

Springer Proceedings in Mathematics & Statistics

Rajiv Misra  
Nishtha Kesswani  
Muttukrishnan Rajarajan  
Bharadwaj Veeravalli  
Imene Brigui  
Ashok Patel  
T. N. Singh *Editors*

# Advances in Data Science and Artificial Intelligence

ICDSAI 2022, IIT Patna, India,  
April 23 – 24

 Springer

**Springer Proceedings in Mathematics &  
Statistics**

Volume 403

This book series features volumes composed of selected contributions from workshops and conferences in all areas of current research in mathematics and statistics, including data science, operations research and optimization. In addition to an overall evaluation of the interest, scientific quality, and timeliness of each proposal at the hands of the publisher, individual contributions are all refereed to the high quality standards of leading journals in the field. Thus, this series provides the research community with well-edited, authoritative reports on developments in the most exciting areas of mathematical and statistical research today.

Rajiv Misra • Nishtha Kesswani •  
Muttukrishnan Rajarajan • Bharadwaj Veeravalli •  
Imene Brigui • Ashok Patel • T. N. Singh  
Editors

# Advances in Data Science and Artificial Intelligence

ICDSAI 2022, IIT Patna, India, April 23 – 24

 Springer



*Editors*

Rajiv Misra  
Department of Computer Science &  
Engineering  
Indian Institute of Technology Patna  
Patna, Bihar, India

Nishtha Kesswani  
Department of Computer Science  
Central University of Rajasthan  
Ajmer, Rajasthan, India

Muttukrishnan Rajarajan  
Department of EE Engineering  
University of London  
London, UK

Bharadwaj Veeravalli  
Department of ECE  
National University of Singapore  
Singapore, Singapore

Imene Brigui  
EMLYON Business School  
Écully, France

Ashok Patel  
Department of Computer Science  
Florida Polytechnic University  
Lakeland, FL, USA

T. N. Singh  
Director  
Indian Institute of Technology Patna  
Bihar, India

ISSN 2194-1009

ISSN 2194-1017 (electronic)

Springer Proceedings in Mathematics & Statistics

ISBN 978-3-031-16177-3

ISBN 978-3-031-16178-0 (eBook)

<https://doi.org/10.1007/978-3-031-16178-0>

Mathematics Subject Classification: 68TXX

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2023

This work is subject to copyright. All rights are solely and exclusively licensed 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 Switzerland AG  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# Preface

It is our privilege and pleasure to welcome you to the International Conference on Data Science and Artificial Intelligence (ICDSAI) 2022 held on April 23–24, 2022, that **represents key ingredients for the 5th Industrial Revolution. The extensive** application of data science and AI is dramatically changing products and services, with a large impact on labor, economy, and society as well.

ICDSAI 2022, organized by the Indian Institute of Technology, Patna, in collaboration with NITIE, India, and Vkonex AI Research, India, aims at collecting scientific and technical contributions with respect to models, tools, technologies, and applications in the field of modern artificial intelligence and data science, covering the entire range of concepts from theory to practice, including case studies, works-in-progress, and conceptual explorations.

The submissions underwent a rigorous peer-review process by the strong program committee that included experts from all over the world.

We report 29% acceptance rate for regular papers with additional 35% of submissions that were accepted as short articles.

The main conference included over four technical sections that focused on fundamentals of data science applications in mechanical engineering, ML and artificial intelligence (AI), BOT, Web development, app development.

Importantly, this conference basically focused on advanced automation and computational optimization of data science in all technology-based applications, as well as included specific plenary sessions, invited talks, and paper presentations focusing on the applications of intelligent computational algorithms and emerging computational power that have significantly extended the potential of building various intelligent applications.

We would like to acknowledge the many individuals and organizations that have made our conference possible. The hard work and support of the members of the Organizing Committee allowed us to deliver a successful conference program on time.

We are very grateful to the members of the Program Committee who tirelessly and timely reviewed submissions from a broad range of machine learning topics and many application areas.

We also thank all the authors who decided to submit fruits of their labor to ICMLBDA 2022. We very much appreciate your support and your trust in our handling of your manuscripts. We hope that you will consider submitting to ICMLBDA again next year. Last but not least, our thanks go to the publishers Springer PROMS and the International Association of Academicians (IAASSE), for their generous support.

