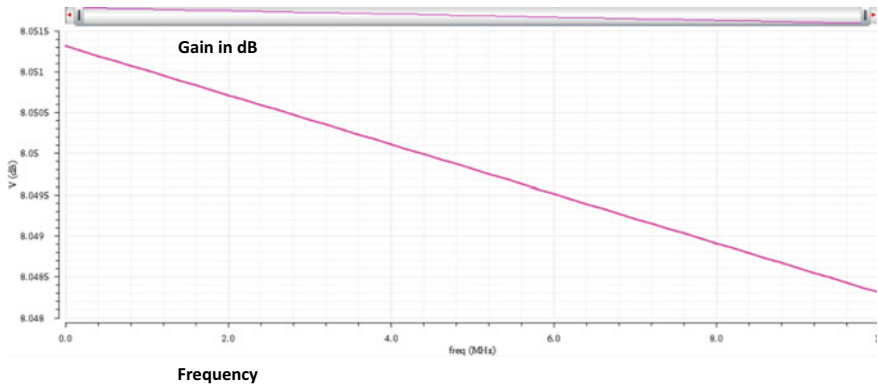


Fig. 7 Mixer gain versus RF frequency

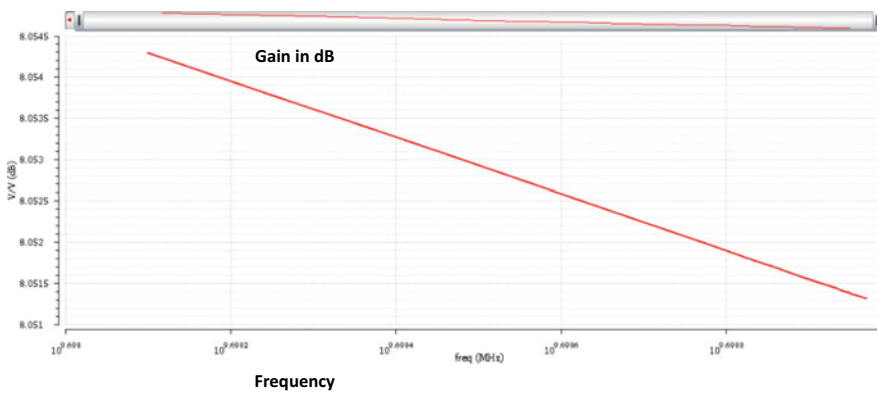


Fig. 8 Mixer gain versus RF frequency (PXF analysis)

Table 3 Simulated values for port isolations

Different port feedthrough	
RF to LO feedthrough	-6.4 kdB
RF to IF feedthrough	-81.7 dB
LO to IF feedthrough	-235.3 dB
LO to RF feedthrough	-81.7 dB

IF, and LO to RF feedthrough, respectively. The PAC and PXF analysis is combined together to simulate port isolations. Table 3 reveals the simulated results obtained for different feedthroughs. The conversion gain is kept constant while performing the isolation parameters.

Under small signal conditions the output signal power increases linearly with input signal power. When the circuit operates with large signals it is affected, and there are possibilities that the circuit operates in nonlinear region of operation. 1 dB point is a measure for this nonlinearity. To simulate and calculate 1 dB compression

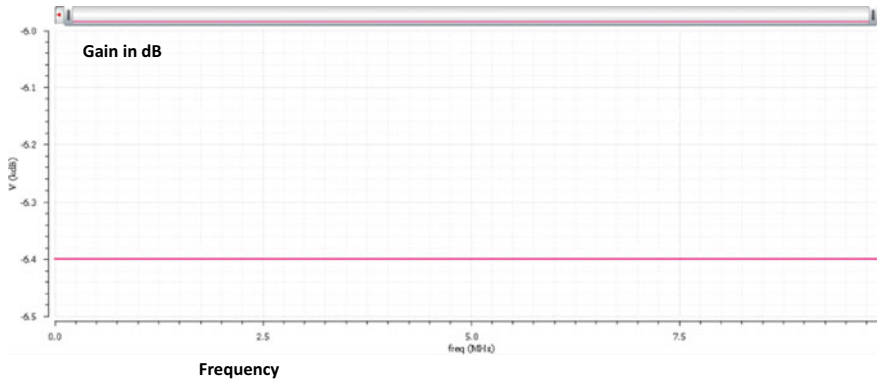


Fig. 9 RF to LO feedthrough

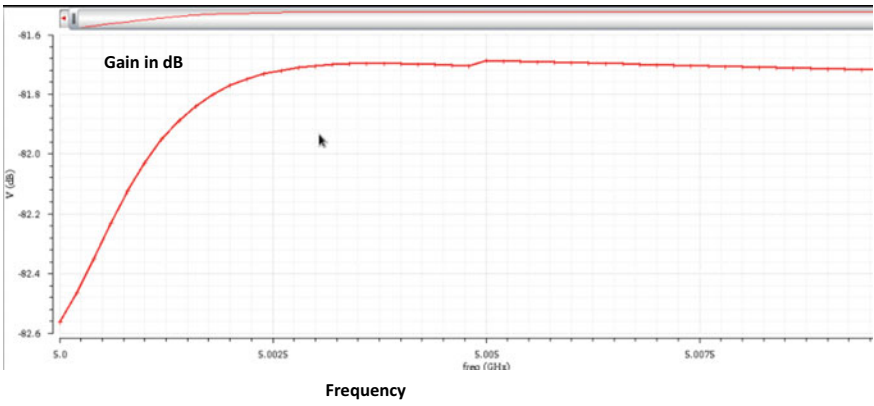


Fig. 10 RF to IF feedthrough

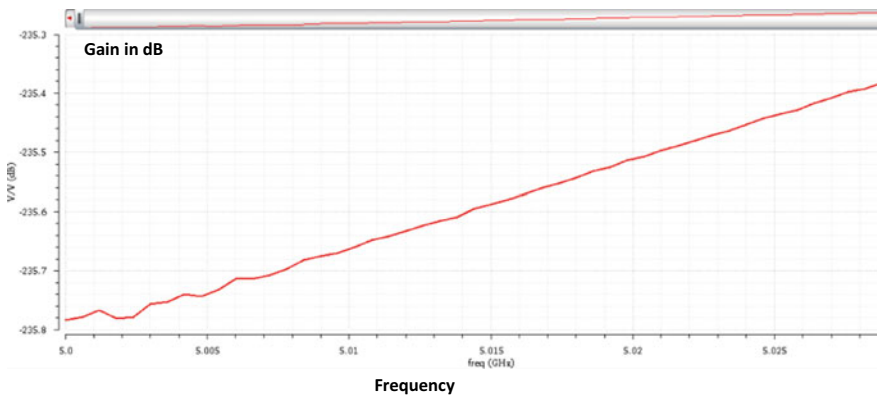


Fig. 11 LO to IF feedthrough

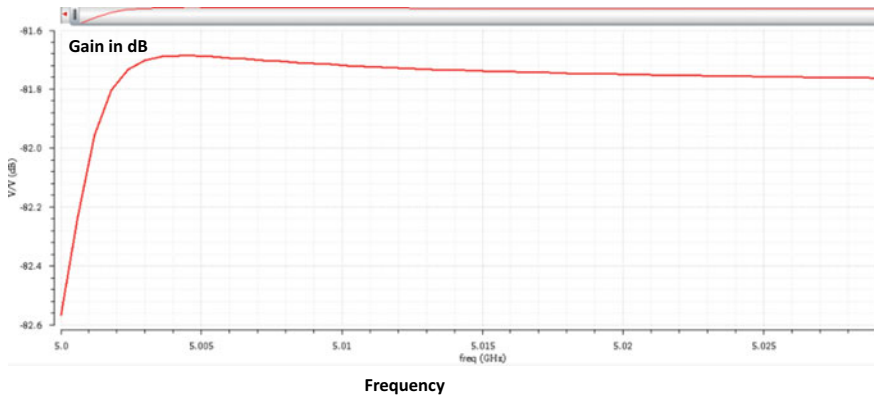


Fig. 12 LO to RF feedthrough

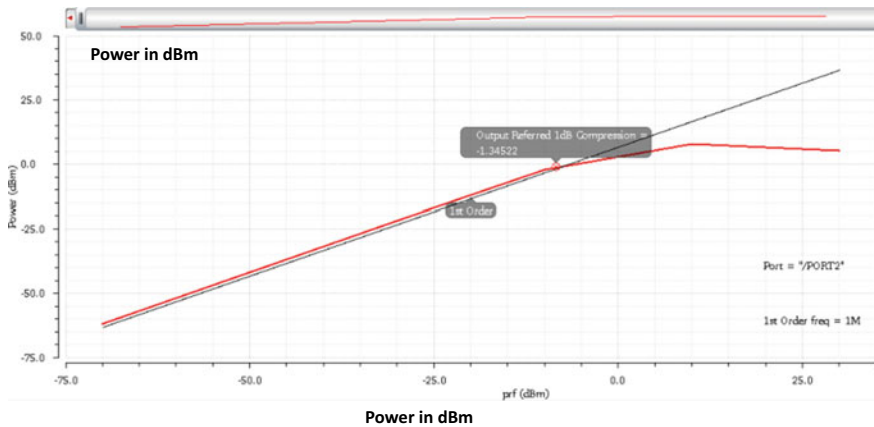


Fig. 13 Output referred IP3

point and IIP3 of the mixer QPAC and QPSS analysis is performed. Figures 12 and 13 show the results for 1 dB compression point and IIP3 calculated at first-order frequency, i.e., at 1 MHz and at third-order frequency, i.e., at 900 MHz. The input referred IP3 is 6.24 dBm and output referred is -1.34522 dBm (Fig. 14).

For the input signal frequencies of 5.001 and 5 GHz, an intermediate frequency of 1 MHz is been generated using transient analysis. Figures 15 and 16 show the IF output signal and its frequency spectrum.

From Fig. 16, it is evident that the first harmonic, i.e., the fundamental frequency is located at 1 MHz. After the fundamental frequency there are even and odd harmonics of the fundamental frequency which should be located at least 30 dB away from the fundamental frequency for better optimum functioning. In the implemented mixer the difference between first and second harmonic signal is 66 dB and the difference between first and third harmonic is 41.2 dB. The proposed mixer is successfully able

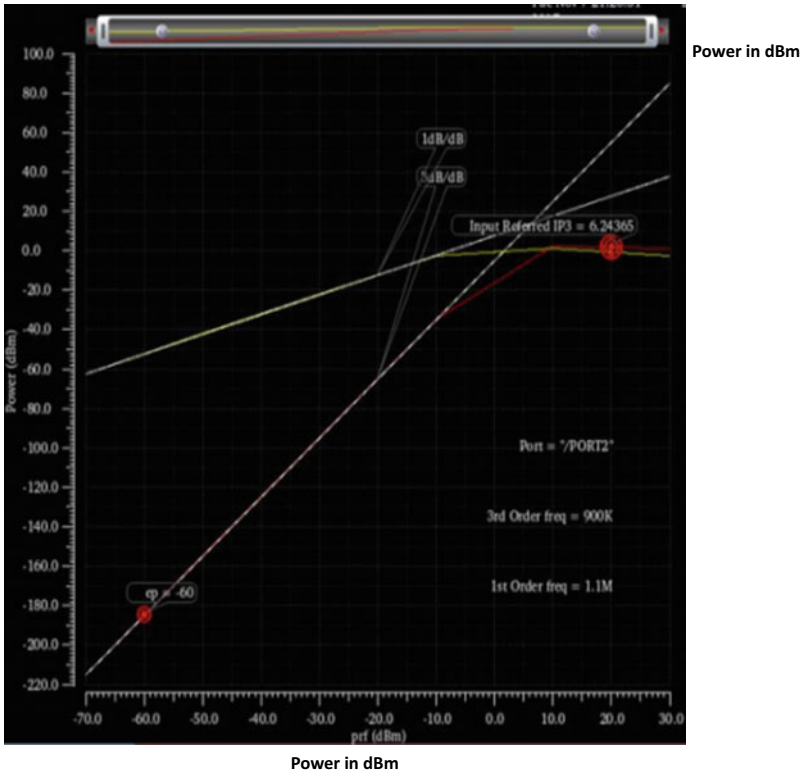


Fig. 14 Input referred IP3

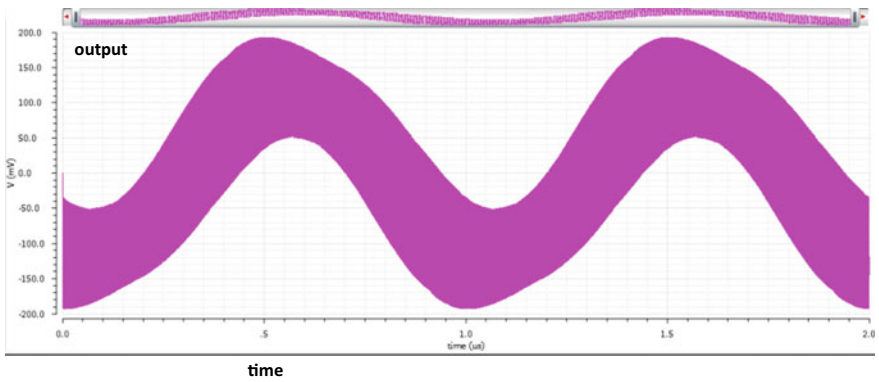


Fig. 15 Transient response of the IF signal

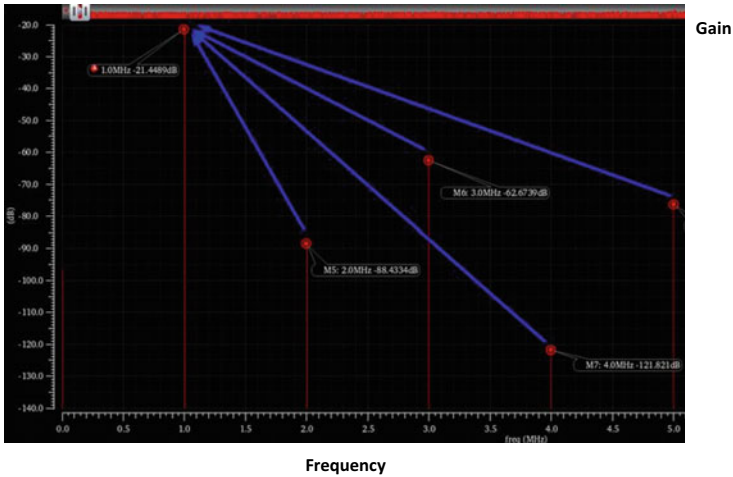


Fig. 16 IF spectrum frequency domain

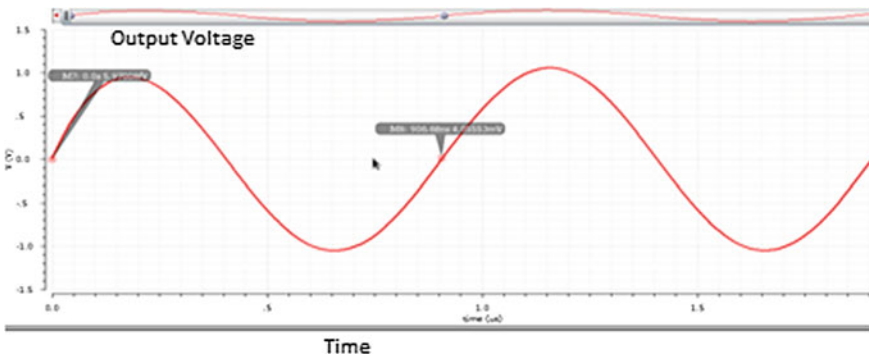


Fig. 17 IF output signal after filtering

to convert a 5.001 GHz RF frequency to a 1 MHz intermediate frequency. From Fig. 15, it is evident that the IF output signal contains a lot of harmonics, hence, a low-pass filter (LPF) is used at the output of the mixer designed for a cutoff frequency of 1 MHz using VCVS. The output of the LPF is clean 1 MHz sinusoidal waveform shown in Fig. 17, Transient analysis.

The implemented architecture of the mixer is tested for the variations in the process corners and variation in temperature. There are 4 process corners in VLSI design most widely used are typical typical (TT), slow-fast (SF), and fast-slow (FS). Table 4 shows the parameter analysis for process corner variation and temperature variation. The temperature is varied from $-40\text{ }^{\circ}\text{C}$ to $+125\text{ }^{\circ}\text{C}$.

Table 4 Parameter analysis for process corners and temperature variations

Parameters	TT	FS	SF	Temperature variation (TT)
RF frequency	5.001 GHz	5.001 GHz	5.001 GHz	5.001 GHz
LO frequency	5 GHz	5 GHz	5 GHz	5 GHz
IF frequency	1 MHz	1 MHz	0.98 MHz	1.3 MHz
CMOS process	180 nm	180 nm	180 nm	180 nm
Supply voltage	1.8 V	1.8 V	1.8 V	1.8 V
Power consumption	8.53 mW	8.5 mW	8.52 mW	8.53 mW
LO power (dBm)	8	7.68	8.04	8
Conversion gain (dB)	8	7.89	8	8.24
1-dB $-IIP3$ (dBm)	-1.34522	-1.28	-1.31	-1.44522
Third order $-IIP3$ (dBm)	6.24365	6.64	5.94	6.64
RF-LO isolation (dB)	-6.4 k	-5.8 k	-5.4 k	-6.4 k
LO-RF isolation (dB)	-81.6	-81.5	-81.05	-81.6
LO-IF isolation (dB)	-235.35	-223.15	-219.3	-235.35
RF-IF isolation (dB)	-81.6	-80.6	-80.72	-81.6

4 Conclusion

An active mixer with double-balanced topology has been designed with specifications shown in Table 2 and simulated using UMC 180 nm technology. The variations of the parameters for different process corners and temperature variation are shown in Table 4. The proposed Gilbert mixer exhibits better performance in terms of parameters like adequate gain, higher isolation between the input and output ports, and less power consumption.

References

1. Su, Z., Dai, F. F., Wilamowski, B. M., Hamilton, M., Zhou, W., Wang, Y., & Fu, J. A 0.8 x 2013;3 GHz wideband folded down-converter with noise cancellation in 0.18 μ m SiGe technology. In *2015 IEEE Bipolar/BiCMOS Circuits and Technology Meeting—BCTM*.
2. Sharma, U. K., Chaturvedi, A., & Kumar, M. (2016). A high gain down-conversion mixer in 0.18 μ m CMOS technology for ultra wideband applications. In *2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN)*.
3. Bi Pham. Research report on "1.9 GHz Gilbert Mixer in 0.18 μ m CMOS for a cable tuner".
4. Ghayvat, H., Bandil, L., Mukhopadhyay, S. C., & Gupta, R. (2015). A 2.4 GHz CMOS Gilbert Mixer in 180 nm technology. In *2015 Fifth International Conference on Communication Systems and Network Technologies*.

5. Kim, H. -R. et al. (2003). A 900 MHz low voltage low power highly linear mixer for direct-conversion receivers. In *2003 10th IEEE International Conference on Electronics, Circuits and systems* (vol. 3).
6. El Oualkadi, A. et al. (2008). CMOS RF down-conversion mixer design for low-power wireless communications. *ACM Ubiquity*, 9(24).
7. Caverly, R. H., Smith, S., & Hu, J. (2000). RF CMOS cells for wireless applications. *Analog Integrated Circuits and Signal Processing*, 25, 5–15.
8. Klumperink, E., et al. (2004). *Upconversion mixer using 0.18 μm technology*.
9. Darabi, H., & Chiu, J. (2005). A noise cancellation technique in active RF-CMOS mixers. *IEEE Journal of Solid-State Circuits*, 40(12), 2628–2632.
10. Gibson, A., Jr. (2011). *Design and simulation of CMOS active mixers*. Research Thesis in the College of Engineering and computer science, Central Florida.
11. Shraddha, B. H., & Iyer, N. C. (2017). Study of double balanced Gilbert mixer cell. *International Journal of Research and Advanced Development (IJRAD)*, 1.

Analysis of PAPR for Performance QPSK and BPSK Modulation Techniques



K. Bhagyashree, S. Ramakrishna and Priyatam Kumar

Abstract In both wireless and wired communication environments, there is demand for larger data rates, which is continuously increasing day by day. Hence, OFDM systems have been developed for digital systems. These systems have many advantages over single-carrier transmission systems like resistance to selective fading. But these systems are characterized by large value of PAPR. Many methods have been discussed in the literature for the reduction in PAPR value. Although these methods provide the reduction, they affect the transmission power, data transmission rates, error rates, and complexity in the computational model. One of the simplest ways of measuring PAPR is CCDF. In this paper, CCDF curves are used to measure the amount of PAPR in OFDM systems and are analyzed.

Keywords CCDF—Complementary cumulative distributing function · OFDM—Orthogonal frequency division multiplexing · PAPR—Peak-to-average power ratio · Clipping · Subcarriers · MIMO—Multiple input multiple output

1 Introduction

The demand for 4G wireless communication systems has increased exceptionally in the field of multimedia transmission. With the upsurge in the number of users and due to the limited bandwidth, there is a necessity for more advanced modern communication techniques which give good spectral and bandwidth efficiency. A system which is immune to multipath fading is coined as multicarrier system. This kind of multicarrier transmission systems is highly reliable and facilitates high data transmission rates for large numbers of users. In multicarrier systems, the main highlight is band-

K. Bhagyashree · S. Ramakrishna · P. Kumar (✉)
Department of ECE, BVB College of Engineering, Hubli 580031, India
e-mail: priyatam@bvb.edu

K. Bhagyashree
e-mail: bhagyashree@bvb.edu

S. Ramakrishna
e-mail: ramakrishnasj444@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_51

width distributed among many subcarriers. On the contrary, single-carrier systems have single carrier occupying the whole bandwidth. Now, these humungous features of multicarrier systems together with OFDM features provide better performance efficiency.

Audio–video broadcasting and mobile multimedia communications are some of the applications which incorporate features of OFDM. Techniques like modulation and multiplexing both are encompassed in OFDM. So, it has many benefits which make OFDM a potential option for 4G communication systems. The OFDM is characterized by subdivision of the available bandwidth into many data streams of low data rate, and then transmitting on individual subcarriers. Using fast Fourier transforms (FFT/IFFT), modulation and demodulation are done. The modulation is done on each subcarrier after the symbol generation. Each subcarrier has a unique central frequency which is orthogonal to the frequency of other subcarriers. Here, orthogonality is maintained using IFFT. Guard bands eliminate interference between the symbols. A wideband signal is converted to a number of narrow band signals.

Although there are many advantages, OFDM systems suffer from many problems such as PAPR, synchronization, and Bit error rate. PAPR is the major among the problems presented in OFDM systems. Many techniques for the reduction in PAPR have already been presented.

2 Literature Review on PAPR Reduction

The literature on peak-to-average power ratio has been reviewed and presented in the following section.

Foomoolijareon and Fernando proposed a solution to the problem caused by peak-to-average power ratio in Orthogonal Frequency Division Multiplexing system in the year 2002. Two methods were suggested [1]. The first method, wherein a list of input vectors are maintained and an appropriate input are selected from the list. The method is done by initially choosing the subcarriers and reducing the number in the input side and the passing this to the Inverse Fourier Transform. Both the methods are supported by simulation outputs showing considerable reduction in the peak-to-average ratio. One important observation here is this was done for restricted channel numbers.

Another work on the same lines was done by Xiadong et al. The work was proposed in the year 1998 which incorporated different techniques to reduce the power ratio in the communication system [2]. Methods like filtering and clipping were employed and spectral power density, bit error rate, etc., were the performance indicators considered. Simulations indicate better results when compared to traditional communication systems.

A reduction algorithm was proposed in the year 2011 by Wei Xeufeng. The new algorithm proposes a new method by incorporating advantages of the conventional method. This is a probability-based algorithm. The simulations show that the peak-to-average power ratio reduces considerably.

2.1 Need for Orthogonal Frequency Division Multiplexing

In many of the wired and wireless transmission systems, today, Orthogonal Frequency Division Multiplexing is like a boon. Figure 1 depicts the block level description of a typical OFDM transmission system with the Rx and Tx blocks. Symbol interference is caused because of high data transmission rates [3]. A guard band interleaving can be done in two ways, one by padding zeros, which is done by including zeros in between the symbols. The other is cyclic prefix; here, the last section of the symbol is copied at the first section of the next symbol. The size of the guard band should be such that it takes into account the response time of the channel to avoid interference [2, 4]. In case of Fast Fourier Transform, cyclic prefix method is favorable than padding zeros because this method periodizes the signal.