Patna, India  
Ajmer, India  
London, UK  
Singapore, Singapore  
Écully, France  
Lakeland, FL, USA  
Bihar, India

Rajiv Misra  
Nishtha Kesswani  
Muttukrishnan Rajarajan  
Bharadwaj Veeravalli  
Imene Brigui  
Ashok Patel  
T. N. Singh

# Organization

## Program Committee Chairs

2022, ICDLAIR

Misra, Rajiv

Kesswani, Nishtha

Rajarajan, Muttukrishnan

Veeravalli, Bharadwaj

Brigui, Imene

Patel, Ashok

Singh, T. N.

## Reviewers

2022, ICDLAIR

Acharjee, Raktim

Indian Institute of Technology Guwahati,  
Electronics and Electrical Engineering,  
Guwahati, India

Balaramudu, C siva

Baskaradas, James

Chahar, Bhawna

Chaudhari, Prabhanjan

Chauhan, Manorama

Choubey, Dr. Dilip Kumar

Indian Institute of Information Technology  
Bhagalpur, Bihar, India, Computer Science &  
Engineering, Bhagalpur, India

DIXIT, SHIVANI

Garg, Shankey

Gupta, Lakhanlal

Islam, Tariqul

Daffodil International University, Dhaka,  
Bangladesh

J, Andrew

Jagat, Rikhi Ram

National Institute of Technology Raipur,  
Computer Science and Engineering, Raipur,  
India

Kampa, Hananya

Kannan, Hemachandran

LET, G SHINE

M.A, ANIL

Mashekova, Aigerim

Nazarbayev University, School of Engineering  
and Digital Sciences, Nur-Sultan, Kazakhstan

Misra, Rajiv

Mutkule, Prasad

P, Rajalakshmy

Panda, Bishnupriya

Parikh, Vishal

Pravinkumar, Padmapriya

SHARMA, AMITA

SIVASUBRMANIAN,

RAVISANKAR

Saifuzzaman, Mohd.

BJIT (Bangladesh Japan Information  
Technology) Limited, Bangladesh, SQA  
Department, Dhaka, Bangladesh

Shah, Vinita

Shukla, Varun

Singh, Vinay

VERMA, Dr. VINOD

KUMAR

VERMA, MONIKA

Vala, Jaykumar

Vella, Joseph G

sar, sumit

# Contents

<b>Sky Detection in Outdoor Spaces</b> .....	1
Dev Kumar Sahoo, Jennifer Lobo, Sanjana Pradhan, Shagufta Rajguru, and K. Rakhi	
<b>Defining, Measuring, and Utilizing Student’s Learning in a Course</b> .....	9
Tanmay Garg, Rajat Agarwal, Mukesh Mohania, and Vikram Goyal	
<b>Holistic Features and Deep-Guided Depth-Induced Mutual-Attention-Based Complex Salient Object Detection</b> .....	21
Surya Kant Singh and Rajeev Srivastava	
<b>Machine Learning Based Decision Support System for Resilient Supplier Selection</b> .....	33
Saurav Kumar, Anoop Kumar Dixit, and Milind Akarte	
<b>An Adaptive Task Offloading Framework for Mobile Edge Computing Environment: Towards Achieving Seamless Energy-Efficient Processing</b> .....	45
Mohammad Ashique E. Rasool, Anoop Kumar Bhola, Asharul Islam, and Khalid Mohiuddin	
<b>Road Surface Classification and Obstacle Detection for Visually Impaired People</b> .....	57
Shripad Bhatlawande, Yash Aney, Aatreya Gaikwad, Vedant Anantwar, Swati Shilaskar, and Jyoti Madake	
<b>A Survey on Semantic Segmentation Models for Underwater Images</b> .....	69
Sai Krishna Anand, Pranav Vigneshwar Kumar, Rohith Saji, Akhilraj V. Gadagkar, and B R Chandavarkar	
<b>An Interactive Dashboard for Intrusion Detection in Internet of Things</b> .....	87
Monika Vishwakarma and Nishtha Kesswani	