2.2 Effect of Peak-to-Average Power Ratio in Case of Orthogonal Frequency Division Multiplexing Systems

Basically in Orthogonal Frequency Division multiplexing, a data stream of high data rate is partitioned into data streams of lower rate [5, 6]. These lower data rate streams are then transmitted at once using many subcarriers, which may overlap with one another. At lower data rates the symbol duration increases causing dispersion in time. The main characteristic feature of OFDM systems is large number of subcarriers. And, high PAPR which offers problem in real-time transmission on the optical fiber

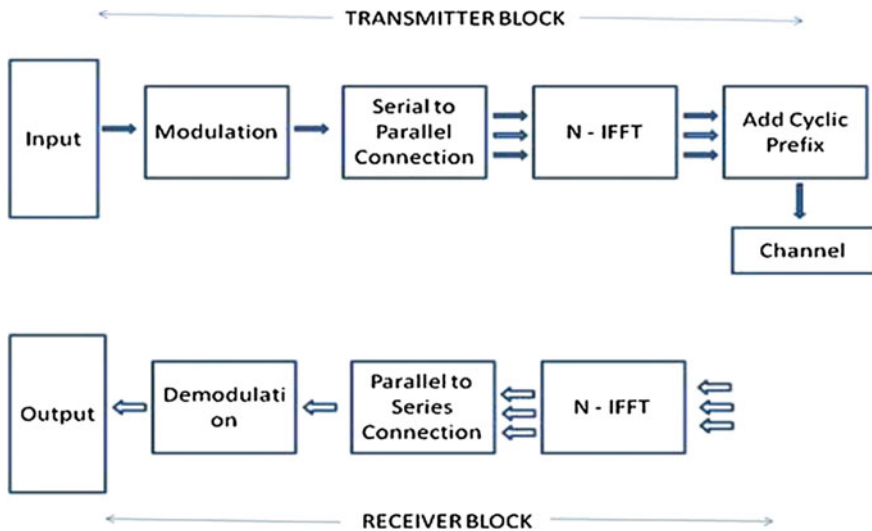


Fig. 1 Basic architecture diagram of the communication system

cables which lead to distortions in the communication bands. This effect elevates with the use of data converters in the system design. This leads to reduction in the performance of the circuit. The main cause for PAPR in the transmission is the presence of the large number of subcarrier which is not in phase. When these signals, which are not in phase, shoot up to maximum value simultaneously the output also raises this causes peak in the output value. In orthogonal communication systems due to many subcarriers large peak is present as against the average value. These demands for sophisticated transmitters which further increase the cost of the entire system.

3 PAPR

OFDM systems are characterized by the presence of many subcarriers in the system. Hence, the peak value is very large than the average value. This ratio is called peak-to-average power ratio.

$$PAPR = \frac{\text{maximum}|y(t)|^2}{E[|y(t)|^2]} \tag{1}$$

Here, numerator gives the peak signal power and denominator gives average power of signal. Let us consider the total number of subcarriers are N , and if all are in the same phase, then the N is the PAPR. Figure 2 shows envelope peaks for the OFDM information signal for N equals 16 which in turn leads to very large peak power [4].

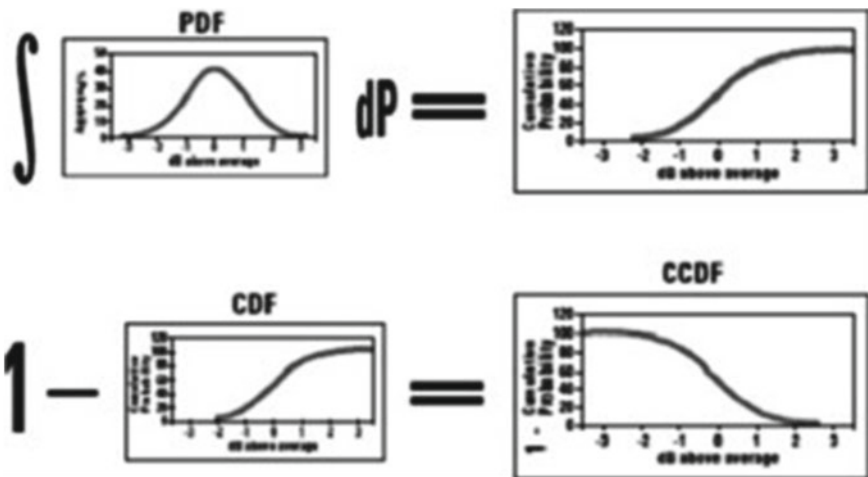


Fig. 2 Relation between PDF, CDF, and CCDF

3.1 PAPR Measurement

To measure and evaluate the performance of PAPR reduction scheme, complementary cumulative disturbing function (CCDF) is a standard candidate. If CDF and CCDF are compared, then it is revealed that what value of the information signal is below a required level is indicated by CDF and what amount of signal is at a given level and above is depicted by CCDF. The probability is defined by the percentage of time the data signal remains at a level or above [7].

The real and imaginary sections of signal in OFDM with many number of subcarriers is considered to be Gaussian. Under such situations, the signal envelope in the case of OFDM has Rayleigh distribution and power is exponential. If one assumes unity power for OFDM signal, then normalized Rayleigh distribution is PDF, that is, Probability Distribution Function [8].

4 Proposed Methodolgy

In the proposed method, a lower order modulation scheme has been incorporated like QPSK as it provides robustness against noise. But it is noted that it provides lower data rates. Here, MATLAB is used for simulation. The following are the steps followed in the method:

- i. Select the input size, subcarrier number and symbol number
- ii. Generate modulated signals
- iii. Assign the symbols modulated to the subcarriers
- iv. Compute inverse FFT
- v. Calculate PAPR
- vi. Plot PAPR versus CCDF graph.

5 Results

In the results, the graphs of PAPR values for different subcarrier numbers are shown. The modulation techniques used here are QPSK and BPSK.

The peak value of the envelope would be directly affected as the subcarrier number is increased, and this is due to large PAPR value. Figure 3, shows the simulated results of CCDF curves in case of an OFDM system with different number of subcarriers. Here, the 16-QAM modulation is used. As there an incremental change in the number of subcarriers, the PAPR value also raises. Major values are in the range 7–10 dB at CCDF 10⁻².

Figure 4 depicts for BPSK the values of PAPR are lower than the other modulation schemes. For any value of M in M—Quadrature amplitude modulation, the value of PAPR is almost the same.

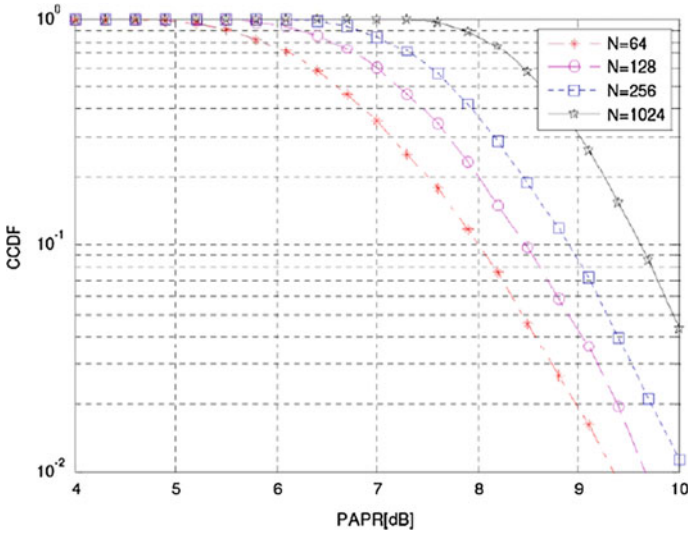


Fig. 3 Plot of CCDF for different values of N

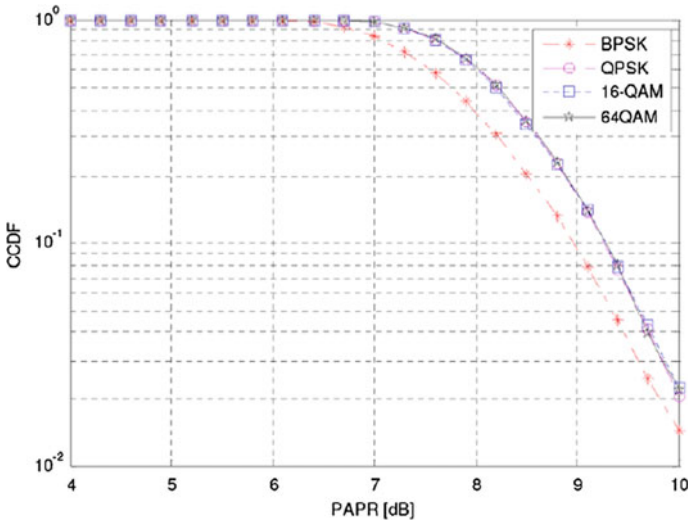


Fig. 4 Plot of CCDF for modulation using BPSK, QPSK in OFDM ($M = 2, 4, 6$ and $N = 512$)

6 Conclusion

In this paper, a method for obtaining the PAPR in transmission systems has been proposed. There are many ways to estimate the PAPR; one of them is based on the probabilistic approach. The reduction of PAPR is the primary issue because of the cost and complexity of different components in OFDM Tx and Rx is affected. The most basic parameter which evaluates the performance of PAPR reduction scheme is CCDF. The PAPR values for a 16-QAM with varying subcarriers range 7–10 dB at CCDF 10⁻².

References

1. Wang, Z. P., Xiao, J. N., Li, F., & Chen, L. (2011). Hadamard precoding for PAPR reduction in optical direct detection OFDM systems. *Optoelectronics Letters*, 7(5), 363–366.
2. Lee, B. M., Figueiredo, R. J. P., & Kim, Y. (2012). A computationally efficient tree-PTS technique for PAPR reduction of OFDM signals. *Wireless Personal Communications*, 62(2), 431–442.
3. Wu, X., Wang, J., Mao, Z., & Zhang, J. (2010). Conjugate interleaved partitioning PTS Scheme for PAPR reduction of OFDM signals. *Circuits; Systems and Signal Processing*, 29(3), 499–514.
4. Hassan, E. S., Xu, Z., Khamy, S. E., Dessouky, M. I., El-Dolil, S. A., & Abd El-Samie, F. E. (2012). Peak to average power ratio reduction using selective mapping with unequal power distribution. *Journal of Central South University*, 19(7), 1902–1908.
5. Renze, L., Longjiang, J., Lang, L., Jie, L., & Weile, Z. (2006). Reducing the peak to average power ratio of OFDM system with low complexity. *Journal of Electronics (China)*, 23(1), 26–28.
6. Hasan, M. M. (2014). PAPR reduction in OFDM systems based on autoregressive filtering. *Circuits, Systems and Signal Processing*, 33(5), 1637–1654.
7. Lee, B. M., Kim, Y., & Figueiredo, R. J. P. (2012). Performance analysis of the clipping scheme with SLM technique for PAPR reduction of OFDM signals in fading channels. *Wireless Personal Communications*, 63(2), 331–344.
8. Baig, I., & Jeoti, V. (2012). A new DCT matrix precoding based RI-OFDMA uplink system for PAPR reduction. In *4th International Conference on Intelligent and Advanced Systems* (p. 680).

Implementation of Modified Array Multiplier for WiMAX Deinterleaver Address Generation



Patil Nikita, Arun Kakhandki, S. Ramakrishna and Priyatam Kumar

Abstract One of the simpler techniques, which involve the implementation of generating the address of two-dimensional deinterleaver used in the WiMAX transmitter and receiver block is proposed using the Xilinx FPGA. The Arithmetic and Logic Unit performs various mathematical operations such as addition, subtraction, division and many other logical operations. Apart from these operations, Multiplication is one of the most fundamental operations to be carried out by this unit. The implementation of multipliers is required for the address generation of the channel interleaver. The multipliers need to be designed in such a way that they require high speed, low power, less area, and less delay, which is of significant interest in the research area. Many attempts have been carried out to reduce the generation of number of partial products in the process of multiplication. Array multiplier is one such multiplier. Most of the arithmetic operations will be performed using multipliers which consume the majority amount of the power in digital circuits. The process of multiplication involves the shift and add operations. The performance of multiplier can be improved by optimizing the adder circuit. The objective of this paper is to build the algorithm for generating the address of channel deinterleaver by using modified array multiplier. The algorithm is built using the Hardware Description Language Verilog and the functioning of the system can be verified through simulation and implemented using the Spartan-6. The simulation and implementation results have been obtained for the three different modulation techniques such as QPSK, 16-QAM, and 64-QAM for some of the information rates which proves to be a very good technique against the conventional methods.

Keywords Field-programmable gate array (FPGAs) · Multiplication · Array multiplier

P. Nikita · S. Ramakrishna · P. Kumar (✉)
Department of EC, BVB College of Engineering, Hubli 580031, India
e-mail: priyatam@bvb.edu

S. Ramakrishna
e-mail: ramakrishnasj444@gmail.com

A. Kakhandki
Department of EC, VDRIT College of Engineering, Haliyal 581329, India

© Springer Nature Singapore Pte Ltd. 2019
N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_52

1 Introduction

Wireless Communication refers to the type of communication in which the data can be transmitted, as well as received through wirelessly. Accessing the data through wireless communication with large rate of bandwidth is one of the more competing tasks compared to the wired communication technology. There are certain standards given by IEEE for carrying out the wireless mobile communication which is widely known as mobile WiMAX [1]. The channel interleaver/deinterleaver included in the WiMAX Tx and Rx block plays a very important role in reducing the effect of contiguous sequence of symbols that result in errors. The address generator circuit is designed and implemented for the channel deinterleaver which is used in the WiMAX Tx and Rx in such a way that requires high speed and less resources. The circuit is designed with low complexity, as well as it invalidates the necessity of generating floor function. There is a very small amount of work performed on the hardware implementation part of the interleaver/deinterleaver used in a WiMAX system as discussed in the literature review [2, 3].

In [4], the authors have described and demonstrated about grouping the incoming data streams in the form of blocks to minimize the occurrence of accessing the memory in a deinterleaver using a traditional LUT-based approach. The work is realized by using Complementary Metal–Oxide–Semiconductor address generator for WiMAX. In [5], the authors have carried out the work by implementing the address generator for IEEE 802.16e channel interleaver with only a $\frac{1}{2}$ code rate by using a hardware description language. In [6], the authors have demonstrated with the help of FSM-based address generator of the same interleaver for the different information rates and modulation techniques. The authors of [5, 6] have implemented on the FPGA platform. The authors of [7] have performed the two-dimensional transformation of the functions which are used in the WiMAX channel deinterleaver for using the hardware architecture efficiently. But the design issues have not been discussed clearly for 64-QAM [8].

Implementation of the floor function in the hardware circuit is very complicated since it requires more amount of resources [7]. One of the traditional techniques, which were used previously, is the look up table-based approach. This technique proved to be less efficient since it performed the operations slowly, utilization of large amount of logic resources, and so on. The proposed design provides the improvisation with respect to the resources, operations to be carried out by the system compared to the conventional look up table approach. Complex and lengthy expressions were used for the modulation techniques, due to the two-dimensional transformation of the functions used in the WiMax channel deinterleaver. The proposed system uses the minimized mathematical expressions and the resulting algorithm is presented. The mathematical equations have been proved using [7]. The proposed algorithm can be realized with the help of hardware circuit, which results in the architecture for address generation having less complexity, compared with the previous technique. The design is optimized in such a way that it requires a common hardware circuit among the modules for all the three modulation techniques [9]. The three modulation

techniques that have been used are QPSK, 16-QAM, and 64-QAM. The proposed algorithm consists of implementation of modified array multiplier, which helps in the reduction of power along with less area and improving the delay of circuit. This architecture is developed using hardware description language Verilog and implemented on the Spartan-6 FPGA. The proposed algorithm is intended to produce the address generated by the channel interleaver and the functioning of the hardware circuit is verified using Isim.

2 Design Methodology

This section describes about the overview of WiMAX transceiver and proposed algorithm.

2.1 WiMAX System

Due to the advancement in the field of wireless communication technology, WiMAX is one of the most emerging technologies that provide accessing for fixed and mobile users. The overview of WiMAX transmitter and receiver block is shown in Fig. 1. As seen in the below figure, the most important blocks of WiMAX are transmitter and receiver in order to transmit as well as receive the data wirelessly.

The description of the transmitter section is as follows. Source is one of the most important parts of the transmitter. It is used to transmit the information or data through the channel to one or more receivers. The data received from the source is sent to the randomizer. This operation is usually carried out for a certain set of data that can be transmitted or received in a cyclic manner. This operation is performed on each

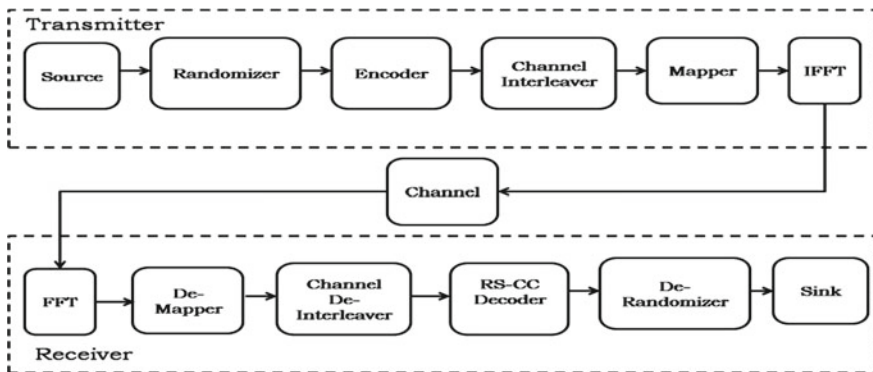


Fig. 1 Overview of WiMAX transceiver

burst on each allocation in order to avoid lengthy sequence of continuous logical ones and zeros. Data is further encoded by using both the coding techniques. The coding techniques used are RS and CC. The encoded data is further applied to the channel interleaver [10, 11]. The channel interleaver performs the permutation on the encoded data to minimize the consequence of continuous sequence of symbols which results in errors. The signals are further modulated and composed using the other two consecutive blocks, such as mapper and inverse fast Fourier transformation technique as shown in Fig. 1. The receiver block consists of the arrangement of the blocks in the bottom-up approach so that it enables the data to be restored in the elementary sequence at the output.

The functional representation of interleaver/deinterleaver is shown in Fig. 2. It provides the internal structure of the channel interleaver/deinterleaver in the WiMax system. The channel interleaver/deinterleaver plays a very important role in the WiMAX system. It consists of two memory blocks, multiplexers, and an address generator. The read and write operations are performed by the memory blocks depending upon the select lines. The data is written into the first memory block (M-1) when select line is selected as one. The read operation is performed on the interleaved data from the second memory block (M-2). Thus, read and write operations are performed simultaneously on the memory blocks depending on the status of select lines. Once the data is read/written up to the specified location as mentioned by the depth of interleaver, the position of the select signal will be switched to alter the read as well as write operations.

The interleaver/deinterleaver block accommodates distinct depth rates to include different information rates and modulation schemes for IEEE 802.16e. The data

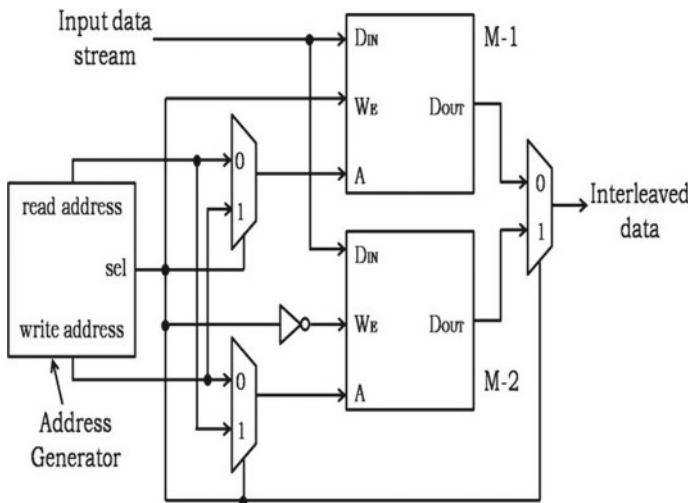


Fig. 2 Functional representation of interleaver/deinterleaver

which is encoded with the help of coding techniques is processed by using the Eqs. (1) and (2) which are noted below.

$$nk = \left(\frac{X_{cbps}}{m} \right) \cdot (k \% m) + \left[\frac{k}{m} \right] \tag{1}$$

$$pk = r \cdot \left[\frac{nk}{r} \right] + \left(nk + X_{cbps} - \left[\frac{m \cdot nk}{X_{cbps}} \right] \right) \% r \tag{2}$$

As seen in the above equations, m represents the number of columns. Here, it is considered as ($m = 16/12$ for WiMAX); the values of nk and pk , which are considered as the result obtained as soon as encoding of data is performed. The value of k as seen in the above equation can range from 0 to $X_{cbps} - 1$; r is given by $r = X_{cpc}/2$; basically fixed for different techniques. It can be considered as two, four, or six for QPSK, 16-QAM, or 64-QAM, respectively. The inverse operation which is performed by the deinterleaver is also defined by the equations which are mentioned below. The values of np and kp represent the permutations obtained for the deinterleaver, where p is considered as the index value of the bits received within a block of X_{cbps} bits.

$$np = r \cdot \left[\frac{p}{r} \right] + \left(p + \left[\frac{m \cdot p}{X_{cbps}} \right] \right) \% r \tag{3}$$

$$kp = m \cdot np - (X_{cbps} - 1) \cdot \left[\frac{m \cdot np}{X_{cbps}} \right] \tag{4}$$

Table 1 discusses about the depth of interleaver/deinterleaver for various information rates with modulation techniques.

Table 1 Addresses determined for channel interleaver

Row No. (p)	Column No. (i)	0	1	2	3	4
0	$X_{cbps} = 96$ bits, $\frac{1}{2}$ code rate, QPSK	0	16	32	48	64
1		1	17	33	49	65
2		2	18	34	50	66
3		3	19	35	51	67
0	$X_{cbps} = 192$ bits, $\frac{1}{2}$ code rate, 16-QAM	0	16	32	48	64
1		17	1	49	33	81
2		2	18	34	50	66
3		19	3	51	35	83
0	$X_{cbps} = 192$ bits, $\frac{1}{2}$ code rate, 64-QAM	0	16	32	48	64
1		17	33	1	65	81
2		34	2	18	82	50
3		3	19	35	51	67

2.2 Proposed Algorithm

The mathematical analysis for generating the address using address generator for the WiMAX deinterleaver has been shown in the above equations. The value of m is chosen as 16. The number of rows (p) is predetermined which are equal to m for all the various rates of interleaver depth which are in bits. The number of columns (i) is obtained by dividing the interleaver depth by m . The equations used for generating the addresses for all the three modulation techniques are as follows.

$$k_n, QPSK = m * i + p \text{ for all the values of } p \text{ and } i \tag{5}$$

$$k_n, 16 - QAM = \left. \begin{array}{l} m * i + p \quad \text{for } p \% 2 = 0 \text{ and for } i \\ m * (i + 1) + p \text{ for } p \% 2 = 1 \text{ and for } i \% 2 = 0 \\ m * (i - 1) + p \text{ for } p \% 2 = 1 \text{ and for } i \% 2 = 1 \end{array} \right\} \tag{6}$$

$$k_n, 64 - QAM = \left. \begin{array}{l} m * i + p \quad \text{for } p \% 3 = 0 \text{ and for } i \\ m * (i - 2) + p \text{ for } p \% 3 = 1 \text{ and for } i \% 3 = 2 \\ m * (i + 1) + p \text{ for } p \% 3 = 1 \text{ and for } i \% 3 \neq 2 \\ m * (i + 2) + p \text{ for } p \% 3 = 2 \text{ and for } i \% 3 = 0 \\ m * (i - 1) + p \text{ for } p \% 3 = 2 \text{ and for } i \% 3 = 0 \end{array} \right\} \tag{7}$$

As seen in the Eqs. (5), (6), and (7), p considers the values from 0, 1, ..., $m - 1$ and i considers the values from 0, 1, ..., $(X_{cbps}/m) - 1$ represents the row and column numbers, respectively, and k_n represents the addresses of deinterleaver. Using the above Eqs. (5), (6), and (7), the addresses have been determined for all the three modulation schemes. Table 2 shows the addresses determined for all the three modulation techniques with code rates.