<b>An Analogous Review of the Challenges and Endeavor in Suspense Story Generation Technique</b> .....	99
V. Kowsalya and C. Divya	
<b>Friend Recommendation System Using Transfer Learning in the Autoencoder</b> .....	113
Bhargav Rao and Aarti Karande	
<b>Analysis on the Efficacy of ANN on Small Imbalanced Datasets</b> .....	129
Gauri Naik, Deep Siroya, Manav Nisar, Bhavya Shah, and Himani Deshpande	
<b>Lightweight and Homomorphic Security Protocols for IoT</b> .....	139
Ishaan Singh, Aakarshree Jain, Ikjot Singh Dhody and B R Chandavarkar	
<b>Tool-Based Approach on Digital Vulnerability Management Hub (VMH) by Using TheHive Platform</b> .....	175
V. Ceronmani Sharmila, M. Arvinth Sithartha, Samita Ramesh Babu, and M. Shruthi	
<b>Performance Analyzer for Blue Chip Companies</b> .....	191
Ishita Badole, Sakshi Chheda, Ojasa Chitre, and Dhananjay Kalbande	
<b>Strengthening Deep-Learning-Based Malware Detection Models Against Adversarial Attacks</b> .....	203
Rohit Pai, Mahipal Purohit, and Dr. Preetida Vinayakray-Jani	
<b>Video-Based Micro Expressions Recognition Using Deep Learning and Transfer Learning</b> .....	221
Samit Kapadia, Ujjwal Praladhka, Utsav Unadkat, Virang Parekh, and Ruhina Karani	
<b>Trustworthiness of COVID-19 News and Guidelines</b> .....	233
Shubhanshu Singh, Lalit Nagar, Anupam Lal, and B R Chandavarkar	
<b>Detection of Moving Object Using Modified Fuzzy C-Means Clustering from the Complex and Non-stationary Background Scenes</b> ...	247
Ravindra Sangle and Ashok Kumar Jetawat	
<b>Deterrence Pointer for Distributed Denial-of-Service (DDoS) Attack by Utilizing Watchdog Timer and Hybrid Routing Protocol</b> .....	261
Sandya J. K., Ashwanth S., Aluri Prameela Manyatha, and V. Ceronmani Sharmila	
<b>Modeling Logistic Regression and Neural Network for Stock Selection with BSE 500 – A Comparative Study</b> .....	285
Selvan Simon and Hema Date	



**Landslide Detection with Ensemble-of-Deep Learning Classifiers Trained with Optimal Features** ..... 313  
 Abhijit Kumar, Rajiv Misra, T. N. Singh, and Vinay Singh

**A Survey Paper on Text Analytics Methods for Classifying Tweets** ..... 323  
 Utkarsh Bansod, Dheetilekha Nath, Chanchal Agrawal, Srishti Yadav, Ashwini Dalvi, and Faruk Kazi

**A Survey on Threat Intelligence Techniques for Constructing, Detecting, and Reacting to Advanced Intrusion Campaigns** ..... 341  
 Ashutosh Anand, Mudit Singhal, Swapnil Guduru, and B R Chandavarkar

**Generalizing a Secure Framework for Domain Transfer Network for Face Anti-spoofing** ..... 357  
 B R Chandavarkar, Ayushman Rana, Mihir M. Ketkar, and Priyanka G. Pai

**Survey on Game Theory-Based Security Framework for IoT** ..... 367  
 Pranav Joshi, Suresh Kamediya, Ritik Kumar, and B R Chandavarkar

**Survey: Intrusion Detection for IoT** ..... 377  
 B R Chandavarkar, Joshitha Reddy D., Surla Lakshmi Poojitha, and Reshma Tresa Antony

**Human-in-the-Loop Control and Security for Intelligent Cyber-Physical Systems (CPSs) and IoT** ..... 393  
 Sanjkeet Jena, Sudarshan Sundarrajan, Akash Meena, and B R Chandavarkar

**Survey: Neural Network Authentication and Tampering Detection**..... 405  
 Rahul Kumar, Ashwin P, Bhumik Naveen, and B R Chandavarkar

**Misinformation Detection Through Authentication of Content Creators** . 425  
 Kruthika K Sudhama, Sree Gayathri Siddamsetti, Pooja G, and B R Chandavarkar

**End-to-End Network Slicing for 5G and Beyond Communications** ..... 435  
 Rohit Kumar Gupta, Sudhir Kumar, Praveen Kumar, and Rajiv Misra

**Transparency in Content and Source Moderation** ..... 445  
 Adithya Rajesh C., Chathanya Shyam D., Pranav D. V., and B R Chandavarkar

**A New Chaotic-Based Analysis of Data Encryption and Decryption** ..... 455  
 Fatema Tuj Johora, Alamin-UI-Islam, Farzana Yesmin, and Md. Mosfikur Rahman

**Trust and Identity Management in IOT** ..... 469  
 Lakshmi Aashish Prateek, Riya Shah, Alan Tony, and B R Chandavarkar

**Plant Pest Detection: A Deep Learning Approach** ..... 489  
Nilkamal More, V. B. Nikam, and Biplab Banerjee

**S.A.R.A (Smart AI Refrigerator Assistant)** ..... 499  
Sachin Singh Bhadoriya, Saniya Kirkire, Rut Vyas, Satvik Deshmukh,  
and Yukti Bandi

**A Location-Based Cryptographic Suite for Underwater Acoustic  
Networks** ..... 511  
Thota Sree Harsha, Venkata Sravani Katasani, Rajat Partani,  
B R Chandavarkar

**Index** ..... 523

# Sky Detection in Outdoor Spaces



Dev Kumar Sahoo, Jennifer Lobo, Sanjana Pradhan, Shagufta Rajguru,  
and K. Rakhi

## 1 Introduction

In various applications, sky detection plays a vital role. Outdoor images and details about the environment are very well depicted by the sky. Sky detection also promotes some image augmentation tasks. Many researchers have worked on the problem of sky detection over the past few years. After studying various algorithms and existing systems and understanding the challenges faced by them, this project aims to build a model to detect the pixels in an image which belong to the sky using semantic segmentation. The focus is to develop an algorithm that will be suited to work effectively under different weather and illumination conditions. The developed model will be available to users for inference through a web-based GUI and an API, thus providing an end-to-end solution for the problem statement.

In this paper, Sect. 2 discusses about the research work done in the past and existing technology in use. Section 3 describes about the proposed algorithm. Section 4 gives a detailed count of the datasets that is used to carry the research work. Section 5 highlights on the implementation details, while Sect. 6 delivers the result obtained. Then the organization of this paper is followed by conclusion and future scope in Sects. 7 and 8, respectively. Finally, this paper concludes with the list of references.

---

D. K. Sahoo · J. Lobo · S. Pradhan · S. Rajguru (✉) · K. Rakhi  
Department of Computer Engineering, Fr Conceicao Rodrigues Institute of Technology, Navi  
Mumbai, India  
e-mail: [shagufta.r@fcrit.ac.in](mailto:shagufta.r@fcrit.ac.in)

## 2 Related Work

The work done in the past on the particular topic had yielded enough results. However, the main task still remained to find the perfect sky detection in hazy images and mostly night sky. There had been a lot of work carried out since the problem was detected, and here are few papers that have actually detected the cause and the solution.

The sky segmentation task can be supportive for miscellaneous applications such as stylizing images using sky replacement, obstacle avoidance, etc. Current applications from personal to public use require the need of sky segmentation. They are mainly used in scene parsing [1], horizon estimation, and geolocation. Various other sectors where sky segmentation can be utilized include image editing, weather classification, and weather estimation. It also operates on weather recognition and uses cameras as weather sensors. Detection of weather is also proven useful for image searching [2] where one can explore outdoor scene images in accordance with the weather-associated properties.

For detecting the sky, it uses random forest to yield seed patches for sky and non-sky regions and then uses graph cut to fragment the sky regions. In this paper, it focuses mainly on the images rather than the video. A dataset of about 10000 is taken and trained for getting the results in a proper manner.