The algorithms can be further transformed into the circuits as shown in Figs. 3 and 4, respectively. The hardware circuit for the Quadrature Phase Shift Keying is shown in Fig. 3. It consists of various blocks such as row counter, comparator, column counter, and multiplexer. The row counter initiates with generating the row numbers between 0 and $m - 1$. The column counter which is built along with the Mux M0

Table 2 Depth of interleaver/deinterleaver for various information rates with modulation techniques

Modulation scheme	QPSK ($s = 1$)		16-QAM ($s = 2$)		64-QAM ($s = 3$)		
	1/2	3/4	1/2	3/4	1/2	2/3	3/4
Code rate	1/2	3/4	1/2	3/4	1/2	2/3	3/4
Interleaver depth, X_{cbps} in bits	96	144	192	288	288	384	432
	192	288	384	576	576	–	–
	288	432	576	–	–	–	–
	384	576	–	–	–	–	–

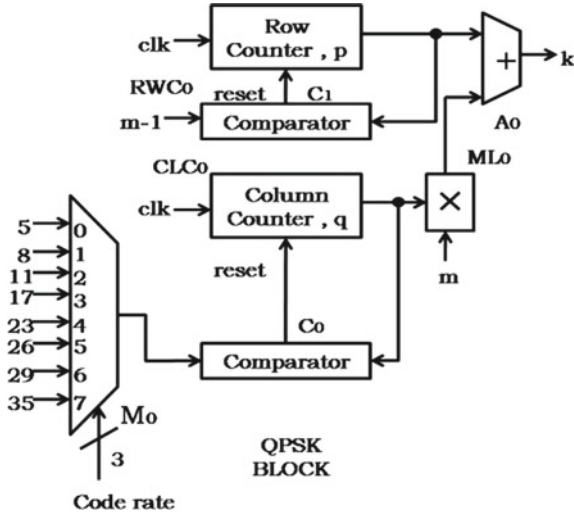


Fig. 3 Hardware structure of QPSK

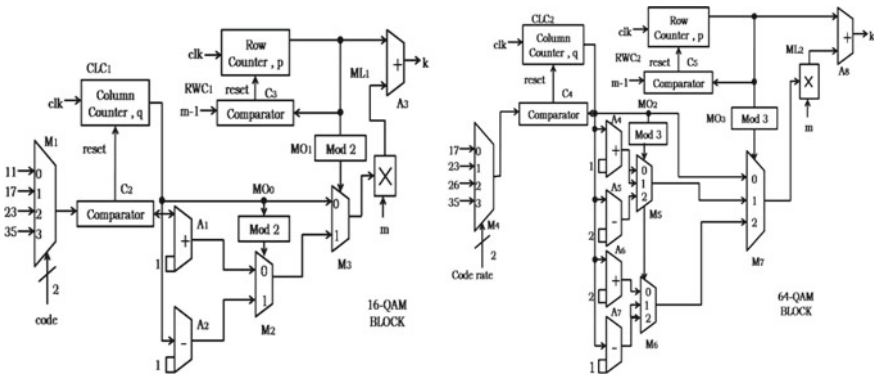
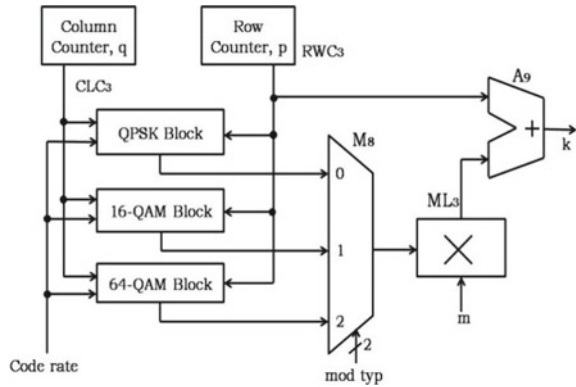


Fig. 4 Hardware structure of 16-QAM and 64-QAM

and comparator C0 is used for generating the column numbers to implement the allowable coded bits per second termed as X_{cbps} . The multiplier and an adder unit are used to perform the required operations to produce the value of k_n (address of the deinterleaver). The hardware circuits for the 16-QAM and 64-QAM are similar to the circuit of the QPSK. The additional modules required for the design of 16-QAM and 64-QAM are an incrementer, which increases the values by 1 and provides them to the mux, an decremter, which decreases the values by 1 and transmits them to the mux, modulo blocks as shown in Figs. 4 and 5. The operation of modulo 2 is performed by the hardware circuit of 16-QAM and the operation of modulo 3 is performed by the hardware circuit of 64-QAM as mentioned in the above part of the Eqs. (6) and (7).

Fig. 5 Schematic structure of deinterleaver address generator



The Schematic structure of the complete deinterleaver address generator is shown in Fig. 5. It consists of row counter (j), column counter (i), QPSK block, 16-QAM block, 64-QAM block, and multiplexer. The multiplexer is used to select the modulation techniques depending upon the combination given to the select lines. The combination of multiplier and adder units is used to generate the address for the deinterleaver.

3 Simulation and Implementation Results

The functioning of the hardware circuit of the deinterleaver address generator is verified by programming using the Verilog with the help of Xilinx tool. The simulation results have been obtained for all the three different modulation techniques using Isim.

The simulation results obtained for the 16-bit modified array multiplier is shown in Fig. 6. The results are generated for the six different combinations of data. This 16-bit modified array multiplier is utilized in the hardware circuit of schematic structure of deinterleaver address generator. The usage of modified array multiplier in this circuit significantly reduces the power.

The simulation results obtained for the QPSK is shown in Fig. 7. The deinterleaver address has been generated for the QPSK as shown in Table 2. The results have been verified using Table 2.

The simulation results obtained for the 16-bit QAM is shown in Fig. 8. The deinterleaver address has been generated for the 16-bit QAM as shown in Table 2. The results have been verified using Table 2.

The simulation results obtained for the 64-bit QAM is shown in Fig. 9. The deinterleaver address has been generated for the 64-QAM as shown in Table 2. The results have been verified using Table 2.



Fig. 6 Simulation results of 16-bit modified array multiplier

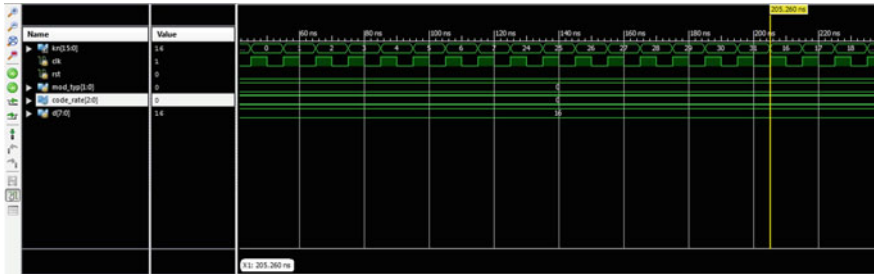


Fig. 7 Simulation results of QPSK

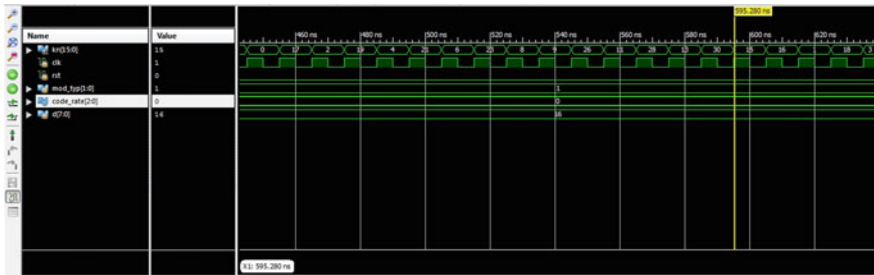


Fig. 8 Simulation results of 16-QAM

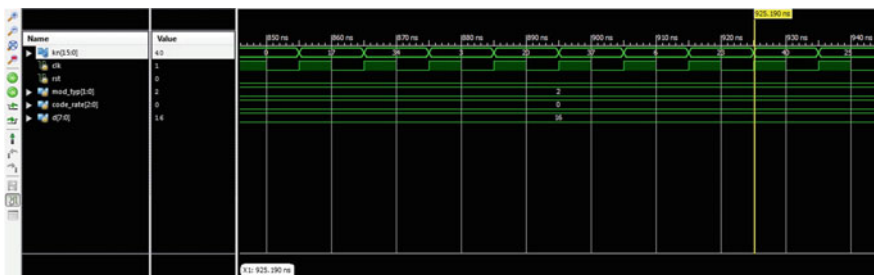


Fig. 9 Simulation results of 64-QAM

Table 3 Synthesis report of the proposed algorithm

Logic circuits	Quantity
Adders/Subtractors	19
Counters	5
Registers	13
Comparators	21
Multiplexers	581

Table 4 Performance of the proposed algorithm with the existing algorithm

Different approaches	No. of registers	No. of LUTs	No. of bonded IOs	Max frequency (MHz)
LUT-based approach	62	714	26	210.402
16-bit modified array multiplier	47	220	30	204.505

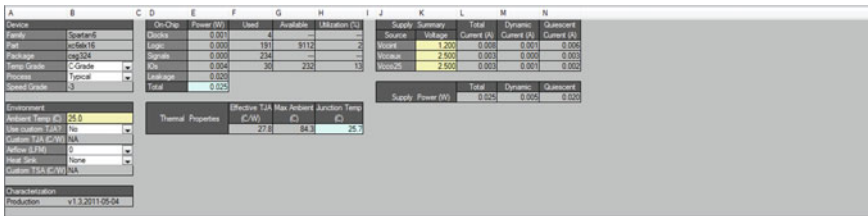


Fig. 10 Power analysis of the algorithm

The implementation details are discussed in Tables 3 and 4. Table 3 specifies about the logic circuits required for the proposed algorithm and Table 4 specifies about the performance of the proposed algorithm with the existing method. It shows the comparison of parameters between the two different approaches. As per the comparison carried out between the two approaches, in terms of resources, the modified array multiplier accommodates with less number of resources. The power analysis of the algorithm is shown in Fig. 10. The dynamic power consumed is nearly 0.005 W and the quiescent power consumed is nearly 0.020 W. The total amount of power consumed is 0.025 W.

4 Conclusion

This paper represents the three different modulation techniques of address generation using the 16-bit modified array multiplier. The proposed algorithm is transformed into the hardware circuit for all three different modulation techniques. The results have

been obtained using the Xilinx tool and implementation is performed using Spartan-6. Thus, it is concluded that the address values for the channel interleaver/deinterleaver are generated by using 16-bit modified array multiplier. The implemented results discuss about the utilization of resources for the system. The comparison between the two approaches such as LUT-based approach and by using modified array multiplier is carried out in terms of resources and frequency. The power analysis is also carried out for the system.

References

1. Konhauser, W. (2006). Broadband wireless access solutions-Progressive challenges and potential value of next generation. *Wireless Personal Communications*, 37(3/4), 243–259.
2. Li, B., Qin, Y., Low, C. P., & Gwee, C. L. (2007). A survey on mobile WiMAX. *IEEE Communications Magazine*, 45(12), 70–75.
3. Andrews, J. G., Ghosh, A., & Muhamed, R. (2007). *Fundamentals of WiMAX: Understanding broadband wireless networking*. Upper Saddle River, NJ, USA: Prentice-Hall.
4. Chang, Y. N., & Ding, Y. C. (2007). A low-cost dual mode de-interleaver design. In *Proceedings of International Conference on Consumer Electronics* (pp. 1–2).
5. Khater, A. A., Khairy, M. M., & Habib, S. E. D. (2009). Efficient FPGA implementation for the IEEE 802.16e interleaver. In *Proceedings of International Conference on Microelectronics* (pp. 181–184). Marrakech, Morocco.
6. Upadhyaya, B. K., Misra, I. S., & Sanyal, S. K. (2010). Novel design of address generator for WiMAX multimode interleaver using FPGA based finite state machine. In *Proceedings of 13th International Conference on Computer and Information Technology* (pp. 153–158). Dhaka, Bangladesh.
7. Asghar, R., & Liu, D. (2009). 2D realization of WiMAX channel interleaver for efficient hardware implementation. In *Proceedings of World Academy of Science, Engineering and Technology* (vol. 51, pp. 25–29), Hong Kong.
8. IEEE Standard for Local and Metropolitan Area Networks-Part 16: Air Interface for Fixed Broadband Wireless Access Systems-Amendment 2. (2005). IEEE Std. 802.16e-2005.
9. Khan, M. N., & Ghauri, S. (2008). The WiMAX 802.16e physical layer model. In *Proceedings of IET Conference on Wireless, Mobile and Multimedia Networks* (pp. 117–120). Mumbai, India.
10. Wallace, C. A suggestion for a fast multiplier. *IEEE Transaction on Electronic Computers*.
11. Nikita, P., & Kakhandki, A. (2017). Study of low power array multiplier for real time applications. *International Journal of Research and Advanced Development (IJRAD)*, 1(3).
12. Srikanth, S., et al. (2016). Low power array multiplier using modified full adder. In *2016 IEEE International Conference on Engineering and Technology (ICETECH)*. IEEE.

Link Quality-Based Mobile-Controlled Handoff Analysis Using Stochastic Models



S. Akshitha and N. G. Goudru

Abstract With limited available frequency spectrum and increasing demand for cellular communication services, the problem of handoff becomes increasingly important. Poorly designed handoff schemes tend to generate very heavy signaling traffic and thereby, a drastic decrease in the quality of service (QoS). In urban mobile cellular radio system, the cell size is small. The handoff procedure has a significant impact on the system performance. Blocking probability of originating calls and forced termination probability of ongoing calls reflects the performance of the system. In this paper, we use mobile-controlled handoff process. MS is moving from one cell to another cell. Relative signal strength and relative residual energy are measured. Important channel quality parameters, for the channels with line of sight (α) and without line of sight (β) are considered. Using these parameters, weights of the channels are determined in three neighboring cells and the servicing cell. The channel weights of four cells are compared with the chosen signal strength threshold value to make a decision of handoff process or to continue service with the same base station (BS). Prioritized hard handoff is preferred in the work. Handoff performance metrics such as arrival rate of originating calls, blocking probability of originating calls, arrival rate of handoff calls, blocking probability of handoff calls, and network parameters such as sender window size, queue length at the servicing link are measured. Performance analysis parameters such as packet loss due to congestion, round-trip delay, and throughput are discussed. It helps in controlling the oscillatory behavior of the switch or router normally used for traffic control at the BS. Using the results and applying an explicit feedback, the link congestion and packet losses can be minimized. It helps for improving the quality of service (QoS) in 3G, 4G, and next-generation mobile communication. Stochastic models are used to analyze the system performance. Using MATLAB programme, result analysis is made through graphs and statistical data.

S. Akshitha (✉) · N. G. Goudru
Department of ISE, Nitte Meenakshi Institute of Technology (Affiliated to Visvesvaraya Technological University), Bangalore 560064, India
e-mail: akshitha.yeshu008@gmail.com

N. G. Goudru
e-mail: goudru.ng@nmit.ac.in

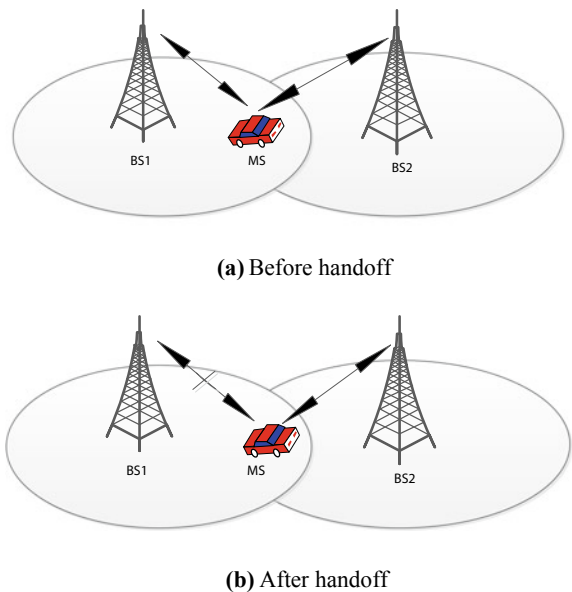
Keywords Cell · Congestion · Hard handoff · Mobility · Probability · Stochastic

1 Introduction

Mobility is one of the most important features of a cellular communication system. In mobile wireless communication system, continuous service is achieved using a procedure called handoff. During handoff process, the channel associated with the current servicing connection is changing while a call is in progress. It is usually initiated either by crossing a cell boundary or by deterioration in the quality of signal in the current channel. In the handoff literature, two types of handoff procedures are dominant, namely hard handoff and soft handoff. Hard handoff is characterized by “break before make” and soft handoff is characterized by “make before break”. In hard handoff, the current resources are released before new resources are used. In soft handoff, both existing and new resources are used during the process. Hence, we are interested in hard handoff. In hard handoff, multiple access techniques such as frequency-division multiple access (FDMA) and time-division multiple access (TDMA) are used. In cellular structure, different frequency ranges are used in adjacent channels in order to minimize the channel interference.

Figure 1a shows that MS is moving from base station (BS1) to adjacent base station (BS2). Mobile station (MS) is serviced by BS₁ and moving toward BS₂ and no handoff is taken place. Figure 1b shows that MS has entered into the handoff region and handoff take place.

Fig. 1 a Before handoff.
b After handoff



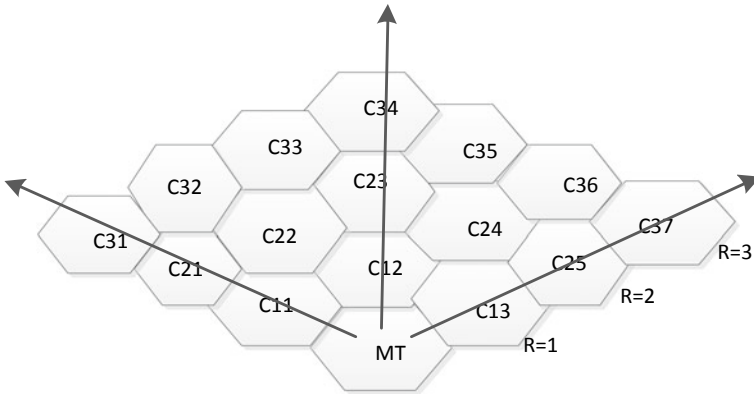


Fig. 2 Cell configuration of MT for $r = 3$

A most widely used cell structure of the coverage of mobile towers (MT) is shown in Fig. 2. The actual design segment taken is at 120° orientation of the coverage area. The mean signal strength of BS_1 decreases as MS moves away and mean signal strength increase as MS approach BS_2 .

In prioritized hard handoff model, we assume the cell radial distance r is small and mobile callers are equally distributed in the coverage area. Mobile station is moving with a constant velocity in any direction from one radial distance to another. Probability of direction of the movement either away from the current base station or along the same radial level from current BS toward new adjacent BS is equal. Movement of MSs in the cells is shown in Fig. 3. The movements are random and no restriction is imposed but handoff is possible only when an MS either crosses cell boundary or relative signal strength (RSS) is lower than the conveniently chosen threshold value.

1.1 Relative Signal Strength Indicator (RSSI) and Threshold Value

RSSI is the relative received signal strength in a wireless environment. RSSI is an indication of the power level being received by the MS after the antenna. Therefore, the higher the RSSI number, the stronger the signal. The BS controller (BSC) is in charge of RSSI management. BSC takes care of release of the radio channel and handoff management. The performance of RSSI scheme between two adjacent BSs is shown in Fig. 4. When threshold value is higher than RSS value, handoff occurs at position A. When threshold value is lower than RSS value, MS would delay handoff until the current signal level crosses the threshold value at position B. When the threshold value is greater than RSS value, handoff would occur at position C. When

the delay is more so that MS drift too far into the new cell, reduces the quality of link from BS_1 and may result in connection drop [1].

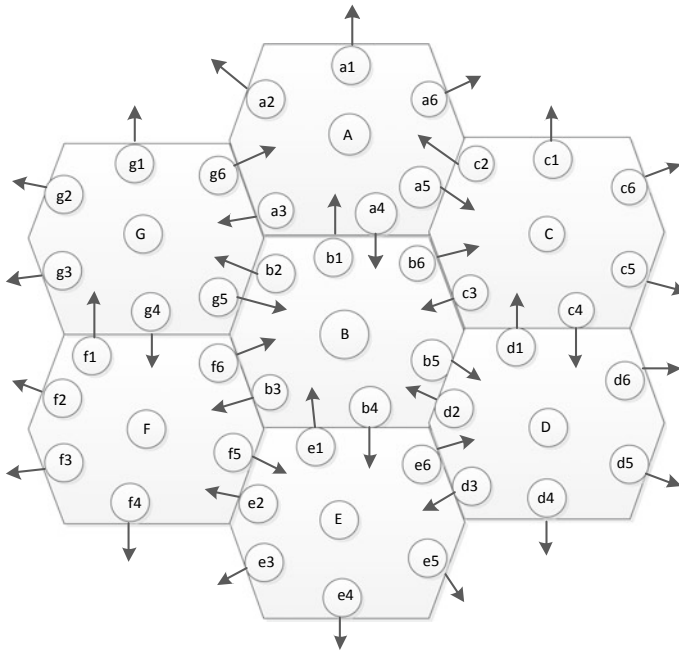


Fig. 3 Movement of the MSs

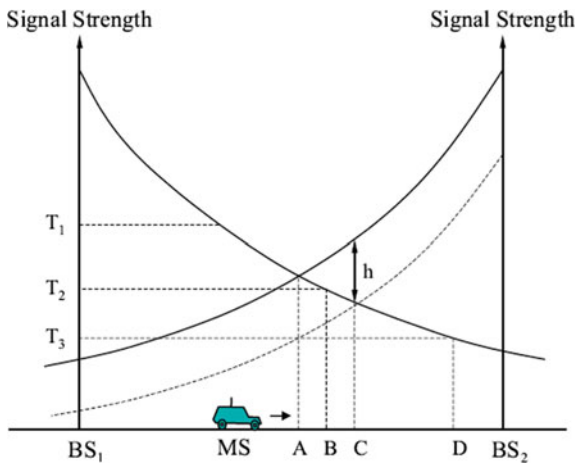


Fig. 4 Signal strength between two BSs for potential handoff

1.2 Handoff Prediction Technique

In mobile-controlled handoff process, each MS is completely in control of the handoff process. This type of handoff has a short reaction time of 0.1 s. MS continuously measures the signal strengths from the neighboring BSs. A handoff will be initiated if the signal strength of the serving BS is lower than threshold value. MS checks for the best candidate BS based on the weight of the channel. Figure 5 illustrates the link quality maintenance process.