## 3 Algorithm

- Step 1: A SkyFinder dataset with 10534 images has been given to the model. The data is stored there. It gives a list of training and testing, and also the cross-validation images are given.
- Step 2: The dataset is split where all the three tasks are performed separately, the training of data, testing of data, and the validation of data. This is the pre-processing part.
- Step 3: The transfer learning is done by ResNet and PSPNet, and the VGG and PSPNet captures the learning data. The model performance analysis is done along with model tuning.
- Step 4: The above step is the processing step that trains the model. Once the model is fully trained, then it goes on to do finding of the sky in the given images.
- Step 5: An image would be given as an input. Then the image would be forwarded for pre-processing from the GUI through the API. The pre-processed input is sent to the model.
- Step 6: The segmented sky image and IoU are source forwarded after inference. Segmented sky image and IoU are then scored forwarded after inference.
- Step 7: The GUI then sends back the segmented sky image to the user. Thus the user then receives an image where the sky is being separated with a different color as explained in Fig. 1.

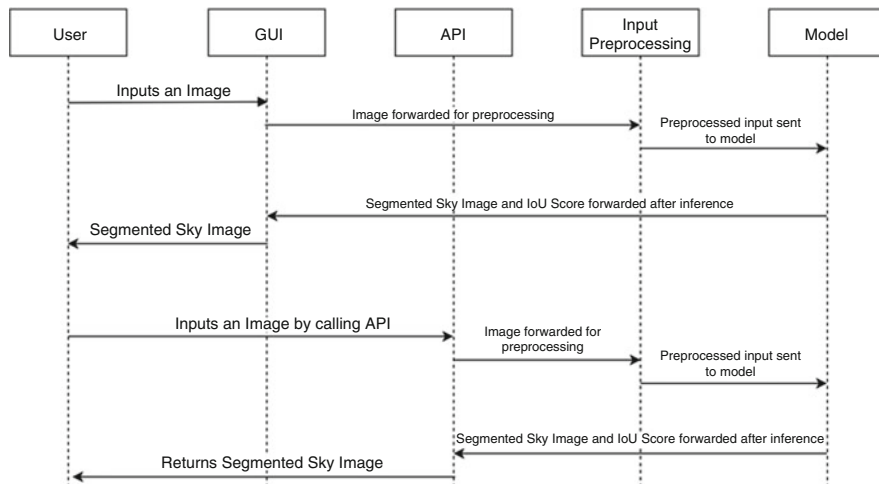


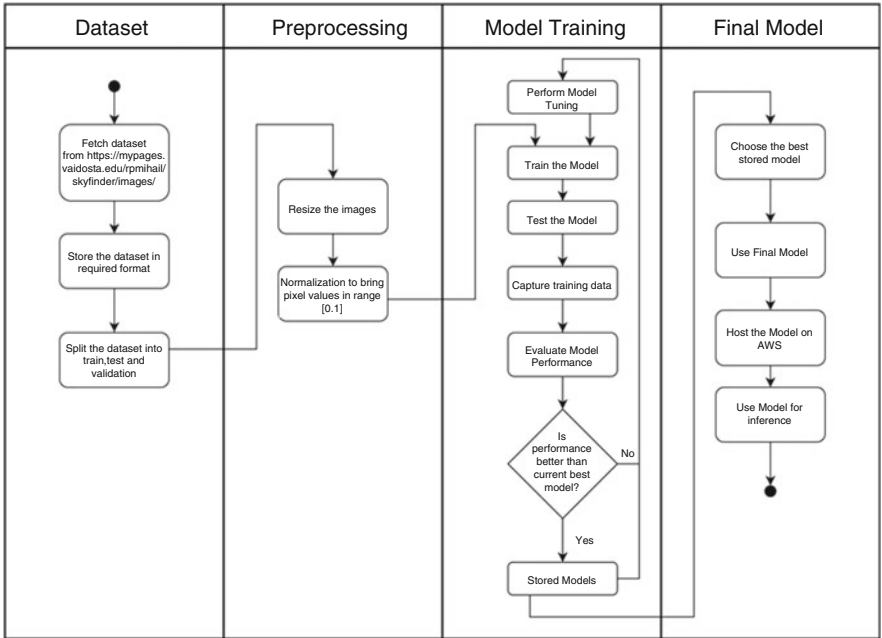
Fig. 1 Sequence diagram of user interaction with sky detection system