2 Related Work

In digital cellular system, a BS includes an antenna, a controller and a number of receivers. Mobile telecommunication switching office (MTSO) connects cells between mobile units. Two types of channels are available between MS and BS. Control channels is used to exchange information for connection setting up and maintaining. Traffic channels which carry voice or data between users. With the growing demand for mobile services, many researchers are working on for achieving faster and guaranteed quality of mobile services. The researchers B.B. Madan and others discuss mobile-assisted handoff (MAHO) where mobile terminal (MT) assists to the serving base station system (BSS) and mobile switching center (MSC) in making handoff decisions. To avoid the chances of handoff drops due to non-availability of channels, MAHO is combined with channels reserved for handoff. This makes system implementation at BSS more complicated [2]. The researchers Amine Boubekri and others use both link strength between mobile and potential static parent nodes and residual energy of the parent nodes to make handoff decisions. They ignore the node energy state, and the results are discussed through simulation using simulators [3]. The authors Kishore Trivedi and others present a closed-form solutions to the probabilities of blocking and dropping in wireless cellular networks. A fixed point iteration method is used to estimate handoff arrival to BS. Allowing failures and repairs of channels tried to give performance analysis for cellular system [4]. The authors Ing-Ray Chen and others apply algorithmic-based admission control has been used by the authors to analyze a class of partitioning and threshold. They give “change by time” scheme for optimal resource allocation without sacrificing QoS requirements [5]. The researchers Buyurman Baykal and others preset expected visitor list (EVL) method to achieve minimum handoff blocking probability keeping QoS levels and also minimize the handoff latency. The call admission control (CAC) run against each EVL entry. Without resource reservation or resource allocation, the analysis is made for varying network conditions [6]. The authors Biswajit Bhowmik and others discuss on modeling hard handoff scheme and implementing traffic model to study the handoff behavior for wireless mobile networks. They tried to calculate originating calls, probability of handoff blocking. [7]. The authors Tao Liu and others suggest a link estimation method to predict link quality status. It is a

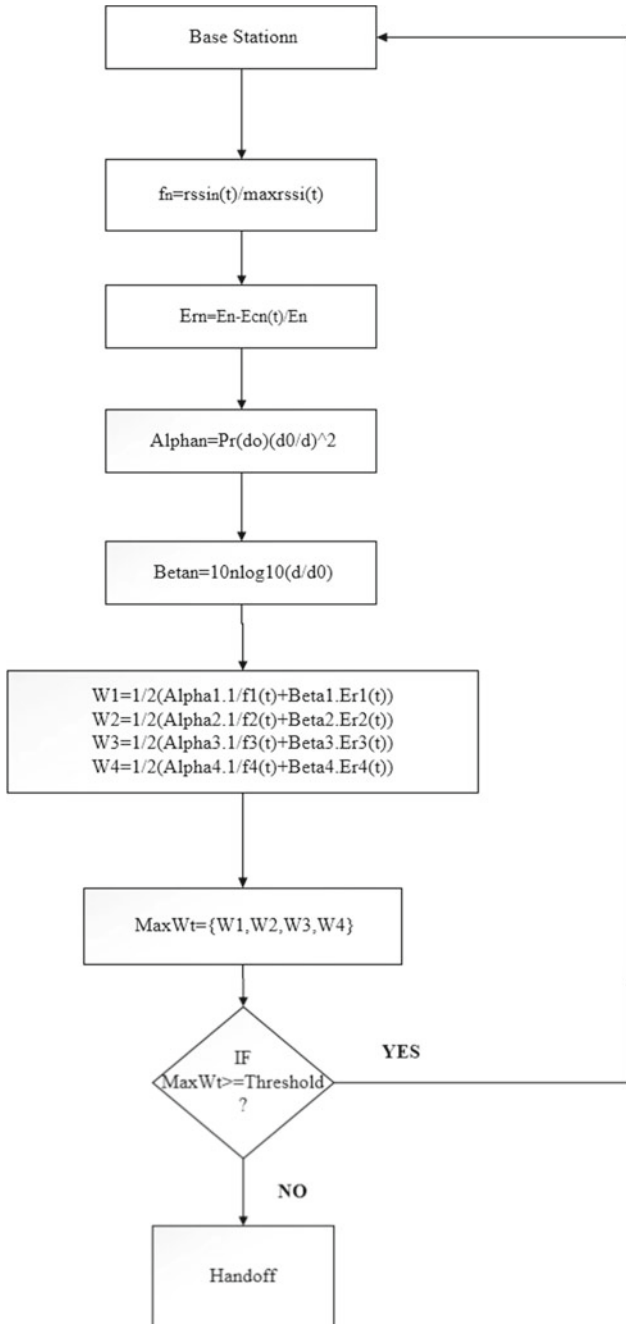


Fig. 5 MS link quality maintaining process

three-stage implementation, involving data collection, offline modeling, and online prediction. Logistic regression model is regarded as a best model for prediction [8]. The researchers Qiuming Liu and others study the influence of network topology on delay analysis in large-scale wireless ad hoc networks. The packet transmission delay is used as an important parameter in the network design. Applying routing algorithm deduced the service curve [9]. The research reviews reveal that very less work has been discussed on link quality-based mobile-controlled hard handoff analysis using stochastic models.

3 System Model for Mobile Wireless Networks

With reference to [3], the link quality metric is given by

$$f_n(t) = \frac{\text{rssi}_n(t)}{\text{maxrssi}_n(t)} \tag{1}$$

where rssi_n is the radio signal strength indicator value measured after time t and maxrssi_n is the best signal strength indicator value measured. The residual energy of the BS is given by

$$E_r(t) = \frac{E_n(t) - E_{c_n}(t)}{E_n(t)} \tag{2}$$

where $E_n(t)$ is the total energy available at the BSs initially and $E_{c_n}(t)$ is the consumed energy. The link weight is estimated by using the relation

$$W_n = \alpha \cdot \frac{1}{f_n(t)} + \beta \cdot E_r(t) \tag{3}$$

where α and β are parameters associated with the channel quality. The parameter α is associated with line of sight and the parameter β is associated for receiver without line of sight. The parameter α and β are estimated by using the relations

$$\alpha = P_r(d_0) \left(\frac{d_0}{d} \right)^2 \tag{4}$$

$$\beta = 10_n \log_n \left(\frac{d}{d_0} \right) \tag{5}$$

With reference to [10], the sender window dynamics is given by

$$\frac{\partial w}{\partial t} = \frac{1}{R(t)} - \frac{w(t)w(t - R(t))}{2R(t - R(t))} p(t - R(t)) + \beta \frac{w(t - R(t))}{R(t - R(t))} \tag{6}$$

The sender TCP window operates on the principle of additive increase and multiplicative decrease (AIMD) congestion control strategy. The parameter α is the rate of decrease in source window and normally fixed as 0.5. $L_a(t)$ is arrival rate of loss due to congestion at any time t and $L_a(t) = \frac{w(t-R(t))}{R(t-R(t))} p(t - R(t - R(t)))$. The loss is proportional to packet sending rate. The first term on right-hand side (RHS) of the Eq. (2.1) represent linear growth of cwnd, until congestion occurs in the bottleneck link, second term deals with congestion control scheme using feedback from RED system, and third term $L_t(t)$ refer to bit error loss and immediate-recovery in wireless. But transmission loss is proportional to the sending rate, therefore $L_t(t)$ is proportional to $\frac{w(t-R(t))}{R(t-R(t))}$. The queue dynamics is given by

$$\frac{\partial q(t)}{\partial t} = \frac{Nw(t)}{R(t)} - C_d \tag{7}$$

where $w(t)/R(t)$ give an increase in queue size because of packets arrival from N—TCP flows. $C_d = q(t)/R(t)$ represent decreasing queue because of departure of packets after servicing from the router. In the proposed scheme, service time is variable. The most important advantages of using instantaneous queue length over average queue length is faster detection of congestion at the BS.

The buffer size at the BS is given by

$$W_{\max} = \frac{C_d}{S} R(t) + M \tag{8}$$

The round-trip time is given by

$$R(t) = \frac{q(t)}{C} + T_p \tag{9}$$

The wireless loss predictor using normal delay gradient is given by

$$\text{fNDG} = \left(\frac{\text{RTT}_i - \text{RTT}_{i-1}}{\text{RTT}_i + \text{RTT}_{i-1}} \right) \left(\frac{W_i + W_{i-1}}{W_i - W_{i-1}} \right) \tag{10}$$

The mathematical version of congestion feedback scheme is given by

$$P(t) = \begin{cases} 0, & q(t) \in [0, t_{\min}] \\ \frac{q(t) - t_{\min}}{W_{\max} - t_{\min}} P_{\max}, & q(t) \in [t_{\min}, W_{\max}] \\ 1, & q(t) \geq W_{\max} \end{cases} \tag{11}$$

4 Simulation and Performance Analysis

A number of experiments using simulation were conducted to study the performance of mobile assisted handoff. The proposed network model with one BS is illustrated as shown in Fig. 6. Similarly, in mobility, we use four base stations BS1, BS2, BS3, and BS4. Computers 1, 2, and 3 are the sources, R₁ and R₂ are the routers, and D₁, D₂, and D₃ are the mobile stations which are destinations. In the experiment, the link between R₁ and R₂ is the bottleneck link. Packet size is 1000 bytes, $P_{max} = 0.01$ s, queue buffer at the router has a minimum threshold value, $t_{min} = 300$ packets, maximum threshold value, $W_{max} = 500$ packets, initial RTT = 10 ms, $C_d = 10$ Mbps, $C_u = 500$ Kbps, $N = 10$, $\beta = 0.1$, t_p (propagation delay) = 10 ms. $S = 20$, $Sc = 10$, $+Sr = 10$.

The graphs in Figs. 7, 8, 9, and 10 shows the variation of relative signal strength as MS moves away from BSs. The reason behind considering four BSs is that the range of 120° has coverage of three adjacent BSs and the other one is current serving BS. The BSs 1, 2, 3, and 4 are assigned with -30 to -60, -61 to -90, -91 to -120, and -121 to -150 MHz frequencies. The RSSI has maximum value nearer to BSs and decrease as MS moves away from the BSs. The variation of RSSI corresponding to BSs 1, 2, 3, and 4 varies over the range [0.9917, 0.5167], [0.9944, 0.6778], [0.9958, 0.7583], and [0.9967, 0.8067], respectively.

The graphs of Figs. 11, 12, 13, and 14 depict the variation of the residual energy as MS moves away from BSs. The residual energy level decrease as MS moves away from BSs. The variation of energy levels corresponding to BSs 1, 2, 3, and 4 varies over the range [0.4790, 0.0159], [0.3184, 0.0081], [0.2417, 0.0055], and [0.1906, 0.0041], respectively.

The graphs of Figs. 15, 16, 17, and 18 shows the variation of the link weight as MS moves away from BSs. The link weight levels decrease slowly as MS moves

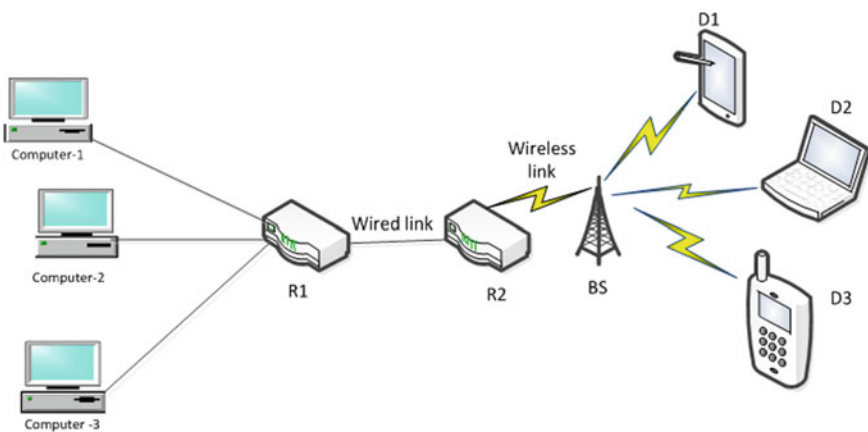


Fig. 6 Network model

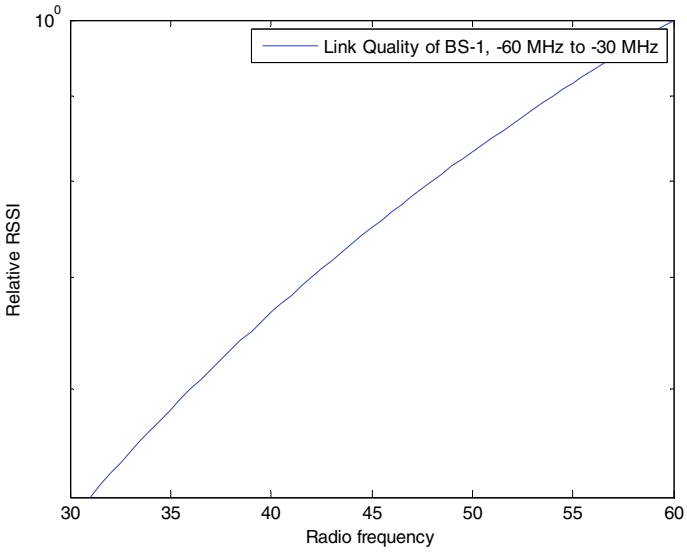


Fig. 7 RSSI variation of BS1

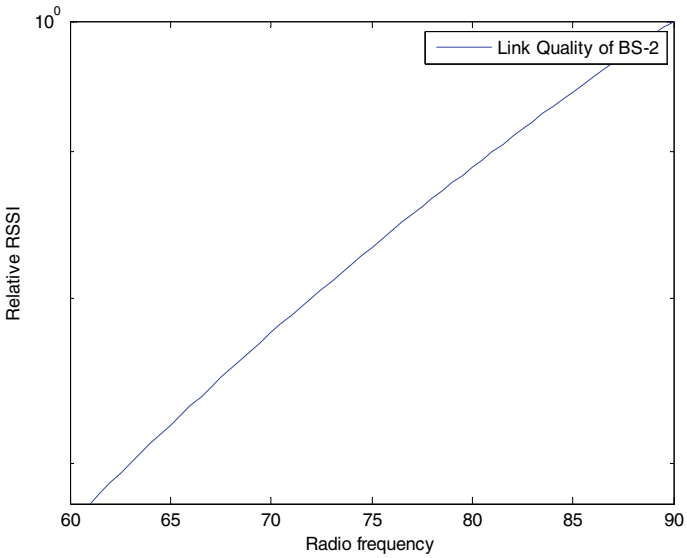


Fig. 8 RSSI variation of BS2

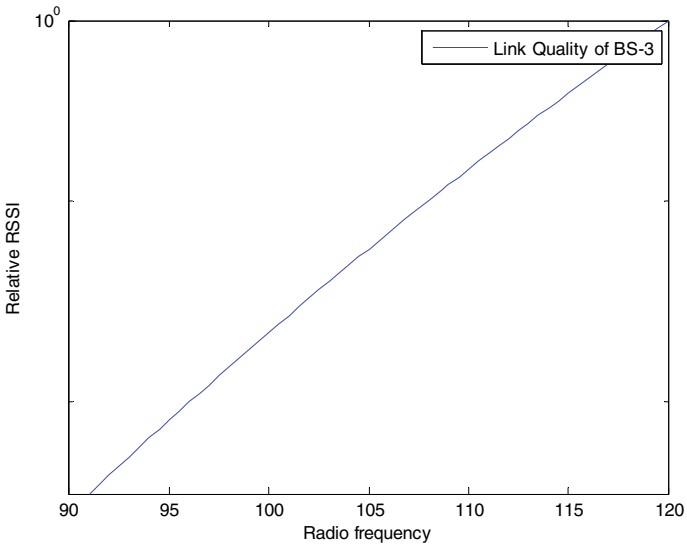


Fig. 9 RSSI variation of BS3

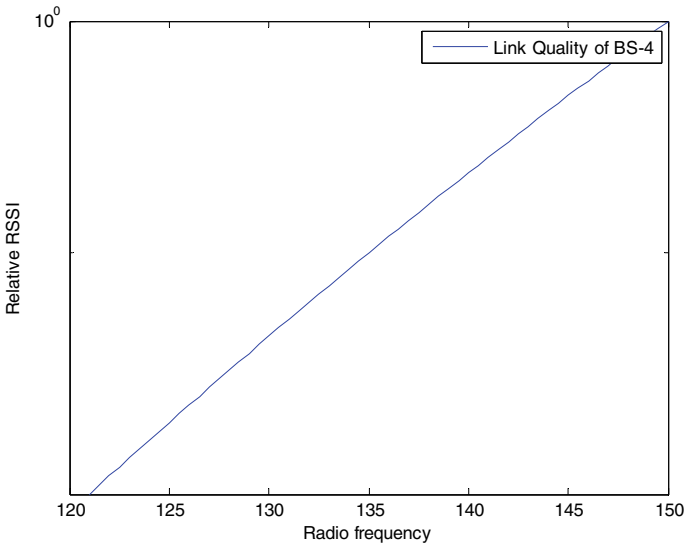


Fig. 10 RSSI variation of BS4

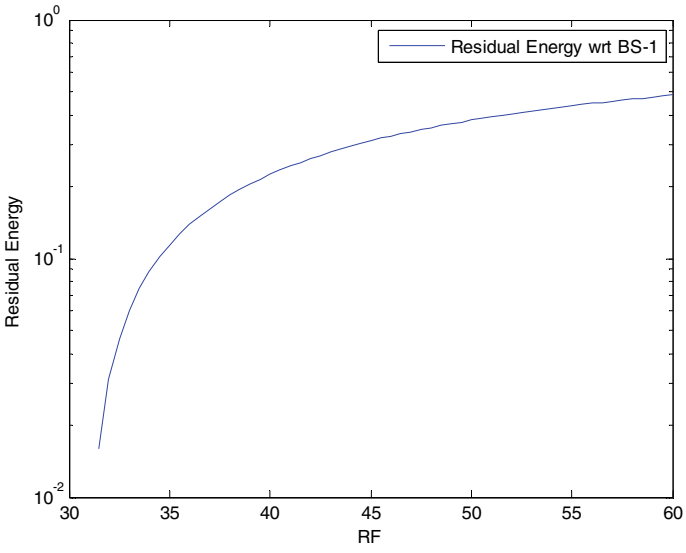


Fig. 11 Residual energy at BS1

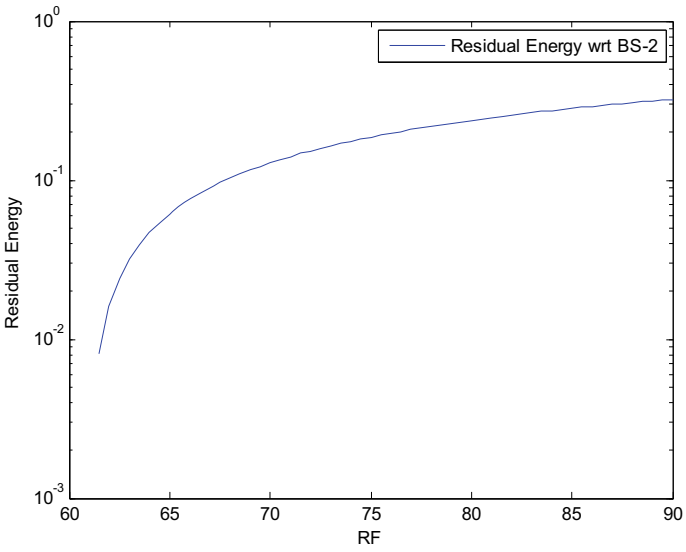


Fig. 12 Residual energy at BS2

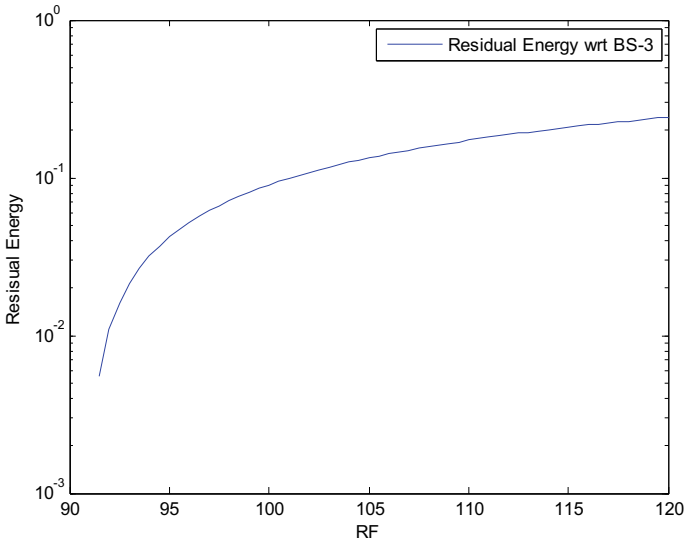


Fig. 13 Residual energy at BS3

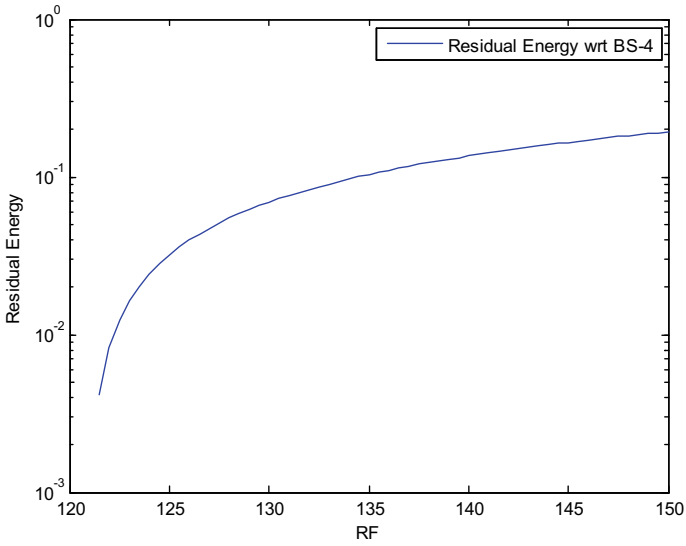


Fig. 14 Residual energy at BS4

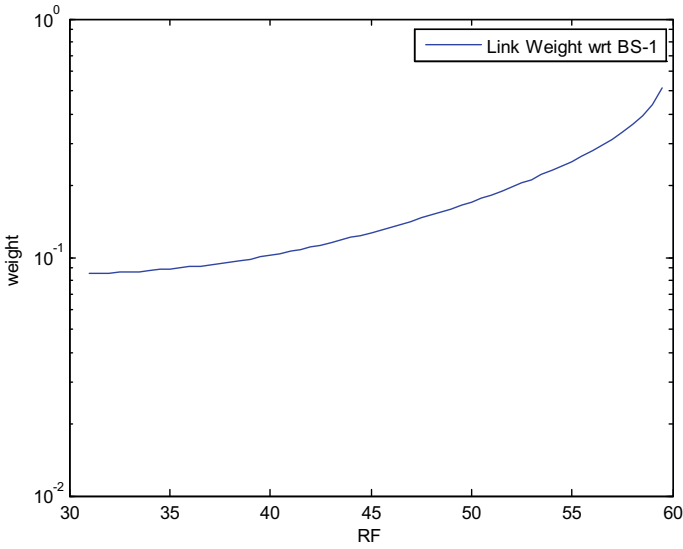


Fig. 15 Variation of link weight versus RF (BS1)

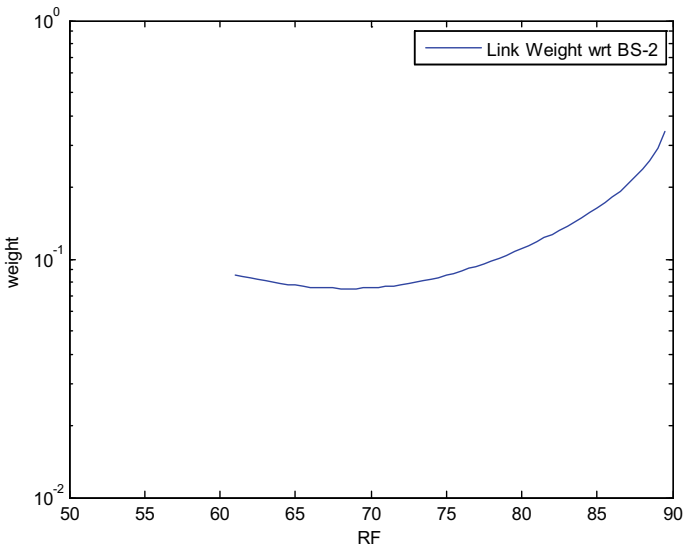


Fig. 16 Variation of link weight versus RF (BS2)

away from the BSs. The link weight levels corresponding to BSs 1, 2, 3 and 4 varies over the range [0.5140, 0.0858], [0.3417, 0.0858], [0.2559, 0.0858], and [0.2046, 0.0858] respectively.