## 4 Dataset

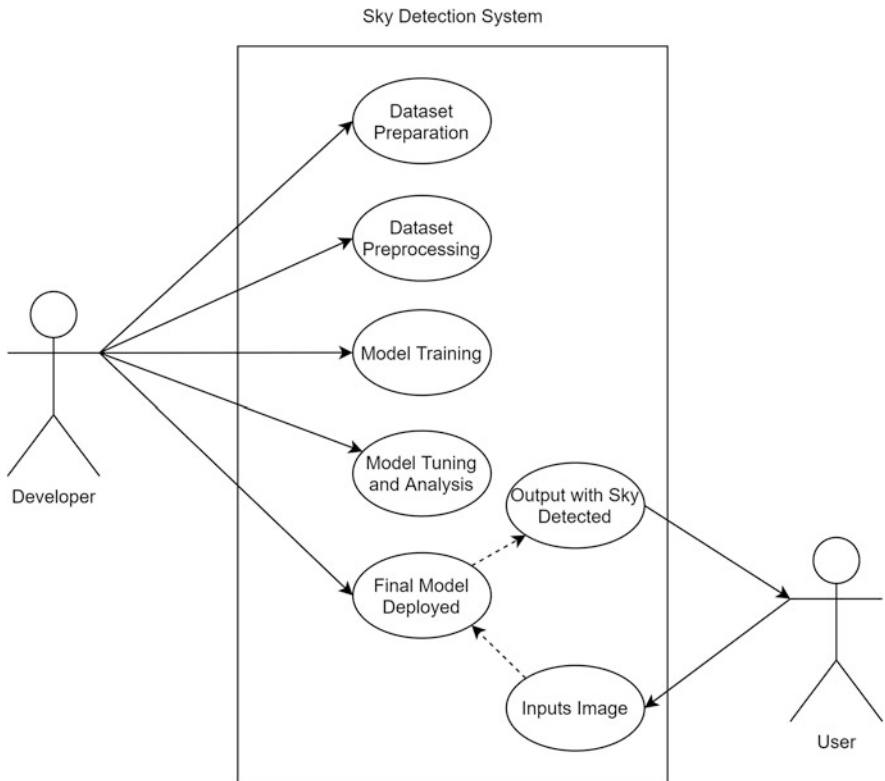
For building the sky detection model [3], images from SkyFinder dataset were gathered. This dataset had around 90,000 outdoor images under varied illumination and weather conditions. The dataset was created by capturing the data from 53 static cameras throughout a period of one or more calendar years. The binary mask segmenting the sky and ground for every camera was created manually. The dataset also provides the associated weather data for all the images in the dataset. The average compass of sky pixels for all the webcams is 41.19%, with standard deviation of 15.71%. The size of the RGB images is  $480 \times 640$  and it has three color channels. Hence, the size of the pixel matrix is  $(307200, 3)$  (Figs. 2 and 3).

## 5 Implementation Details

In this project, the deep learning approach for semantic segmentation uses the Pyramid Scene Parsing Network (PSPNet). The deep learning-based semantic segmentation [4] makes use of an encoder-decoder structure. The encoder comprises several convolution layers, on-linear activation, batch normalization, and pooling layers. The initial layers learn coarse concepts like colors and edges, while the latter layers learn complex concepts. The pooling layer performs down-sampling and decreases the image size while increasing the number of channels. The output of the encoder contains high-level information and is then fed to the decoder as input.



**Fig. 2** Activity diagram of sky detection system



**Fig. 3** Use case diagram of sky detection system

**Table 1** Quantitative analysis of images

Run number	Number of images	Frequency weighted IoU	Mean IoU
1	7898	0.5493	0.5528
2	10534	0.53558	0.5409
3	10534	0.7227	0.7181

The decoder performs up-sampling using the high-level information provided by the encoder and produces the segmentation maps. The method of transfer learning was used for training the model. Transfer learning is a methodology in which a prototype trained on one task is regenerated on another relevant task. The ResNet and VGG pretrained models were used for this purpose in conjunction with the PSPNet. A feature map was created by a deep network like ResNet. The feature map was then passed to the Pyramid Pooling Module. The Pyramid Pooling Module performs pooling at four different scales. The average pooled outputs for the four scales are  $1 \times 1 \times c$ ,  $2 \times 2 \times c$ ,  $3 \times 3 \times c$ , and  $6 \times 6 \times c$  where  $c$  is the number of input channels. The ResNet was used to create the feature map for the Pyramid Pooling Module of the PSPNet. The major issue witnessed by intense CNNs is the problem of convergence due to dispersing/exploding gradients. The residual block of ResNet has a residual [5] connection that carries the layer input  $X$  directly to the addition operator, which results in  $F(X)+X$  that is then fed to the activation function.

## 6 Result

For developing the sky detection model, the model was trained with 10534 images in which ten epochs gave a mean IoU score of 0.7182.

Table 1: Quantitative evaluation of number of images and frequency weighted IoU and mean IoU while training our Sky Detection Model.

From Table 1, the analysis of sample run results is as follows:

1. Run 1 was done with 7898 images in five epochs.
2. To improve the performance after Run #1, we added more images.
3. Run 2 still was not performing well. So now we increased epochs to ten.
4. Run 3 has given better IoU (Figs. 4, 5, 6, and 7).

## 7 Conclusion

We have used the SkyFinder dataset for the sky detection models. In this paper, the PSPNet model for the sky detection model has been implemented. The basic implementation of the web app is presented, along with the model training that implements transfer learning [6].

Fig. 4 Residual block

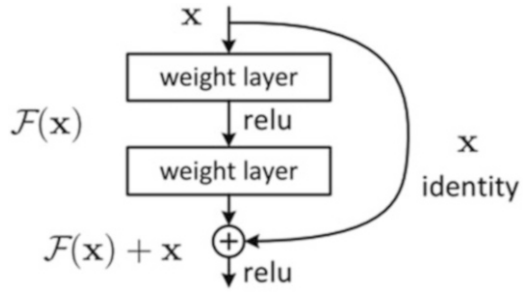


Fig. 5 Sample from Run 1

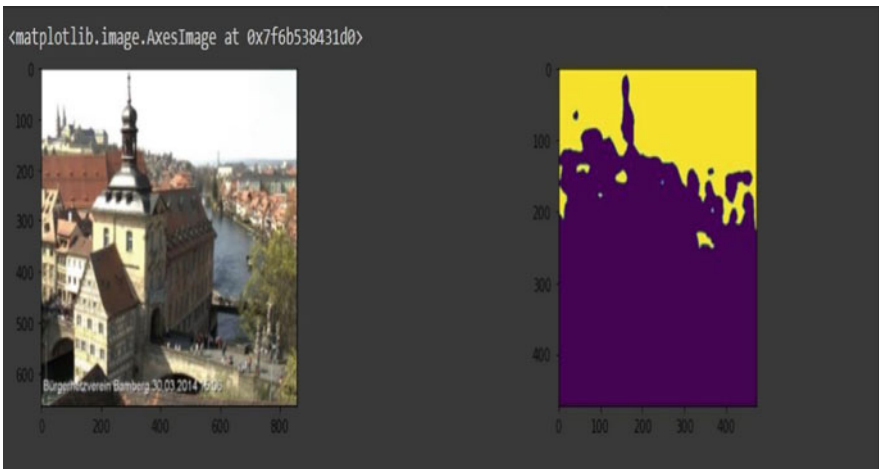


Fig. 6 Sample from Run 2 shows better segmentation around the spire on top left of image