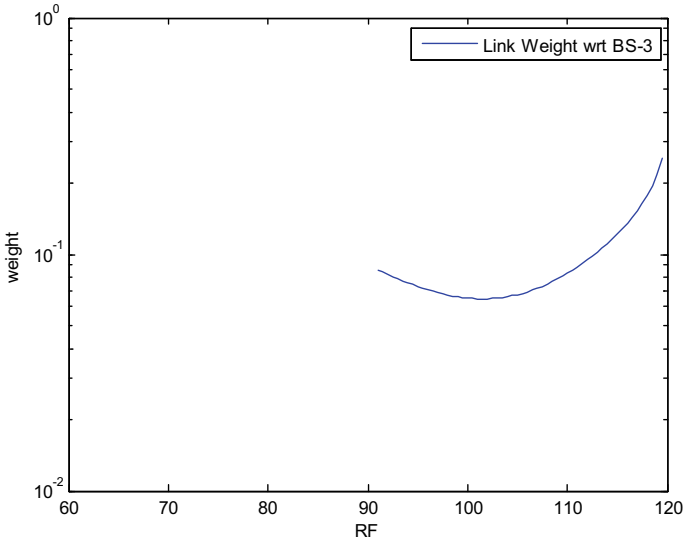


Fig. 17 Variation of link weight versus RF (BS3)

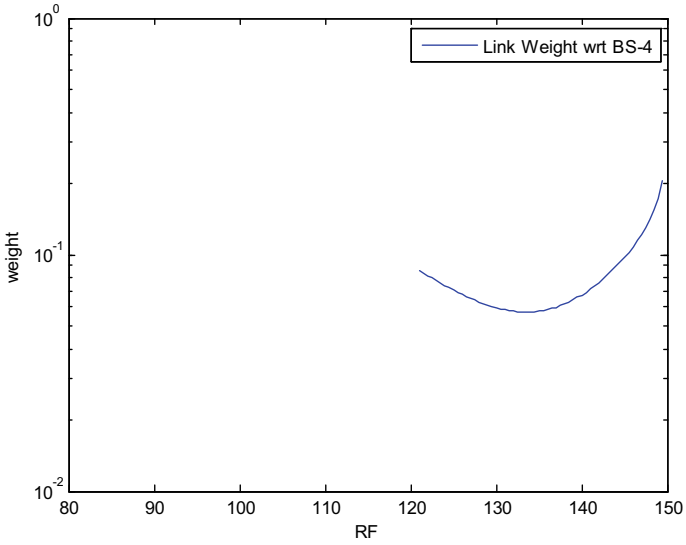


Fig. 18 Variation of link weight versus RF (BS4)

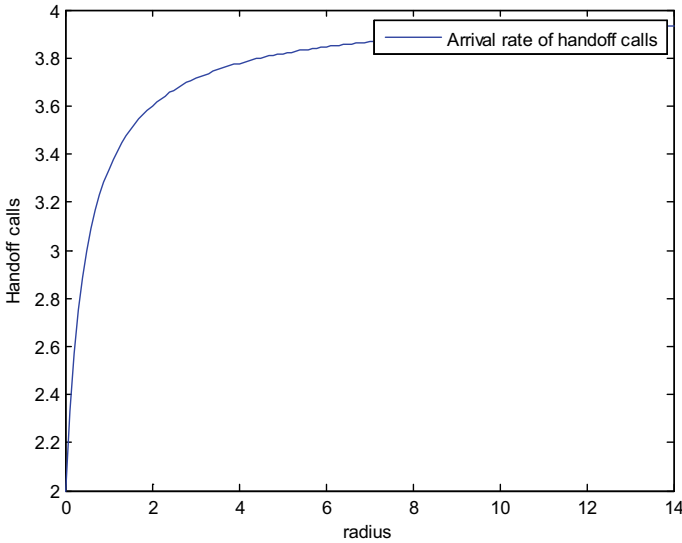


Fig. 19 Arrival rate of handoff calls versus radius

In cell structure, the adjacent cell is considered to be just upper or lower level where MS is moving toward or away from the BS. The handoff request arrival rate in radius order is demonstrated by a graph in Fig. 19. The statistical data illustrates that λ_H varies over the range of [3.3333, 3.9310].

The graph presented in Fig. 20 demonstrates the growth rate of blocking probability of originating calls in different radial distances. The value of B_H varies over the range [0.001, 0.026].

The graph presented in Fig. 21 demonstrates the growth rate of blocking probability of handoff request calls in different radial distances. The value of B_H varies over the range [0, 1].

The variation of TCP source window size is an important factor for the evaluation of performance of throughput. Graph of Fig. 22 show the source window dynamics. It varies over the range of [7, 67] packets.

Figure 23 describes the behavior of queue in the bottleneck link due to handoff blocking. Queue varies over the range of [14, 500] packets. During simulation, it is observed that queue size is larger, round-trip time is higher causing reduction in sender window size. By maintaining smaller queue length, performance of throughput can be improved.

The graph of Fig. 24 illustrates round-trip time delay distributed over the radius. The rtt-delay changes in the interval [0.032 s, 0.3832 s]. Analysis of the graph leads to a conclusion that smaller the rtt-delay, faster the end-to-end transportation and higher the sending rate. In reality, if rtt value of latest packet is smaller than previous packet, the network ignores that value for some time. Thus, an efficient rtt value can be selected to use as metric for balance load of handoff request calls (λ_H).

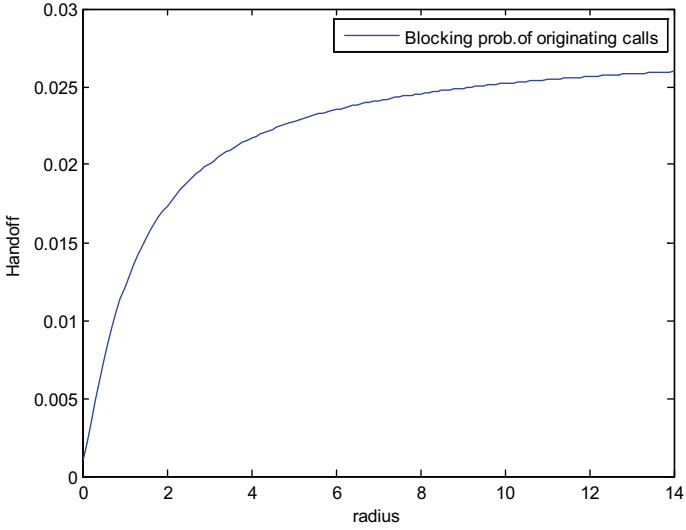


Fig. 20 Blocking probability of originating calls versus radius

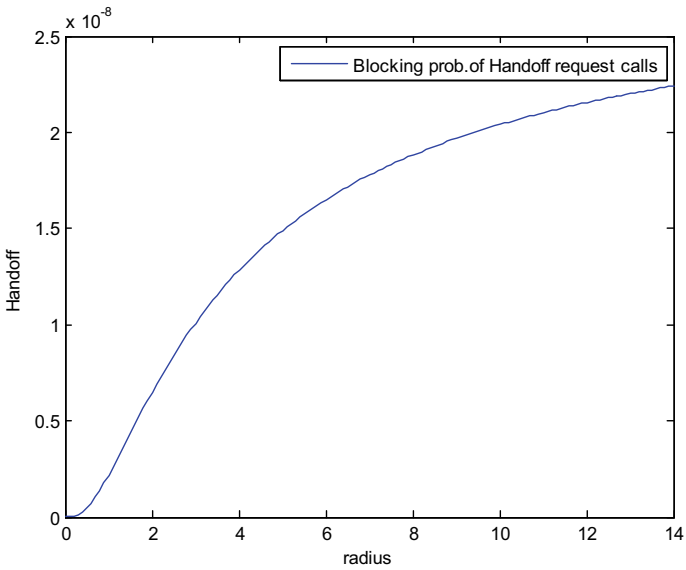


Fig. 21 Blocking probability of handoff calls versus radius

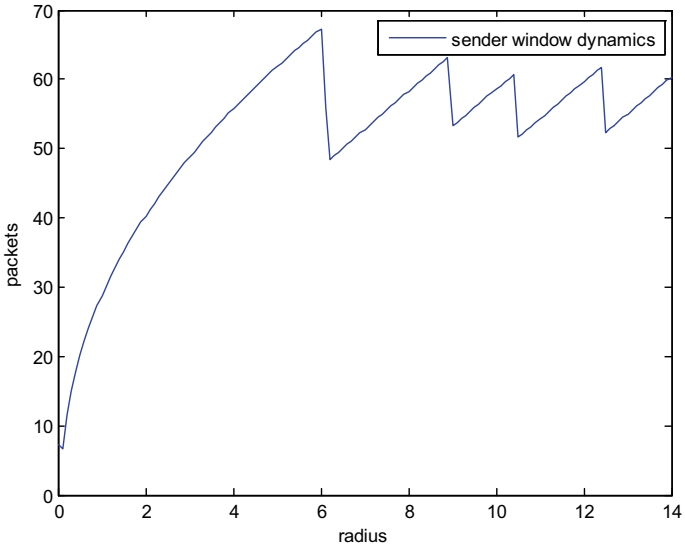


Fig. 22 Dynamics of sender window versus radius

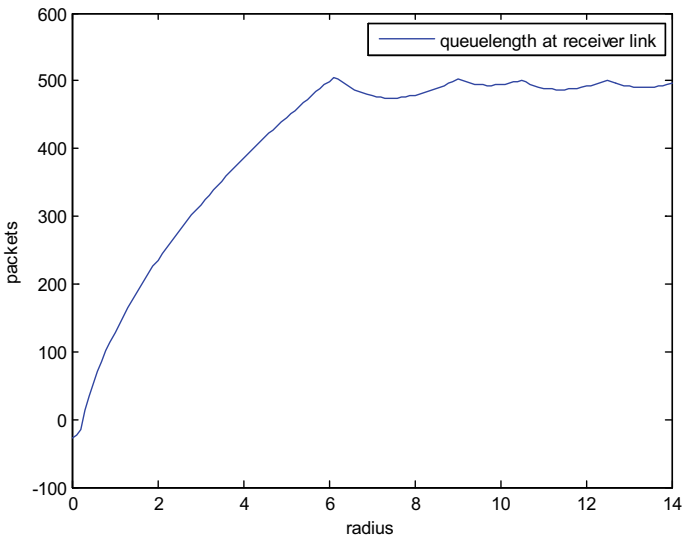


Fig. 23 Queue length at receiver MS link versus radius

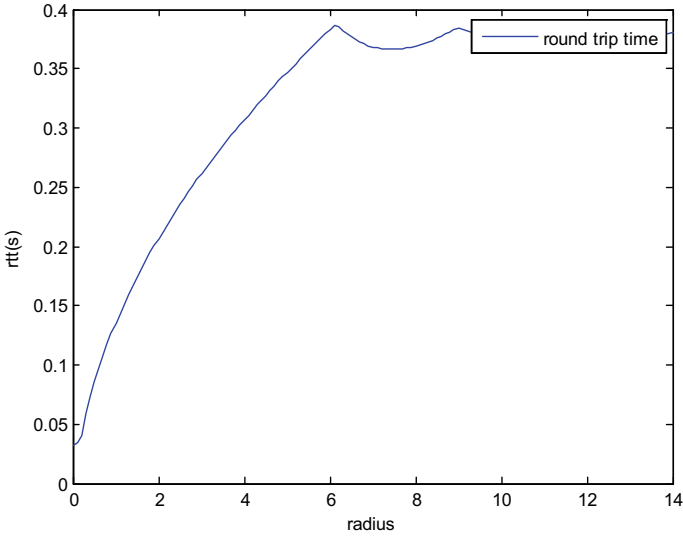


Fig. 24 Variation of round-trip delay versus radius

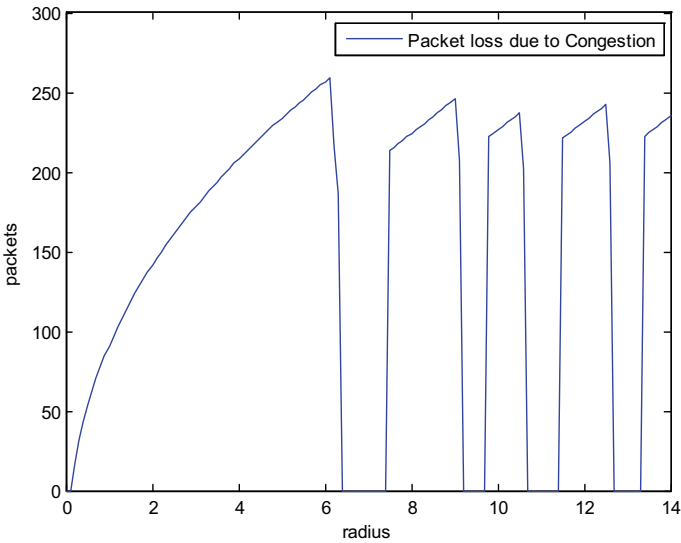


Fig. 25 Packet loss due to congestion versus radius

The packet loss because of congestion is illustrated in the graph of Fig. 25. The main cause of congestion occurrence is due to blocking of handoff calls. The graph of Fig. 26 demonstrates the loss of data packets due to handoff blocking.

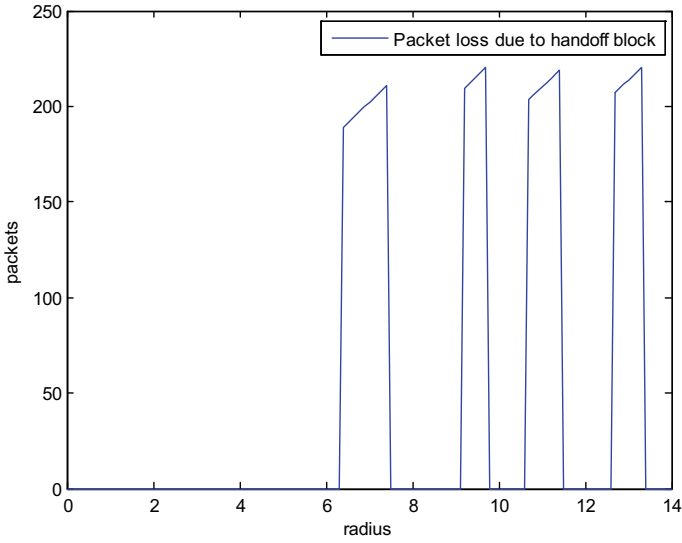


Fig. 26 Packet loss due to handoff blocking versus radius

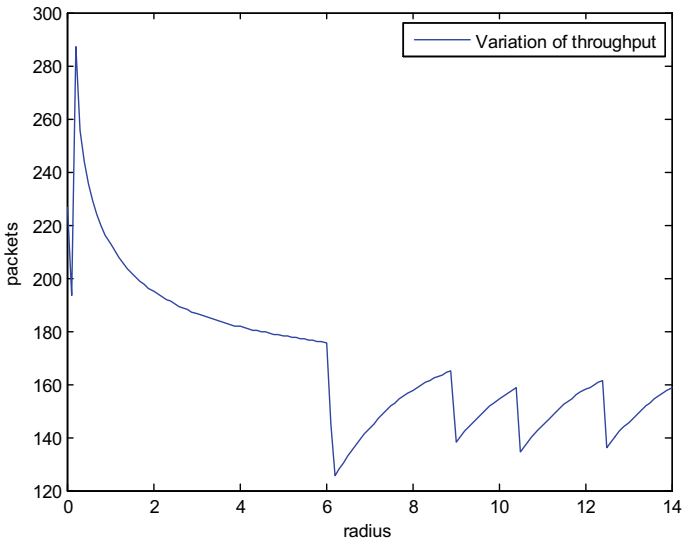


Fig. 27 Variation in throughput versus radius

The performance of average throughput is presented in the graph of Fig. 27. As MS move longer distance away from the BS, the performance of throughput decreases by small amount and fluctuates over the rang of [140, 170] packets.

5 Conclusion

The paper presents a novel research work on handoff management. An important factor that defines packet loss is handoff call blocking. Packet loss during the hard handoff is responsible for deterioration in the quality of service. Data packets that are transported from the old base station till the time MS gets connected to the new BS are lost. The packet loss is also proportional to the handoff loop time. The results discussed faithfully convincing that model-based analysis is one of the best approaches for improving the QoS in mobile wireless communications. Through this analysis, it is possible to minimize many unnecessary handoff request calls, even when the signal of the current base station is still at an acceptable level. This helps in energy saving.

References

1. Zeng, Q. A., & Agrawal, D. P. (2002). Handoff in wireless mobile networks. In I Stojmenović (Ed.) *Handbook of wireless networks and mobile computing*. Wiley & Sons, Inc., ISBN 0-471-41902-8 ©2002.
2. Madan, B. B., Dharmaraja, S., & Trivedi, K. S. (2006). Combined guard channel and mobile assisted handoff for cellular networks. *IEEE Transactions on Vehicular Technology*, 1–9.
3. Boubekri, A., Ajib, W., & Boukadoum, M. (2017). EAM: Energy aware mobility over wireless sensor networks. In *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*. 978-1-5090-5538-8/17/\$31.00 ©2017 IEEE.
4. Trivedi, K. S., Dharmaraja, S., & Ma, X. (2002). Analytic modeling of handoffs in wireless cellular networks. *Information Sciences*, 148(1–4), 155–166. ELSEVIER.
5. Chen, I.-R., Yilmaz, O., & Yen, I.-L. (2006). Admission control algorithms for revenue optimization with QoS guarantees in mobile wireless networks. *Wireless Personal Communications, Springer*; 38, 357–362.
6. Akan, Ö. B., & Baykal, B. (2005). Handoff performance improvement with latency reduction in next generation wireless networks. *Wireless Networks, Springer Science*, 11, 319–332.
7. Bhowmik, B., Pooja, Sarkar, P., & Thakur, N. (2012). Modeling prioritised hard handoff management scheme for wireless mobile networks. *International Journal of Networks and Information Security*, 4(8), 21–32.
8. Liu, T., & Cerpa, A. E. (2014). Data-driven link quality prediction using link features. *ACM Transactions on Sensor Networks*, 10(2), 37.
9. Liu, Q., & Jiang, X. (2017). Per-flow end-to-end delay bounds in heterogeneous wireless networks. In *3rd IEEE International Conference on Computer and Communications*. 978-1-5090-6352-9/17/\$31.00 ©2017 IEEE.
10. Goudru, N. G., & Vijaya Kumar, B. P. (2016). Enhancement of performance of TCP using normalised throughput gradient in wireless networks. *Journal of Information and Computing Science (JIC)*, 11(3), 214–234. (Published by world Academic Press, England, UK).

A Comparative Analysis of Lightweight Cryptographic Protocols for Smart Home



Rupali Syal

Abstract Smart home is an application of Internet of Things. A smart home is a network of smart resource-constraint devices which require secure and confidential transmission of information. Lightweight cryptographic protocols are suitable for IOT applications like smart home. Lightweight cryptographic protocols are characterized by small key size, small block size, and less number of iterations. This paper presents a comparative analysis of lightweight cryptographic protocols in terms of the encryption type, advantages, and resistance to attacks.

Keywords Internet of things · Lightweight cryptography · Smart home

1 Introduction

Internet of Things commonly called as IOT is the network of smart devices which communicate with each other. Within the network, many devices are resource-constrained devices with limited battery, processing power, and memory. These points become weak points in the IOT architecture which are susceptible to attacks. For maintaining the integrity of information, the need is to transfer crucial information from these smart devices in encrypted format. Lightweight cryptographic protocols encrypt data and are computationally less intensive.

Lightweight cryptographic protocols can be broadly classified as block ciphers, hash functions, message authentication codes, and stream ciphers [1]. Lightweight block ciphers are characterized by small block length, small key size, and less number of iterations. Some lightweight block ciphers are PRESENT, SIMON, SPECK, RC5, TEA, and XTEA. Lightweight hash functions have small message size, small internal state, and output size contrary to conventional hash functions, which have large internal state size and high power consumption. The examples of lightweight hash functions are PHOTON, Quark, and SPONGENT. Message authentication code generates tags based on message and key which serve as authentication token which

R. Syal (✉)

Computer Science and Engineering, Punjab Engineering College, Chandigarh, India
e-mail: rupali@pec.ac.in

© Springer Nature Singapore Pte Ltd. 2019

N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,

https://doi.org/10.1007/978-981-13-5953-8_54

663

can be verified by the recipient [1]. Stream ciphers are state ciphers that operate on plaintext bits and keystream to generate ciphertext stream. The stream ciphers are categorized based on the use of linear/nonlinear feedback shift register. Grain, Trivium, Mickey, and Fruit are the examples of stream ciphers [2].

Lightweight cryptography can be implemented both in hardware and software. The target hardware devices for implementation are RFID and sensors [1]. IBM has launched the world's smallest computer, which has computing power of $\times 86$ chip. It has several hundred thousand transistors. It can perform functions like monitor, analyze, communicate, and act on data [3]. The future smart devices will have such tiny computer embedded, which can perform software encryption. For hardware implementation of lightweight ciphers, the NIST report specifies the resource requirement in terms of gate equivalents and logic blocks for ASIC and FPGAs, respectively. For software implementations, the resources used are in terms of registers, RAM, and ROM.

This paper presents a brief survey of various lightweight cryptographic protocols in Sect. 2. Section 3 presents the various levels of security required in a smart home scenario. This section also discusses the various ways for providing security at a particular level. Since IOT architecture of smart home has various resource-constraint devices, therefore, lightweight cryptographic protocols are suitable for hardware/software implementation in IOT. Section 4 presents comparative analysis of lightweight cryptographic protocols in terms of encryption type, advantages, and resistance to attacks. Section 5 gives the conclusion.

2 Related Work

NIST has presented a report on standardization of lightweight cryptographic algorithms. The authors have surveyed various block ciphers, hash functions, and stream ciphers for implementation on resource-constraint devices. The metrics for hardware and software are discussed. A questionnaire is presented at the end which considers various aspects of lightweight cryptographic protocols [1].

Lightweight cryptography is required for secure and efficiency in end-to-end communication in applications of IOT. The complexity and overhead associated with lightweight cryptography is less compared to conventional cryptographic techniques. The authors [4] have concluded that block ciphers have a practical use in the field of IOT. At present, an ultra-lightweight block cipher is suitable for extremely constrained environments like RFID tags, smart cards and sensor networks [5]. It is based on substitution and permutation. PICO [6] is an ultra-lightweight block cipher based on substitution and permutation network that operates on 64-bit plaintext and has key length of 128 bits. It has compact key scheduling like SPECK.

Block ciphers SIMON and SPECK are lightweight ciphers with flexible block and key sizes. They support block sizes 32, 48, 64, 96, and 128 bits and three key sizes with each block size [7]. Simon is a Feistel-based block cipher which requires three XOR operations in each round. LiCi, an ultra-lightweight block cipher is based

on Feistel network. It has 31 rounds and operates on 64-bit plaintext and 128-bit key. The data complexity of LiCi is 2^{64} . It has 52 and 48 active S boxes in linear and differential trails, respectively [8]. The design of S box is robust as single bit change in input does not result in single bit change in output.

Feistel based block ciphers apply round function to only one half of input block, whereas substitution permutation-based block ciphers apply slightly complex round function. The latter can be implemented with fewer rounds as compared to more executions of round function in former to maintain security [9].

A tiny encryption algorithm, TEA based on Feistel iteration and large number of rounds was presented in [10] where the authors presented the C code for TEA. Encryption and decryption in TEA use mixed algebraic group operations. A fast TEA algorithm for embedded and mobile devices which minimize the memory requirements were presented in [11].

Stream ciphers are based on Vernam cipher and use a random keystream based on their internal state. They are categorized into synchronous and self-synchronous ciphers based on the updated operation of internal state. Based on feedback operation, the stream ciphers are categorized into linear-feedback shift registers (LFSR) and feedback with carry shift registers. The authors [12] have discussed alternative ciphers that use simple operations like add, rotate and XOR operations which result in fast, efficient, and timing attack-resistant implementations.

Lightweight ciphers were implemented on 8-, 16-, and 32-bit processors to evaluate the execution time, RAM footprint, binary code size, and cycle count [13]. Lightweight Encryption Scheme (LES) [14] employs the technique of identity based and stateful encryption. The authors state that the scheme provides flexibility in encryption key management and efficiency in encryption process.

Security is required at all layers of IOT architecture [15]. At physical layer, it includes RFID and wireless sensor security. At communication layer, it requires secure transmission over network. At application layer, the software must be protected from malwares. The authors [15] have presented the advantages and disadvantages of various lightweight ciphers.

3 Smart Home: An Application of IOT

A smart home is a home with many smart devices like smart refrigerators, smart washing machines, smart light control systems, smart security mechanisms, and other electronic devices, which communicate with each other. These devices can be controlled by members of the family from within the home or through a remote location with a smart phone or a laptop [16]. The smart devices can be controlled either through voice, remote control, tablet or smartphone. The communication between smart devices within a home also raises the issue of confidentiality, integrity, and authenticity.

IOT can provide an efficient and smarter security mechanism for the smart home [17, 18]. The enhancement to various levels of security is as follows:

- (a) **Physical security**
Smart door locks are physically accessible to anyone. The inbuilt sensors in door locks can capture the fingerprint of invader and transmit to central monitoring home system. Based on the data obtained from fingerprints, an alarm can be sent to home members about tampering of door lock. Physical access to other home devices can be controlled either by fingerprint matching or through voice control.
- (b) **Logical Security/ Access to hardware or data**
The smart resource-constraint devices must be protected with a firewall or gateway which has strong security mechanism. The authentic users are allowed to control smart devices remotely. User behavior must be analyzed and predictive mechanisms must be implemented in gateway/firewall.
- (c) **Message confidentiality**
The message sent from smart device must be encrypted for secure transmission. Since the smart devices are resource constrained, therefore, lightweight ciphers will be more suitable for more efficiency.

4 Comparative Analysis of Lightweight Algorithms

This section presents the comparative analysis of lightweight cryptographic protocols (LWCP). The LWCP can be implemented on hardware and software. Lightweight ciphers based on identity-based encryption, SP network, Feistel structure, and stream cipher are compared based on encryption type, advantages, and resistance to attacks in Table 1.

4.1 Ciphers

4.1.1 Identity-Based Encryption

The identity of smart devices is represented by string which acts as a public key. A private key is generated for the public key. Though the scheme does not require a certificate but requires pairing and is computationally more expensive.

4.1.2 Substitution Permutation Network

PRESENT is a block cipher with block size 64 bits and key size 80 bit or 128 bit. It has 31 rounds and each consist of XOR operation for round key, SBoxLayer, and p Layer.

Table 1 Comparative analysis of lightweight ciphers

References	Lightweight encryption type	Advantages	Attacks
Lightweight Encryption Scheme [14]	Combines identity based and stateful Diffie–Hellman encryption	Flexibility in encryption key management Efficiency gains for encryption process Does not require complex certificate handling	Indistinguishable encryption against chosen plaintext attack
PRESENT [5, 9, 19]	64-bit block- 80 and 128-bit keys 31 rounds Substitution Permutation Network 4-bit S boxes	One of the first lightweight ciphers Efficient on hardware platform as it uses fully wired diffusion layer	Susceptible to side channel attacks, Related key attack, Equivalent key attack, Differential attack
PICO [6]	Ultra-Lightweight Block Cipher Substitution Permutation Network Operates on 64-bit plaintext supports key length 128 bits 32 rounds	Large number of Active S boxes in fewer rounds Robustness and Avalanche effect due to very strong substitution layer Strong and Compact Key Scheduling Good Performance on Hardware and Software	Thwart Linear and Differential attacks Good resistance against Biclique attacks
LiCi [8]	Ultra-Lightweight Block cipher Feistel-Based Network Operates on 64-bit plaintext with 128-bit key length and generates 64-bit ciphertext 31 rounds	More number of active S boxes in minimum number of rounds Good Performance on Hardware and Software	Resist Linear and Differential attacks Good resistance against Biclique and Zero correlation
TEA [9, 10, 11]	Feistel structure Takes 32-bit words. 64-bit input divided into two parts 128-bit key divided into four parts 64 rounds Does not use S boxes	Simplicity Ease of implementation	Susceptible to equivalent key attack and related key attack
FRUIT [2]	Synchronous Stream Cipher Use Linear and nonlinear Feedback shift register	Efficiency in terms of hardware constraints	Relatively secure to Related key attacks, Cube attack, algebraic attack and Weak key IV (initial vector)

4.1.3 Feistel Network

TEA, a tiny encryption algorithm assumes 32-bit words. It is based on Feistel structure and has large number of rounds. The key is divided into 4 vectors and data is divided into 2 vectors each 32-bit words. It takes a key schedule constant and the cipher runs for 32 rounds.

4.1.4 Stream Cipher

FRUIT an ultra-lightweight cipher works on 80-bit register with 43-bit LFSR and 37-bit NFSR, 80-bit key and 70-bit public initial vector (IV) [2].

5 Conclusions

A smart home requires a lightweight, robust, and secure cryptographic protocol for confidential and integral communication between smart resource-constraint devices. A trade-off exists between simplicity (in terms of block/key size, number of iterations) and resistance to attacks. It is imperative for the lightweight protocol to thwart attacks so that the smart home devices work as per the directions given by the family members. This paper presents a comparative analysis of lightweight ciphers in terms of cipher type, their advantages, and resistance to attacks. PRESENT was one of the first lightweight ciphers. TEA is simple but susceptible to related and equivalent key attack. Block ciphers like LiCi, PICO, and stream cipher FRUIT have good resistance against attacks.

References

1. McKay, K., Bassham, L., Turan, M. S., & Mouha, N. (2017). *Report on lightweight cryptography*. NISTIR 8114.
2. Bhasin, A., & Mishra, G. (2016). Recent advances in lightweight stream ciphers. *Special issue REDSET 2016 of CSIT 2016*, pp. 1–4.
3. <https://mashable.com/2018/03/19/ibm-worlds-smallest-computer/#om8X3Zzhugqp>. Accessed on March 24, 2018.
4. Katagi, M., & Moriai, S. (2008). *Lightweight cryptography for the internet of things*. Sony Corporation.
5. Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., et al. (2007). PRESENT: An ultra-lightweight block cipher. In *Cryptographic Hardware and Embedded Systems, CHES2007* (Vol., 4727, pp. 450–466) Springer.
6. Bansod, G., Pisharoty, N., & Patil, A. (2016). PICO: An ultra lightweight and low power encryption design for ubiquitous computing. *Defence Science Journal*, 66(3), 259–265.
7. Nithya, R., & Kumar, D. S. (2016). Where AES is for internet, SIMON could be for IOT. *Procedia Technology*, 25, 302–309.

8. Patil, J., et al. (2017). LiCi: A new ultra-lightweight block cipher. In *ICEI* (pp. 40–45).
9. Hatzivasilis, G., Fysarakis, K., Papaefstathiou, I., & Manifavas, C. (2017). A review of lightweight block ciphers. *Journal of Cryptographic Engineering*, 1–44.
10. Wheeler, D. J., & Needham, R. M. (1994). TEA a tiny encryption algorithm. In *Fast Software Encryption* (LNCS, 1008, pp. 363–366). Springer.
11. Hunn, S. A. Y., et al. (2012). The development of tiny encryption algorithm (TEA) crypto-core for mobile systems. In *ICEDSA* (pp. 45–49).
12. Manifavas, C., Hatzivasilis, G., Fysarakis, K., & Papaefstathiou, Y. (2016). A survey of lightweight stream cipher for embedded systems. *Security Communications Network* (Wiley), 1226–1246.
13. Dinu, D., et al. (2015). *Triathlon of lightweight block ciphers for the internet of things*. Cryptology ePrint Archive Report.
14. Al Salami, S., Baek, J., Salah, K., & Damiani, E. (2016). Lightweight encryption for smart home. In *11th International Conference on Availability, Reliability and Security* (pp. 382–388).
15. Naru, E. R., et al. (2017). A recent review on lightweight cryptography in IOT. In *International Conference on IT in Social, Mobile, Analytics and Cloud* (pp. 887–890).
16. <https://www.smarthomeusa.com/smarthome/>. Accessed on 28 March 2018.
17. Madakam, S., & Date, H. (2016). Security mechanisms for connectivity of smart devices in the internet of things. In *Connectivity frameworks for smart devices, computer communications and networks* (pp. 23–41). Springer (Ch 2).
18. <https://www.iotforall.com/iot-physical-security-technology/>. Accessed on March 29, 2018.
19. Singh, S., et al. (2017). Advanced lightweight encryption algorithms for IOT devices: Survey, challenges and solutions. *Journal of Ambient Intelligence and Humanized Computing* (Springer).

Algorithm Study and Simulation Analysis by Different Techniques on MANET



Nithya Rekha Sivakumar and Abeer Al Garni

Abstract This paper provides a protocol in ROUGH method which governs APBMAN method and FLOODING method to manage the route request packets on Fisheye State Routing in Grid. ROUGH method finds the runner node set by the discovery of the similarity relation between the one-hop and two-hop neighbors. In this paper, comparison is made with the results of three techniques such as ROUGH method, FLOODING method, and APBMAN method in Grid FSR protocol. It is found that ROUGH method has shown improved performance in several important parameters like Throughput, energy consumption, Packet Delivery Ratio, Delay, Overhead, and Normalized Overhead with respect to Pause time. Certainly, ROUGH method is good with effect to Speed in Average Consumed Energy, Total Consumed energy, Packet Delivery Ratio, and Throughput. But, there is a certain increase in Delay as there is also an increase in Speed. Delay is well decreased in FLOODING method.

Keywords Weighted rough set (ROUGH) · Propagation neighborhood information algorithm (FLOODING) and probabilistic broadcasting algorithm (APBMAN) · Speed · Pause time

1 Introduction

The well-known Fisheye State Routing (FSR) protocol [1, 2] determines a route when no route exists or route breaks. To reduce flooding, it is necessary to create a new path from source to destination. This paper provides a protocol in Weighted Rough Set model (ROUGH method), which regulates APBMAN method and FLOODING

N. R. Sivakumar (✉)

Department of Computer Sciences, College of Computer and Information Sciences,
Princess Nourah Bint Abdulrahman University, Riyadh 11671, Kingdom of Saudi Arabia
e-mail: rekhasiva24@gmail.com

A. Al Garni

Department of Information Technology, College of Computer and Information Sciences,
Princess Nourah Bint Abdulrahman University, Riyadh 11671, Kingdom of Saudi Arabia

© Springer Nature Singapore Pte Ltd. 2019

N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_55

671

method to regulate the route request packets in Grid FSR. ROUGH method discovers the runner node set by finding the similarity relation among the one-hop and two-hop neighbors [3]. ROUGH method deliberates the object (node) importance also and this method gives better results compared with FSR in Grid. In this research, comparison is made with the results of ROUGH method in Grid Fisheye State Routing (FSR) protocol with APBMAN method and FLOODING method. By determination, it is found that ROUGH method has improved concert in several important parameters like Throughput, Energy Consumption, Packet Delivery Ratio, Delay, Overhead, and Normalized Overhead with respect to Pause time. Also, another comparison is made with the same parameters with respect to Speed.

2 Related Work

2.1 Probabilistic Flooding with FSR Protocol in Grid

The concepts highlighted in this paper [3–5] with the great need for a new broadcasting strategy that can dynamically adjust the broadcast probability to take into account the current state of the node in two hops in order to ensure a certain level of control over rebroadcasting, and thus help to improve reachability and saved rebroadcasts.

The easy way to implement by guaranteed message dissemination is the straightforward broadcasting. In this scheme, a source broadcast messages to every neighbor who, in turn, rebroadcasts received messages to its neighbors [6]. This procedure lasts, until all reachable nodes have received and rebroadcast the message once. In this scheme, the nodes broadcast the information with a probability P , by getting a broadcast information for the first time, so that all the node has the same probability to rebroadcast the message, regardless of its number of neighbors [7]. In condensed networks, multiple nodes share similar transmission range. Hence, the probability governs the frequency of rebroadcasts and thus could save linkage resources without moving delivery ratios. It should be noticed that in scarce networks, there is much less shared coverage, thus some nodes will not receive all the broadcast messages, unless the probability parameter is high. Another possible area for improving includes investigating the effect of nodes transmission ranges on the rebroadcast probability [5, 8].

2.2 Propagation Neighborhood Information with FSR in Grid

When there is demand, the nodes obtain the routes by using the flooding method to attain the routes. It is reduced evenly also when the flooding is done in the initial stage. The cache and the time out period are used for every route. Here, the time

out value is chosen and the nodes are determined. The nodes are well-found with a small cache to save the routes. However, the variation in the approach comes from the fact that the expiry of the time out period does not trigger an update. After the time out period, the routes are removed from the cache. The primary emphasis of the protocol is on distribution information about the neighborhood. In other words, the neighborhood reflects the entries of routes in the cache. It also checks the information from other neighbour nodes.

2.3 Weighted Rough Set FSR in Grid

The proposed protocol zeroes in reduction of the redundant flooding in Route Request Phase (RREQ) of FSR [3, 9]. In this technique, when there is change in topology, a special hello messages are announced. It carries not only the existing status of the neighbor node but also sends neighbor node attributes. In this to provide the quick response to create new routes, the routing tables are structured within the neighbor nodes. The primary objective in the existing FSR algorithm is more effectively utilized in the present work as follows. The broadcast of discovery packets take place only when needed.

The algorithm of the Improved FSR in Grid RREQ Period and the algorithm for Relative Weight Calculation is explained in my previous research which follows this technique for the below comparison results.

3 Design of Implementation Algorithm Study

3.1 Algorithm 1

- Step 1 All nodes in the network send hello message periodically.
- Step 2 Neighbor nodes receive hello message and update Neighbor table with expire time.
- Step 3 If neighbor timer is expired, nodes checks for Neighbor expire time with current clock time and if current clock time is greater than Neighbor expire time then Neighbor entry is purged from Neighbor Table.
- Step 4 Nodes periodically calculates own probability value using the formula, $\text{Probability} = (\text{Maximum number of Neighbors} - \text{current number of Neighbors}) / \text{Maximum number of Neighbors}$.
- Step 5 Nodes send broadcast message with unique broadcast message id and probability value.
- Step 6 Neighbor nodes receive broadcast message and check message is already received by using message id and sender id.
Or if it is already received, then simply discard the message.

Step 7 If it is not received, then it compares the node probability value with broadcasting probability.

Step 8 If the value is less than broadcasting probability, then it broadcast the message else, discard the message.

3.2 Algorithm 2: Improved Algorithm for ROUGH Model

1. Battery Power, Speed, and Pause time are maintained with all nodes which are inbound and outbound flow of data.
2. The Complete neighbor nodes exchange “hello” messages periodically.
3. The neighbor Table which is with node information receive the above messages also with traffic, Speed and Pause time.
4. In neighbor table, there with one hop information from source node which stores information in neighbor nodes.
5. Now, it initiates path discovery with RREQ to the selective nodes whenever source node likes to transmit data packets.
6. Then, Source node will send RREQ straight without broadcasting if destination to packet is in first hop.
7. Else it selects neighbor node from neighbor table if destination is not found and then will transmit packet.
8. Then, finally by applying approximation with ROUGH method, it now forms defined rules by selecting neighbor nodes.

4 Results by Simulation

Here, the Network Simulator 2 is used to investigate. The simulation was done with 45 nodes around 15 min by placing the nodes randomly over $2000 \times 2000 \text{ m}^2$ with 20 s for Pause time.

4.1 Trials on Effect to Speed

Using the following metrics, the three different algorithms are performed.

4.1.1 Average Consumed Energy (ACE) Versus Speed

ACE is the energy exists within nodes. In Fig. 1, ACE is less in ROUGH method with respect to Speed. ACE in FLOODING method is high than APBMAN method and ROUGH by Speed.

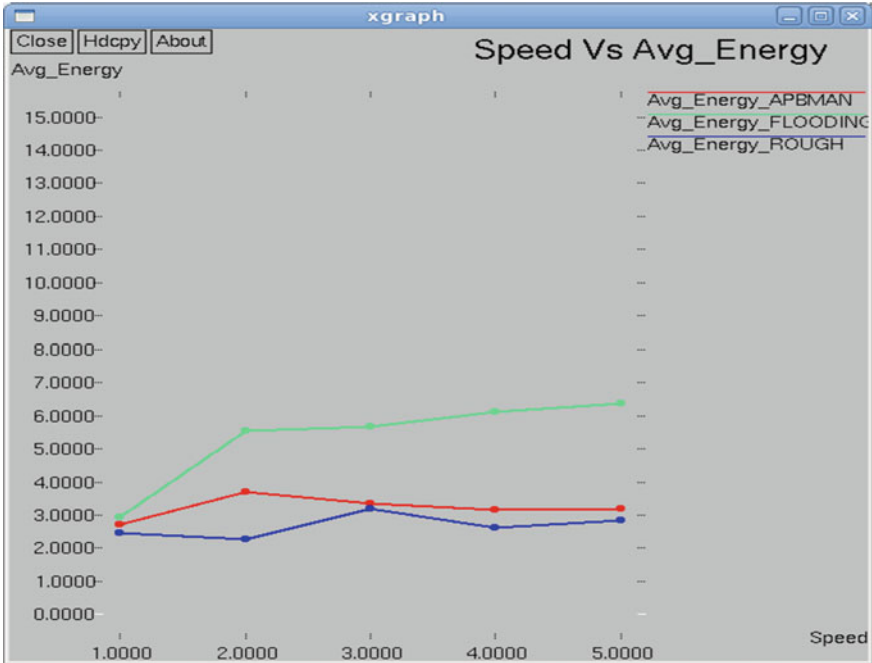


Fig. 1 Speed versus average energy

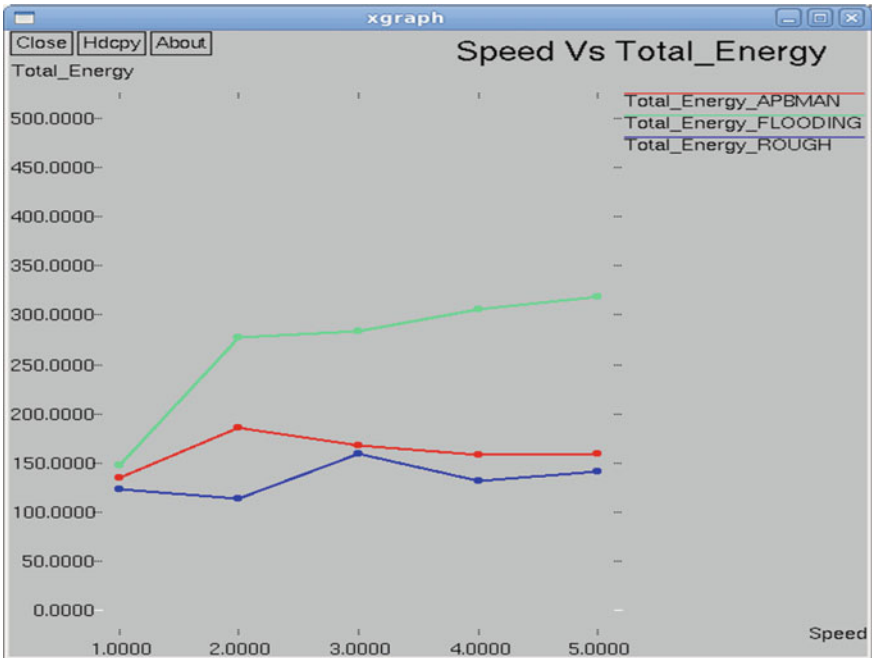


Fig. 2 Speed versus total energy

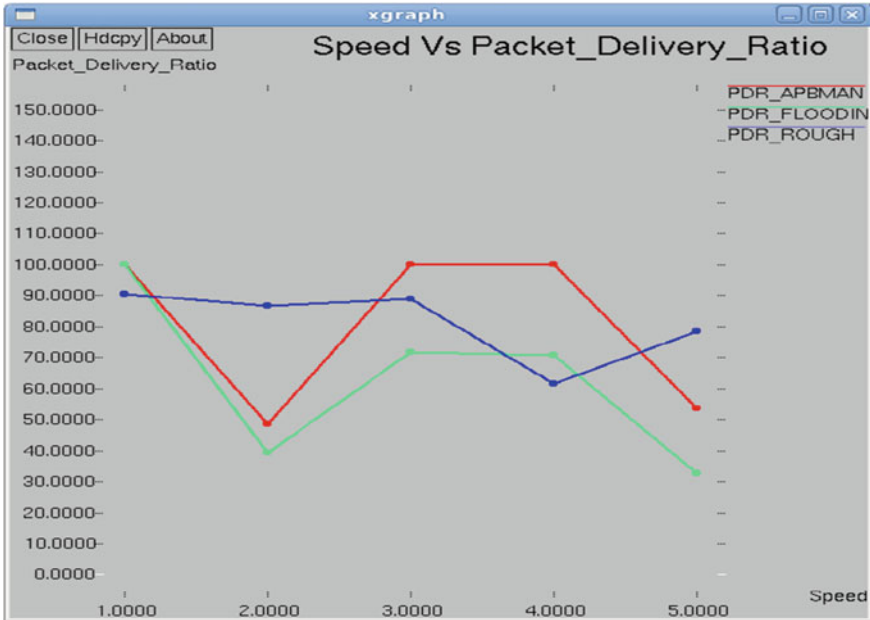


Fig. 3 Speed versus packet delivery ratio

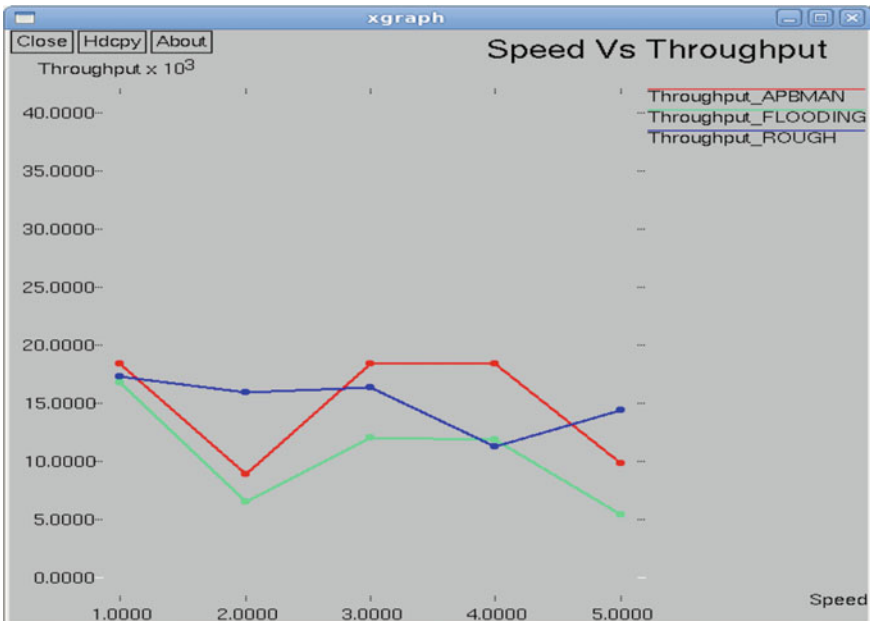


Fig. 4 Speed versus throughput

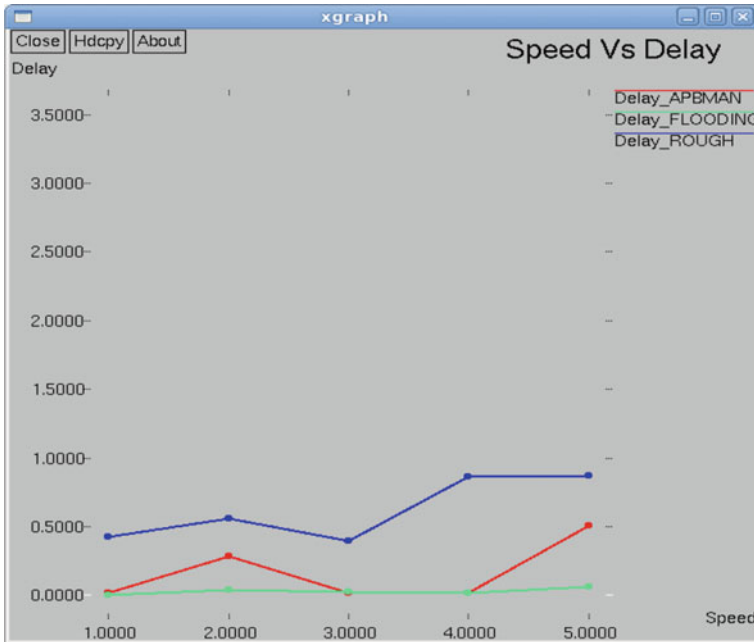


Fig. 5 Speed versus delay

4.1.2 Total Consumed Energy (TCE) Versus Speed

TCE is the energy consumed through the whole network. Figure 2 elucidates that the energy consumed is less in ROUGH method than APBMAN and FLOODING method by Speed.

4.1.3 Packet Delivery Ratio Versus Speed

Packet Delivery Ratio has substantial decrease in APBMAN and FLOODING method as Speed increases. But Fig. 3 elucidates that packet delivery is more effective to certain period in ROUGH method as between 80 and 90% than APBMAN and FLOODING method. Figure 3 shows that a Packet Delivery Ratio result is varied for different mean Node Speeds.

4.1.4 Throughput Versus Speed

This can be said as the average rate of effective delivery of messages on the whole network by all channels. It is shown in Fig. 4 with respect to Speed. Figure 4 shows

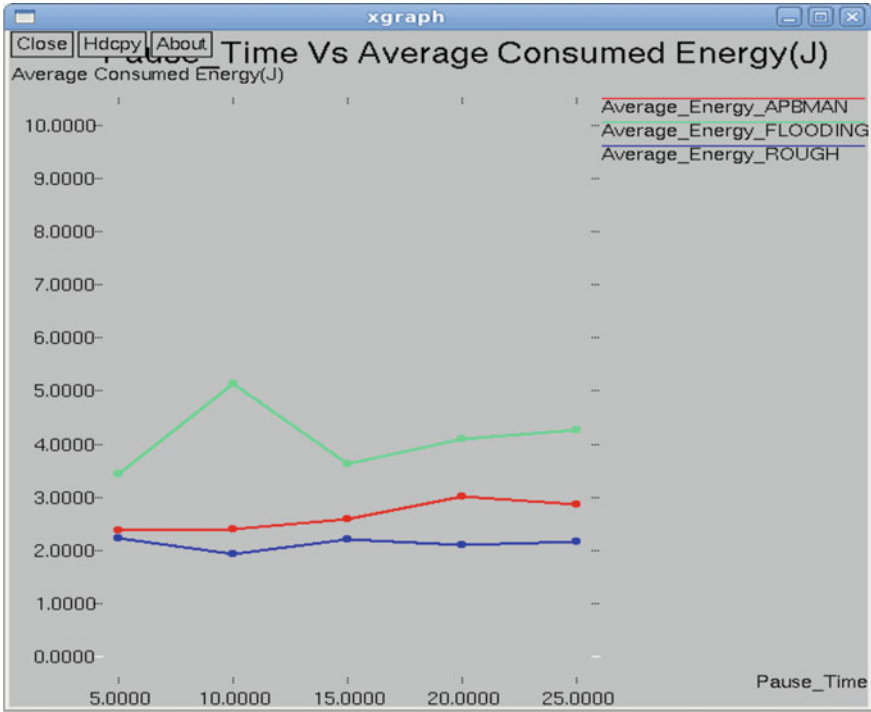


Fig. 6 Pause time versus average consumed energy

that a Throughput result is varied for different mean Node Speeds as proportionally equal to Packet Delivery Ratio in Fig. 3.

4.1.5 Delay Versus Speed

Delay is well decreased in FLOODING in Fig. 5. But it has increased in ROUGH (WRS Model) and APBMAN technique. For each simulation, the Delay has been varied from 1, 2 to 5 m/s.

4.2 Trails on Effect to Pause Time

Using the following metrics, the three different algorithms are performed.

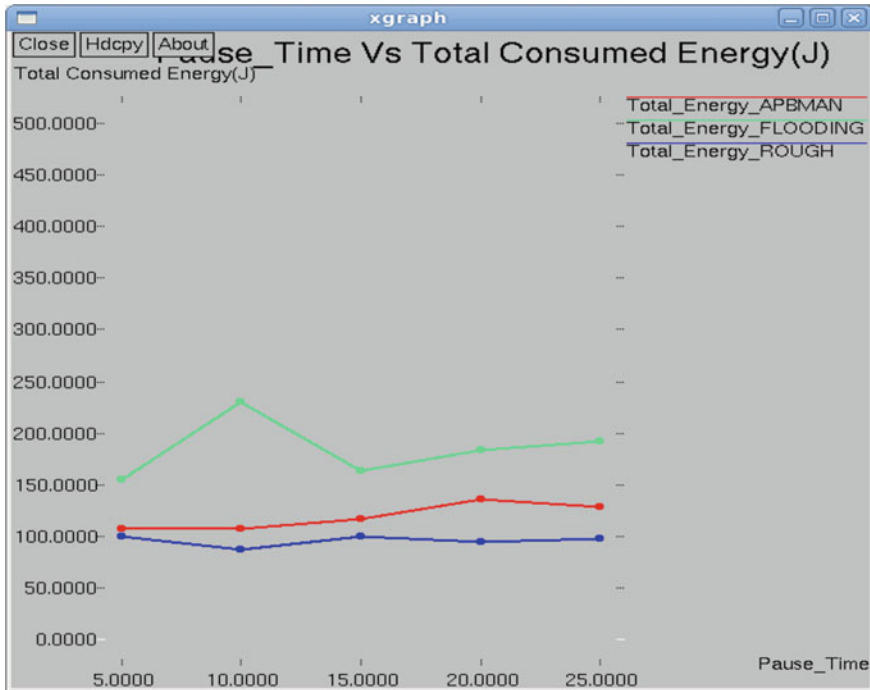


Fig. 7 Pause time versus total consumed energy

4.2.1 Average Consumed Energy (ACE) Versus Pause Time

In Fig. 6, the energy consumed is less in ROUGH method by Pause time. ACE in ROUGH method is reduced than APBMAN and FLOODING method by 20 s Pause time in simulation.

4.2.2 Total Consumed Energy (TCE) Versus Pause Time

TCE is the energy consumed through the whole network. Figure 7 elucidates that the energy consumed is less in ROUGH method than APBMAN and FLOODING method by Pause time.

4.2.3 Packet Delivery Ratio Versus Pause Time

Figure 8 elucidates that packet delivery is more effective in ROUGH method by giving 90% efficiency than APBMAN and FLOODING method.

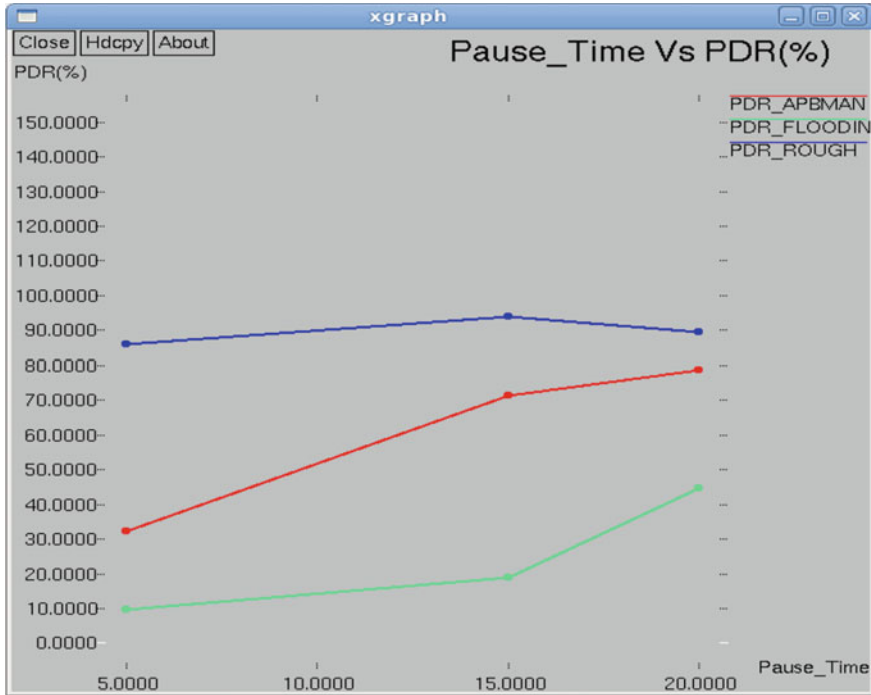


Fig. 8 Pause time versus packet delivery ratio

4.2.4 Throughput Versus Pause Time

Figure 9 elucidates that Throughput with ROUGH method transfers data packets are greater rate than APBMAN and FLOODING method to Pause time.

4.2.5 Overhead Versus Pause Time

Efficient result is produced in ROUGH method than other techniques which have produced the efficient performance with Overhead in Fig. 10. The whole efficacy of MANET routing scheme is determined with the network overhead.

4.2.6 Normalized Overhead Versus Pause Time

Figure 11 clearly elucidates that ROUGH method have been moderately reduced than APBMAN and FLOODING method by Pause time.

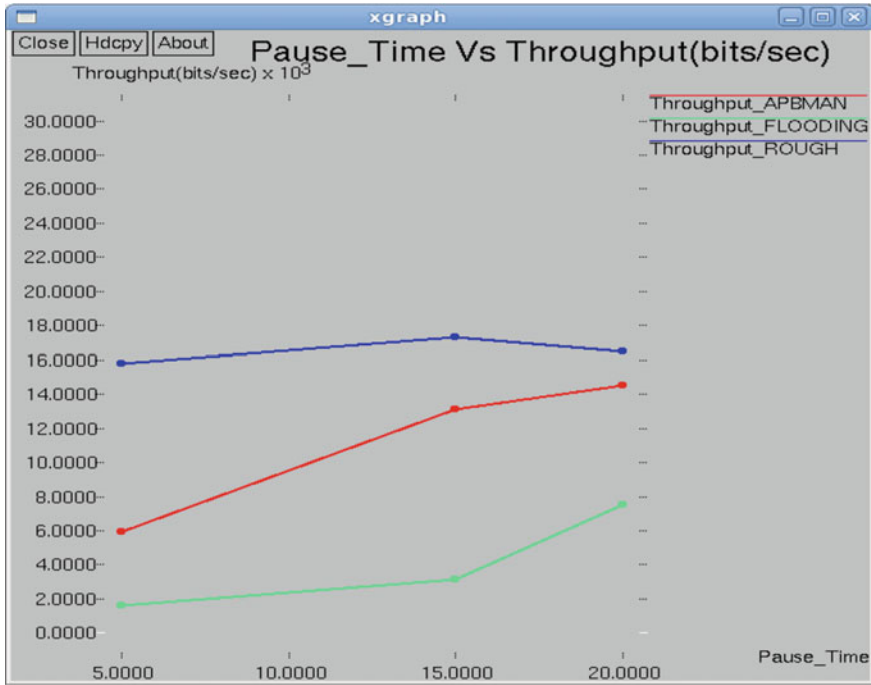


Fig. 9 Pause time versus throughput

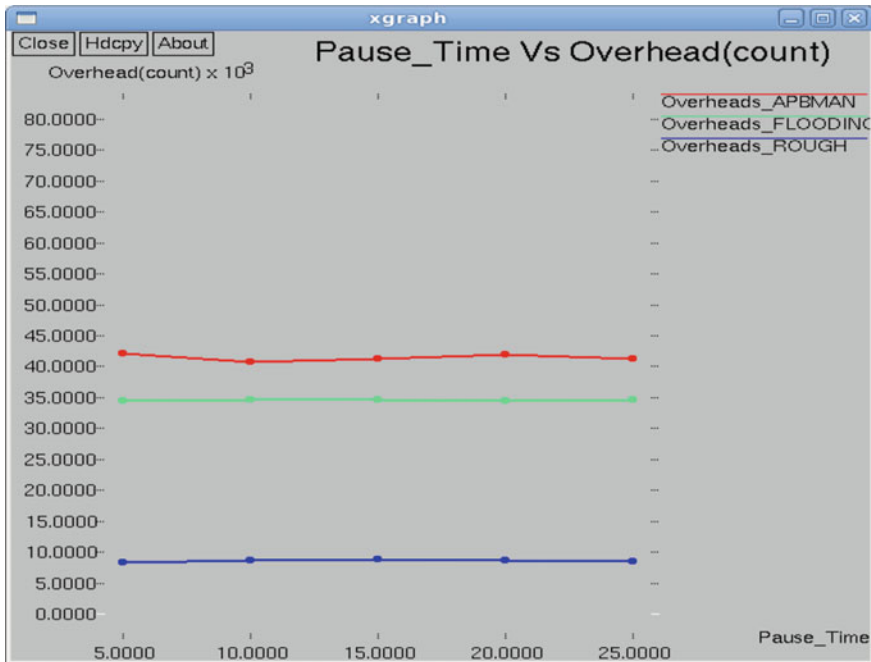


Fig. 10 Pause time versus overhead

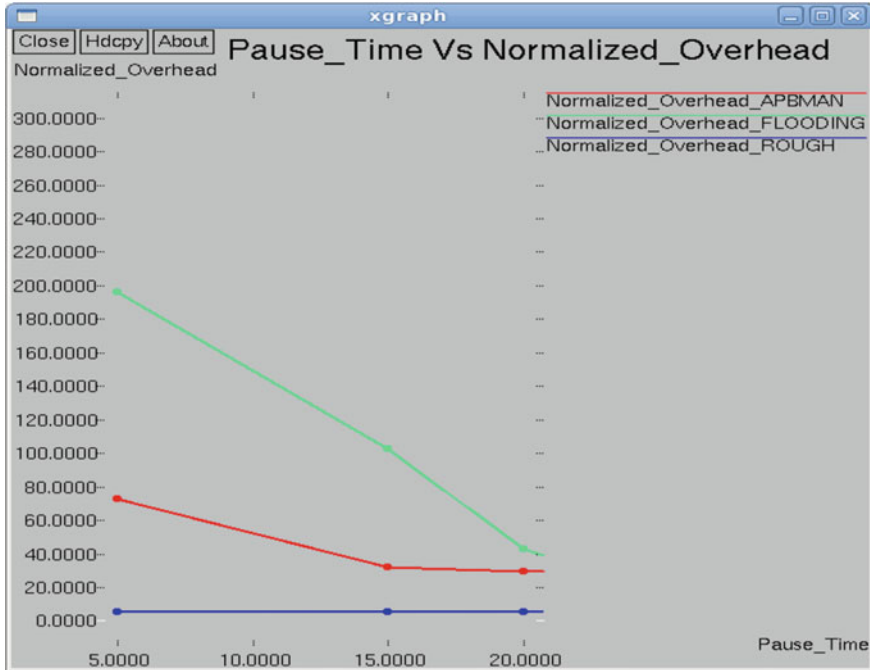


Fig. 11 Pause time versus normalized overhead

5 Conclusion

This paper explains the three different algorithms which are compared and discussed to reduce flooding with different metrics. ROUGH method, here, reduces unwanted route request packets as it has an efficient path through the entire network through the determination from above results. The virtual study between APBMAN method, FLOODING method, and ROUGH method was considered with FSR Protocol in Grid. Efficient growth in ROUGH method with effect to Pause time has been determined using MANET outcomes. Certainly, ROUGH method is good with effect to Speed in Average consumed energy, Total Consumed energy, Packet Delivery Ratio, and Throughput. But there is a certain increase in Delay as there is also an increase in speed. Delay is well decreased in Propagating Neighborhood (FLOODING) method.

References

1. Pei, G., Gerla, M., & Chen, T. W. (2000). Fisheye state routing: a routing scheme for ad hoc wireless networks. In *Proceedings of the IEEE International Conference on Communications, (ICC'00)* (Vol. 1, pp. 70–74). New Orleans.
2. Chu, T. H., & Hwang, S. I. (2006). Efficient fisheye state routing protocol using virtual grid in high density adhoc networks. In *Proceedings of the 8th International Conference on Advanced Communication Technology* (Vol. 3, pp. 1475–1478).
3. Nithya Rekha, S., & Chandrasekar, C. (2012). A strategy to reduce flooding in grid fisheye state routing (GFSR) protocol with weighted rough set method in MANET. *International Journal of Mobile Network Design & Innovation—Inderscience Journal*, 4(4), 192–200.
4. Tseng, Y. C., & Yao, N. S. (2002). The broadcast storm problem in a mobile ad hoc network. In *Wireless Networks* (Vol. 8, pp. 153–167).
5. Colagrosso, M. D. (2007). Intelligent broadcasting in mobile ad hoc networks: Three classes of adaptive protocols. *EURASIP Journal on Wireless Communications and Networking*.
6. Nithya Rekha, S., & Chandrasekar, C. (2012). Performance analysis of probabilistic rebroadcasting in grid FSR for MANET. *International Journal of Computer Science Issues (IJCSI)*, 9(2), 293–300 (Indexed by DBLP & Elsevier, EBSCO, Scirus) International Impact factor: 0.242.
7. Nithya Rekha, S., & Chandrasekar, C. (2012). An improved approach in flooding with packet reachability in FSR (fisheye state routing) protocol using MANET. *Journal of Theoretical and Applied Information Technology (JATIT)*, 40(1), 98–104. (E-ISSN 1817-3195/ISSN 1992-8645) (Indexed by Scopus).
8. Nithya Rekha, S., & Chandrasekar, C. (2012). A comparative analysis of probabilistic broadcasting to reduce flooding with FSR (Fisheye State Routing) protocol and grid FSR using MANET. In *Proceedings of the American Institute of Physics (AIP)—The Sixth Global Conference on Power, Control and Optimization*. Las Vegas, USA (Indexed by Thomson Reuters and SCOPUS, and promoted by EBSCO, Springer09 and Springer10).
9. Nithya Rekha, S., & Chandrasekar, C. (2014). An energy efficient routing to reduce flooding in weighted rough set method using MANET. *International Journal of Communication Networks and Distributed Systems—Inderscience Journal* 13(1), 83–105.
10. Rodoplu, V., & Meng, T. H. (1999). Minimum energy mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 17(8), 1333–1344.

Cloud-Based Agricultural Framework for Soil Classification and Crop Yield Prediction as a Service



K. Aditya Shastry and H. A. Sanjay

Abstract Agriculture is one of the important occupations in India. Digitization in the field of Indian agriculture is in the initial stage. Indian farmers are suffering from various issues such as ignorance about soil parameters and inability to predict the yield of crops. Also, various agriculture-related information from the government agencies is not communicated to the farmers. To address the above-said issues, we have built a cloud-based agricultural framework which enables the Indian farmers, agricultural departments, and agro industries to extract useful agricultural information. The designed agricultural cloud framework is providing two services, i.e., soil classification as a service and crop yield prediction as a service. For soil classification, hybrid support vector machine (M-SVM) and for wheat yield prediction, customized artificial neural network (M-ANN) was developed. To store the agricultural data, we are using Amazon S3 and for deployment of the services, we have used Heroku cloud. The performance improvements in the range of 2–43%, 4–35%, and 1–11% were observed for M-SVM with respect to k-Nearest Neighbor (k-NN), Naïve Bayes (NB), and standard SVM classifiers, respectively. M-ANN performed with an improvement of 2% over standard artificial neural network (ANN) and 5% over multiple linear regression (MLR) models. We also observed that our agricultural cloud framework is able to provide reliable and accurate agricultural services.

Keywords Agriculture · Classification · Prediction · Cloud · Framework

1 Introduction

Agriculture is the primary occupation in India. Yet, in many developing countries, digitization in agriculture is still evolving [1]. Cloud computing services provide a collective pool of resources, computing power, anywhere at anytime [2]. Using cloud computing in agriculture has several benefits [3]. First, agricultural data will be available at anytime from any location. Second, agriculture functions such as crop

K. Aditya Shastry (✉) · H. A. Sanjay
Department of ISE, NMIT, Bangalore 560064, India
e-mail: adityashastry.k@nmit.ac.in

© Springer Nature Singapore Pte Ltd. 2019
N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 882,
https://doi.org/10.1007/978-981-13-5953-8_56

685

yields may be provided to stakeholders. Further, data security is also provided [3]. Data mining is another important area which involves extracting useful information hidden in data [4]. Incorporating data mining analytics in agriculture is beneficial for providing improved crop yields through crop yield prediction [5]. Keeping these points in mind, we have developed a cloud-based agricultural framework for providing soil classification as a service and crop yield prediction as a service. The end users of the framework will be agro industries, government organizations, farmers (landlords), or any other agricultural-related agencies.

Proper soil classification is very much required for crop growth. Classification of soils based on its characteristics is critical to assess the soil quality [6]. Soil classification is performed based on its chemical properties and its texture [7]. Proper soil classification based on chemical properties aids farmers to choose the magnitude of fertilizer and farmyard manure to be applied at various stages of the development cycle of the crop [8]. The soil texture is another important property for agriculture soil classification, which impacts the soil fertility, ability of soil to hold water, exposure of soil to air, tillage of soil, and soil strength [9]. In India, traditional soil classification involving tables, flowcharts are followed. They make use of the standard statistical analysis techniques which consume a lot of time and cost more. In view of this, we have developed an efficient soil classifier based on M-SVM for performing soil classification centered on chemical properties and texture [10]. The end users of soil classifier will be the soil experts who may enter values or read a file. In case of chemical properties, the parameters considered for soil classification are power of hydrogen (pH), electrical conductivity (EC), organic carbon (OC), available phosphorous (AvP), available Boron (AvB), and available potassium (AvK). This data was obtained from ICRISAT [11]. Sand, silt, and clay are considered as parameters for classifying soils based on texture with data obtained from NBSS [12]. This input soil data will then be classified by our proposed soil classifier using M-SVM based on texture and chemical properties of soil.

Crop yield prediction plays a very significant role in agriculture. Accurate prediction of crop yields leads to better harvests [13]. Currently, the primary method of crop yield prediction in Karnataka state is performed using regression models which are not able to capture nonlinear relationships between attributes [14]. Artificial neural network (ANN) is a significantly better model than regression models for performing predictions on continuous data [15]. In this regard, we have developed an M-ANN model in which number of hidden layers; number of neurons in hidden layer and learning rate (LR) are varied [16]. Parameters considered for crop yield prediction are weather parameters such as rain, soil parameters such as soil evaporation, transpiration, and extractable soil water (ESW) and fertilizer parameters such as nitrogen (N). The end users of crop yield prediction software will be agro industries who will have various data available on which prediction can be done. They can pass on this expert information to the farmers. The crop yield prediction has been performed on real-world datasets obtained from [10, 11].

Some of the benefits of deploying the application to the cloud include easy scaling of application resources, cheaper storage, easy disaster recovery, all time availability of application to end users at all times, etc. [17]. To make soil classification and

crop yield prediction available to end users at all times, the soil classifier and crop yield predictor software's have been deployed on Heroku cloud. The datasets of soil, weather, and fertilizer have been stored in Amazon S3. The main advantages of deploying to the cloud are that, the software's are available to all users at anytime, and also secure with proper backup.

The remainder of the paper is structured as follows; Sect. 2 discusses some of the recent and significant works done in the development of cloud frameworks for agriculture. Sections 3 and 4 present the soil classifier and crop yield predictor. Section 5 describes the deployment of the soil classifier and crop yield predictor applications on Heroku cloud. Section 6 describes the results obtained from soil classifier, crop yield predictor and cloud framework followed by conclusion and references.

2 Related Work

In this section, we review some of the significant works done in the development of cloud-based agricultural frameworks. In [18], the authors have built a cloud-based decision support system for orchards. The system provides cloud-based automation from sensor inputs. No data mining has been incorporated. In [19], the authors propose a cloud-based data analytics framework for crop management in India. The authors plan to use machine learning library. It is just a proposal and has not been implemented. Further, they do not discuss in detail which data mining techniques to use and what type of services are provided to end users. In [20], the authors have implemented Internet of Things (IoT) architecture for precision agriculture. Sensors collect data, for reading air temperature, humidity, moisture, leaf wetness, wind speed, and rain volume. The authors have shown how IoT is used in agriculture for sensing data without doing any analysis. In [21], the authors have developed an agricultural autonomic system in cloud which provides information on fertilizer, crop, pesticide weather, equipment, etc. No analysis is incorporated in cloud. In [22], the authors developed a cloud-based system to manage cultivation. It is record management software without any intelligent analysis involved. In most of the works, sensors collect agricultural data and reproduce the same data without any intelligent analysis. That is, no data mining is incorporated in cloud framework for agriculture. Also, most of the cloud agricultural frameworks are proposals without implementations. As per our survey, no cloud framework with data mining algorithms exists in Indian agricultural scenario to analyze the data. In our work, relevant agricultural services such as soil classification and crop yield prediction are provided through data mining algorithms in cloud environment. Hence, our work has combined the benefits of two powerful technologies, i.e., data mining and cloud computing.

3 Cloud-Based Agricultural Framework

Here, we have built soil classification and crop yield prediction systems in the cloud environment for providing soil classification and crop yield prediction as a service. For this purpose, we made use of open-source platform called Heroku cloud platform for deploying cloud applications. The entire coding of soil classification and crop yield prediction was done using MATLAB. This MATLAB code was converted into Java code using MATLAB coder. This conversion was done, as most cloud deployment services are Java friendly. The soil classification application is useful for geological engineers for soil survey as well as agriculturists who are interested in soil types. The crop yield prediction application can be used directly by agro industries to give valuable information to farmers who may be illiterate and not be able to use the software directly. Figure 1 depicts the deployment process followed for deploying soil classification and crop yield prediction on Heroku cloud which contains three phases.

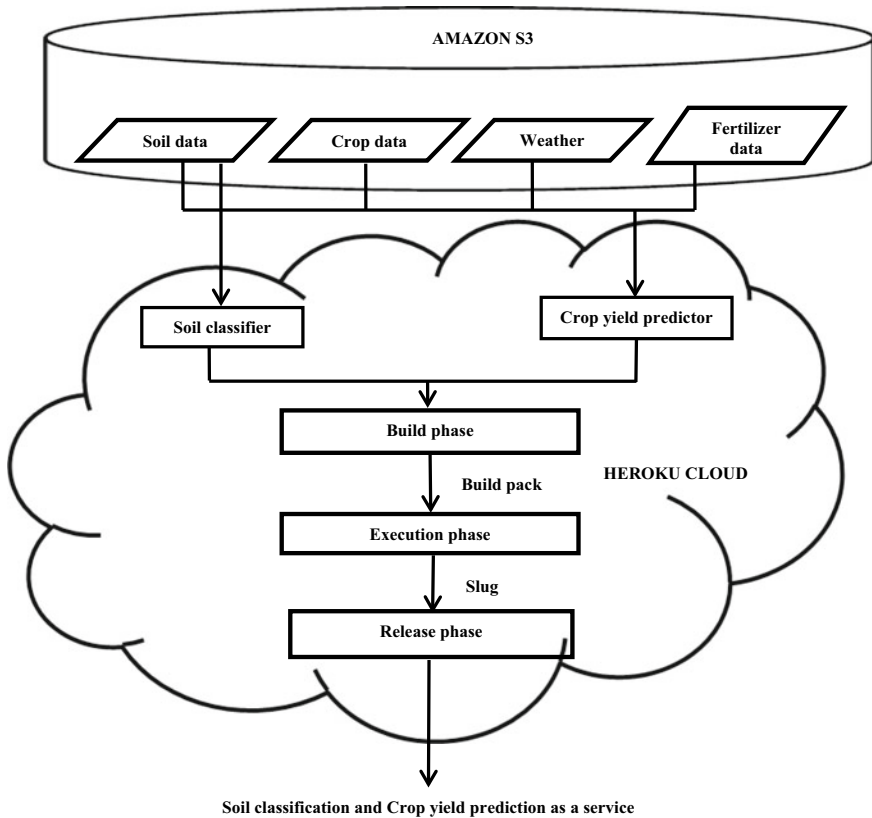


Fig. 1 Agricultural cloud framework

In the first phase, Amazon S3 was used to store the soil data, crop data, weather data, and fertilizer data in the form of buckets. Soil samples containing data about pH, EC, OC, AvP, AvB, AvP, sand, silt, and clay were collected from NBSS [12] and ICRISAT [11]. Crop data contained the data about area sown, production in tones, and historic crop yields of wheat crop from various districts of Karnataka. These were collected from IndiaStat [23]. The weather data contained rain, temperature, and precipitation data collected from India Water Portal [24]. The fertilizer data contained data about nitrogen, phosphorous, and potassium collected from Indiastat [23].

In the second phase, soil classifier and crop yield predictors were developed. The soil classifier user interface contained text fields for the user to enter pH, EC, OC, AvP, AvB, AvP, sand, silt, and clay values. It also contained import file option in which user imports the soil data file in excel or text format. In case of the soil classifier, M-SVM developed in our earlier work [10] was used. It is briefly described in Sect. 4.

In case of crop yield predictor, the user interface contained text fields for user to enter fertilizer information (nitrogen, phosphorous, and potassium), weather information (rain, temperature, and precipitation), soil information (pH, EC, OC, AvP, AvZn, AvB, AvK, and AvS), and crop information (area sown and crop production). Before entering the values, the user has to choose the type of crop. Currently, the system has been implemented to predict crop yields of wheat crop for Karnataka state, India. The front end contains an import file option where the user chooses text file containing fertilizer, soil, and crop information. The prediction was done using the hybrid artificial neural network (M-ANN) developed in our earlier work [16]. It is briefly discussed in Sect. 5.

In the third phase, the actual deployment to Heroku cloud is done. In the build phase, the applications (soil classifier and crop yield predictor) containing source code, dependencies were built using build pack. In the execution phase, a slug is created. Slugs represent compact replicas of applications enhanced for delivery to the dyno manager [25]. This slug was run-on dynos which are isolated Unix containers that offer an environment to run the app [25]. The release phase binds the config, slug, and add-ons.

4 Soil Classifier (M-SVM)

In our earlier work [10], we had developed a hybrid kernel SVM classifier which is briefly described here. The agricultural data is first divided into training and test sets. The tuning of SVM parameters is done for the training set. Here, the kernel parameters C and σ are first selected using genetic algorithm (GA). In this work, using linear, quadratic, RBF, MLP, and polynomial kernels, ten hybrid kernels were developed. The hybrid SVM classifier is then run-on the test set samples. The model is then evaluated using the performance metrics Accuracy, Sensitivity, Specificity, Precision, and F-Score. Gradient descent method is used to select the SVM param-

eters, if performance is poor. Each kernel is then applied for classification. Superior performance was achieved on the real world and benchmark datasets for the hybrid kernel QRK (Quadratic and RBF kernel) for GA selected SVM parameters. Algorithm 1 describes the hybrid kernel SVM for multiclass agricultural datasets.

Algorithm-1 Hybrid Kernel SVM

```

1: procedure M-SVM(TrngSet TstSet)
2:   [TrngInput,TrngCls]=SplitData(TrngSet);
3:   [TstInput,TstCls]=SplitData(TstSet);
4:   Compute[C1,γ1]=GD(TrngInput,TrngCls);
5:   Compute[C2, γ2]=GA(TrngInput,TrngCls);
6:   for k←1 to numClasses do
7:     H- Kernel=[Linear+(Quad/Poly/RBF/MLP)|(Quad+(Poly/RBF/MLP)|Poly
      +(RBF/MLP)|RBF+MLP];
8:     models(k)←TrainSVM(TrngSet,C1|C2,γ1|γ2,H-Kernel);
9:   end for
10:  for j←1 to size(TstSet) do
11:    for k←1 to numClasses do
12:      BinSVMClassify(models(k),TstSet)
13:    end for
14:  end for
15:  [Accuracy,Sensitivity,Specificity,Precision,FS] ←Predict(TstCls,Pred-Class)
16: end procedure

```

The performance of ensemble classifiers is found to be comparatively better than single classifiers in many instances. Similarly, it is found that multiple kernels are more useful when compared to single kernels. This concept called multiple kernel learning (MKL) has two advantages: First, single kernels suffer from bias; while multiple kernels associate the advantages of different kernels. Second, different kernels utilize inputs from different representations having various similarity measures [26]. Data from multiple sources are collated when kernels are combined [26]. Structures which are not apprehended by single kernels are captured by hybrid kernels [26]. In this work, hybrid combinations of SVM kernels are developed and run-on actual and standard agricultural datasets. Experiments indicated that the QRK kernel exhibited superior performance than other hybrid kernels. This kernel is given by Eq. (1)

$$\text{QRK}(a, b) = (\alpha a^T b + m)^2 + \exp(-\gamma \|a - b\|^2) \quad (1)$$

where a, b are inner products and m is an optional constant, T is the transpose.

QRK kernel exploits the benefits of quadratic and RBF kernels which are the extraction of global and local features, respectively.

5 Crop Yield Predictor

In our earlier work [16], we had developed a customized artificial neural network model for wheat yield prediction which is described here. M-ANN is developed by changing N_h (amount of hidden layers from 1 to 2), N_n (quantity of neurons in every hidden layer from 10 to 1000), and learning rate (LR). Best performance was achieved for 2 hidden layers comprising of 60 neurons in the first layer and 30 neurons in the second layer with $LR = 0.20$. The transfer functions used in hidden and output layers were logsig and purelin, respectively. Wheat dataset containing 1674 records was obtained from [23]. After the elimination of redundant, inconsistent and missing values, the dataset contained 1645 records. M-ANN model was trained on 70% of the total records (1151 records). 15% of total records (246 records) were considered as validation set in order to generalize the M-ANN model. Remaining 15% of records (246 records) were utilized as test set to ascertain the accuracy of M-ANN model in terms of R^2 and PPE. M-ANN model achieved a high R^2 and low PPE values when compared to MLR and ANN models which are described in detail in the results section. The Algorithm-2 describes the M-ANN for predicting wheat yield [16].

Algorithm-2 Customized-ANN

- 1: **procedure** M-ANN (*TrngSet TstSet*)
 - 2: [TrngInput,TrngCls]=SplitData(TrngSet);
 - 3: [TstInput,TstCls]=SplitData(TstSet);
 - 4: Use Levenberg Marquardt algorithm for training;
 - 5: Use logsig transfer function for hidden layers and purelin transfer function for the output layer;
 - 6: Customize feed forward back-propagation network by varying the following parameters
 - i) Number of hidden layers (1 to 2)
 - ii) number of neurons in hidden layers (20 to 100)
 - iii) learning rates (0.25, 0.5)
 - iv) network weights (random)
 - 7: [Accuracy, Percentage Prediction Error] ←Perf(TestClass,Pred-Class)
 - 8: **end procedure**
-

6 Experimental Setup and Results

Experimental setup for M-SVM classifier comprised of Windows 7 64-bit operating system possessing 6 GB RAM, with 500 GB hard disk. M-SVM was programmed in MATLAB R2016a. Several trials were performed on four standard agricultural

Table 1 Experimental results of M-SVM soil classifier

Data	Classifier	Accuracy	Sensitivity	Specificity	Precision	F-Score
<i>Benchmark datasets</i>						
Urban land cover	k-NN	35.7	53.93	31.82	14.41	22.75
	NB	62.52	84.27	56.94%	29.41	43.6
	SVM	68.05	86.33	66.99	32.02	44.52
	M-SVM	79.09	88.76	77.03	45.14	59.85
<i>Real world datasets</i>						
Soil texture	NB	51.56	81.25	41.67	31.71	45.61
	k-NN	52.63	85.66	45.23	33.33	47.21
	SVM	59.38	93.75	47.92	37.5	53.57
	M-SVM	68.75	95.75	60.42	44.12	60
pH based soil classification	k-NN	58.49	91.3	26.67	51.11	67.65
	NB	81.13	100	73.3	72.4	80.77
	SVM	96.23	100	93.33	92	95.83
	M-SVM	98.11	100	96.67	95.83	95.87
OC based soil classification	k-NN	75.47	92.31	59.26	68.57	78.69
	NB	81.13	92.31	66.67	73.53	83.33
	SVM	94.34	96.15	96.3	96	94.12
	M-SVM	96.23	96.15	96.3	96.15	96.15
AvB based soil classification	k-NN	73.58	75.86	70.83	75.86	75.86
	NB	90.57	86.21	95.83	96.15	90.91
	SVM	94.34	93.1	95.83	96.43	94.74
	M-SVM	96.23	96.5	95.83	96.55	96.55

datasets from UCI [27] and four real-world datasets from NBSS [12] and ICRISAT [11]. M-SVM classifier was compared with NB, k-NN and default SVM classifiers. NB classifier was run using different distributions (Gaussian, multivariate, and multinomial). The k value in k-NN classifier was varied from 1 to 1000. Experiments using standard SVM were performed using linear, quadratic, polynomial, and RBF kernels. The proposed M-SVM was run using ten hybrid kernels with C and σ values obtained from genetic algorithm. Table 1 depicts the comparison of results obtained from the classifiers.

As can be observed from Table 1, the proposed M-SVM performed superior when compared to NB, SVM, and k-NN classifiers with respect to the performance metrics. For wheat yield prediction using M-ANN, experiments were carried out using MATLAB R2016a on Windows 7 environment, Intel® Pentium® CPU P6200 @2.13 GHz processor with 6 GB RAM and 500 GB Hard Disk. The accuracy of the three models (MLR, D-ANN, and M-ANN) using R^2 statistic and the percentage error (PPE) was measured. Table 2 depicts the prediction results based on R^2 value.

Table 2 Experimental results of M-ANN based on R^2

Prediction models	R^2			PPE		
	Training set (%)	Validation set (%)	Test set (%)	Training set (%)	Validation set (%)	Test set
MLR	100	100	92.52	4.068	7.053	4.196
D-ANN	96	78	95	2.357	2.5533	2.2408
M-ANN	99	90	97	0.6629	0.3968	0.5275

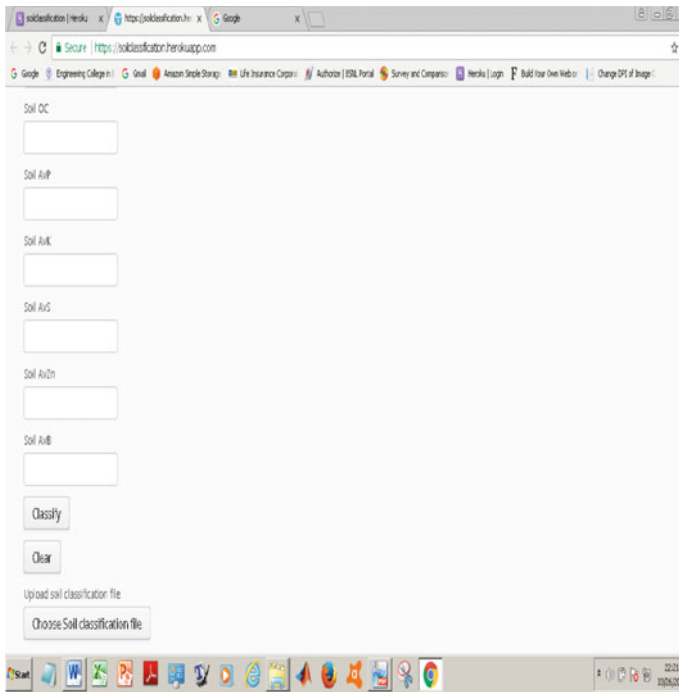


Fig. 2 Soil classification as a service in Heroku cloud environment

Table 2 depicts the comparative results between MLR, D-ANN, and M-ANN based on R^2 and PPE values.

MLR model attained 100% accuracy on training and validation sets. A decrease of 7.48% accuracy was seen on the test set. This was due to the inability of MLR model in grasping the nonlinear association between input parameters. The decrease in performance of D-ANN was observed on validation set as the network failed to generalize. Superior performance was observed for M-ANN model on test set showing improvements of 4.48% over MLR and 2% over D-ANN models. It is observed that the average percentage prediction error for the M-ANN model is the lowest for all the data sets. Hence, the M-ANN model is capable to forecast the yield of wheat better than the MLR and D-ANN models.

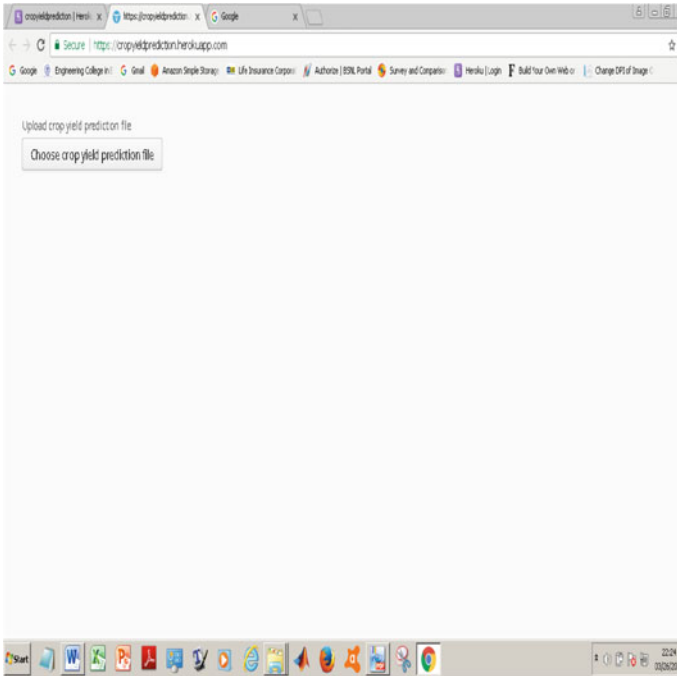


Fig. 3 Crop yield prediction as a service in Heroku cloud environment

With respect to deployment of the soil classification and crop yield prediction applications on Heroku cloud, Heroku cloud infrastructure was used along with Amazon S3. The entire code of MATLAB was converted into Vaadin Java program. The datasets were stored in Amazon S3. Figures 2 and 3 depict the snapshots of soil classification and crop yield prediction provided as a service in Heroku cloud environment, respectively.

7 Conclusion

Though India is progressing in various sectors, agriculture is a sector in which technology is not being completely utilized. In this regard, we have developed an agricultural cloud framework for providing soil classification and crop yield prediction as a service to end users. End users of this framework may be agricultural agencies, government organizations, and farmers (landlords). As per our survey, no cloud service exists for providing soil classification and crop yield prediction as a service to end users. In this regard, we have deployed the soil classification and crop yield prediction applications to Heroku cloud. The main benefits of this include all time available to access the applications from anywhere at anytime. Further, data loss does

not happen as data gets stored in cloud storage and is secure. The soil classification system can be used by soil experts, while the crop yield prediction system can be used by agro industries, which in turn can provide valuable suggestions to farmers. The soil classifier (M-SVM) and crop yield predictor (M-ANN) developed from our earlier works [10, 16] were deployed on Heroku cloud. The agricultural framework was able to provide reliable and accurate services to end users. In future, we plan to develop mobile agricultural apps with sophisticated functionalities.

References

1. Deichmann, U., Goyal, A., & Mishra, D. (2016). Will digital technologies transform agriculture in developing countries? *Agricultural Economics*, 47, 21–33. <https://doi.org/10.1111/agec.12300>.
2. Shawish, A., & Salama, M. (2014). Cloud computing: Paradigms and technologies. In *Inter-cooperative collective intelligence: Techniques and applications* (Studies in Computational Intelligence, Vol. 495, pp. 39–67). Berlin Heidelberg: Springer-Verlag. https://doi.org/10.1007/978-3-642-35016-0_2.
3. Mekala, M. S., & Viswanathan, P. (2017). A survey: Smart agriculture IoT with cloud computing. In *2017 IEEE International Conference on Microelectronic Devices, Circuits and Systems (ICMDCS)* (pp. 1–7). Vellore. <https://doi.org/10.1109/icmdcs.2017.8211551>.
4. Milovic, B., & Radojevic, V. (2015). Application of data mining in agriculture. *Bulgarian Journal of Agricultural Science*, 21, 26–34.
5. Mucherino, A., Papajorgji, P., & Pardalos, P. (2009). A survey of data mining techniques applied to agriculture. *Operational Research*, 9(2), 121–140. <https://doi.org/10.1007/s12351-009-0054-6>. Springer-Verlag.
6. Kibblewhite, M. G., Ritz, K., & Swift, M. J. (2008). Soil health in agricultural systems. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 363, 685–701. <https://doi.org/10.1098/rstb.2007.2178>.
7. Mattsson, Berit, Cederberg, Christel, & Blix, Lisa. (2000). Agricultural land use in life cycle assessment (LCA): Case studies of three vegetable oil crops. *Journal of Cleaner Production*, 8, 283–292. [https://doi.org/10.1016/S0959-6526\(00\)00027-5](https://doi.org/10.1016/S0959-6526(00)00027-5).
8. Karthik, D., Vijayarekha, K., & Manickkam, V. (2014). Land characterizations based on soil properties using clustering techniques. *World Applied Sciences Journal (Data Mining and Soft Computing Techniques)*, 29, 60–64. <https://doi.org/10.5829/idosi.wasj.2014.29.dmsct.11>.
9. Hristov, Biser. (2013). Importance of soil texture in soil classification systems. *Journal of Blakan Ecology*, 16, 137–139.
10. Shastry, K. A., Sanjay, H., & Deexith, G. (2017). Quadratic-radial-basis-function-kernel for classifying multi-class agricultural datasets with continuous attributes. *Applied Soft Computing*, 58, 65–74. <https://doi.org/10.1016/j.asoc.2017.04.049>.
11. Wani, S. P., Sahrawat, K. L., Sarvesh, K. V., Baburao, M., & Krishnappa, K. (Eds.). (2011). *Soil Fertility Atlas for Karnataka, India* (pp. 312). Patancheru 502 324, Andhra Pradesh, India: International Crops Research Institute for the Semi-Arid Tropics. ISBN 978-92-9066-543-4.
12. National Bureau of Soil Survey and Land Use Planning, Soil Data. <https://www.nbsslup.in>.
13. Paul, M., Vishwakarma, S. K., Verma, A. (2015). Analysis of soil behaviour and prediction of crop yield using data mining approach. In: *2015 IEEE International Conference on Computational Intelligence and Communication Networks (CICN)* (pp. 766–771). <https://doi.org/10.1109/cicn.2015.156>.
14. Ramasubramaniam, V. (2005). Forecasting techniques in agriculture. *Journal of the Indian Society of Agricultural Statistics*. IASRI, New Delhi.

15. Stastny, J., Konecny, V., & Trenz, O. (2011). Agricultural data prediction by means of neural network. *Agricultural Economics*, 57, 356–361. <https://doi.org/10.17221/108/2011-agricecon>.
16. Aditya Shastry, K., Sanjay, H., & Deshmukh, Abhijeeth. (2016). A parameter based customized artificial neural network model for crop yield prediction. *Journal of Artificial Intelligence*, 9, 23–32. <https://doi.org/10.3923/jai.2016.23.32>.
17. Savu, L. (2011). Cloud computing: Deployment models, delivery models, risks and research challenges. In *International Conference on Computer and Management (CAMAN)* (pp. 1–4), Wuhan. <https://doi.org/10.1109/caman.2011.5778816>.
18. Tan, L. (2016). Cloud-based decision support and automation for precision agriculture in orchards. In *5th IFAC conference on Sensing Control and Automation Technologies for Agriculture*.
19. Balaji Prabhu, B. V., & Dakshayini, M. (2016). A novel cloud based data analytics framework for effective crop management. *IJCTA*, 9(22), 257–264.
20. Khattab, A., Abdelgawad, A., & Yelmarthi, K. (2016). Design and implementation of a cloud-based IoT scheme for precision agriculture. In *28th IEEE International Conference on Micro-electronics (ICM)* (pp. 201–204). <https://doi.org/10.1109/icm.2016.7847850>.
21. Gill, Sukhpal S., Chana, I., & Buyya, R. (2015). Agri-info: Cloud based autonomic system for delivering agriculture as a service.
22. Murakami, Y., Utomo, S. K. T., Hosono, K., Umezawa, T., & Osawa, N. (2013). iFarm: Development of cloud-based system of cultivation management for precision agriculture. In *2013 IEEE 2nd Global Conference on Consumer Electronics (GCCE)* (pp. 233–234), Tokyo. <https://doi.org/10.1109/gcce.2013.6664809>.
23. IndiaStat, Fertilizer and crop yield data, <https://www.indiastat.com/agriculture>.
24. India Water Portal, Rainfall data, <http://www.indiawaterportal.org>.
25. Heroku, Slugs and dynos, <https://www.logicline.de/en/blog/2015/11/a-more-technical-look-at-heroku-about-procfiles-dynos-and-the-slug/>.
26. Shi, L., Duan, Q., Ma, X., & Weng, M. (2012). *The research of support vector machine in agricultural data classification* (pp. 265–269). Berlin Heidelberg: Springer.
27. Lichman, M. (2013). *UCI machine learning repository*. <http://archive.ics.uci.edu/ml>.