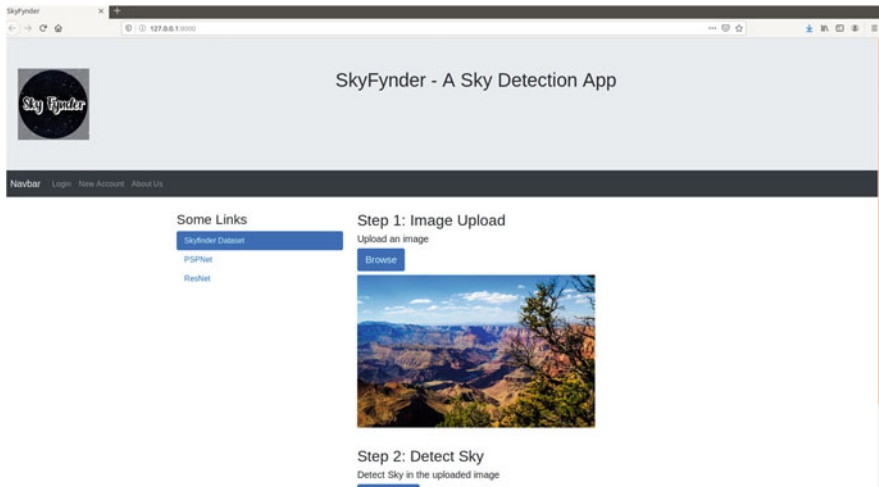


Fig. 7 Image displayed on web app

## 8 Future Scope

The future scope includes building a sky detection model for video footage as well as developing a mobile app for sky detection. The model deployed on a mobile will have to be lightweight and provide fast inference.

## References

1. H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," CoRR, vol. abs/1612.01105, 2016.
2. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," CoRR, vol. abs/1512.03385, 2015.
3. O. Liba, L. Cai, Y.-T. Tsai, E. Eban, Y. Movshovitz-Attias, Y. Pritch, H. Chen, and J. T. Barron, "Sky optimization: Semantically aware image processing of skies in low-light photography," 2020.
4. Y. Song, H. Luo, J. Ma, B. Hui, and Z. Chang, "Sky detection in hazy image," Sensors, vol. 18(4), 2018.
5. Zhao, Zhijie, Qian Wu, Huadong Sun, Xuesong Jin, Qin Tian and Xiaoying Sun. "A Novel Sky Region Detection Algorithm Based On Border Points." International Journal of Signal Processing, Image Processing and Pattern Recognition 8 (2015): 281-290.
6. Zhu, Yida, Haiyong Luo, Qu Wang, Fang Zhao, Bokun Ning, Qixue Ke and Chen Zhang. "A Fast Indoor/Outdoor Transition Detection Algorithm Based on Machine Learning." Sensors (Basel, Switzerland) 19 (2019): n. pag.

# Defining, Measuring, and Utilizing Student's Learning in a Course



Tanmay Garg, Rajat Agarwal, Mukesh Mohania, and Vikram Goyal

## 1 Introduction

With the recent pandemic outbreak, there has been a paradigm shift to online platforms for education. These platforms provide lecture videos, assignments, quizzes, and other relevant features to support quality education. This environment is growing every day with increasing students and resources. The main goal of online learning is to ensure that all students are effectively and efficiently learning the course content despite a lack of personal meetings with the instructor [8].

In the online learning environment where the personal student–instructor interaction is significantly less, the instructor does not have appropriate information about the student and therefore cannot address the individual needs. Instructors primarily assess the student's performance by their marks in the quizzes or exams. With classes moving online, the batch size for classes has increased drastically, making it more challenging for teachers to ensure that each student's performance and effective learning improve. Nowadays, educational institutes and universities have educational databases that are primarily utilized for critical decision-making, such as predicting students' performance and dropout rates [1] and many other tasks.

We coined the term learning velocity (LV) to address the difficulties mentioned earlier in academia. It measures how adequately a student is learning the course content. Students learn better when they perform well in assessments and spend less time on the course contents. After discussion with professional teachers, we

---

The authors "Tanmay Garg and Rajat Agarwal" Indicates equal contribution.

---

T. Garg (✉) · R. Agarwal · M. Mohania · V. Goyal  
Indraprastha Institute of Information Technology, New Delhi, India  
e-mail: [tanmay18368@iiitd.ac.in](mailto:tanmay18368@iiitd.ac.in); [rajata@iiitd.ac.in](mailto:rajata@iiitd.ac.in); [mukesh@iiitd.ac.in](mailto:mukesh@iiitd.ac.in); [vikram@iiitd.ac.in](mailto:vikram@iiitd.ac.in)

realized the need to segregate students based on their learning rate. Different classes are then taught at different speeds. It helps the student feel more comfortable with the lectures' pace and helps them grasp concepts better.

We have categorized students into "Good," "Average," and "Poor" classes based on LV value. Each group represents a particular nature concerning the quiz performance and the pace of the course content. The scope of LV is not confined to segregating students. However, it can also help in several downstream tasks such as predicting the answer guessing, recommendation of contents and learning pathway, revising specific topics, providing insights to instructors for personalized focus to some students for these topics, and many other exciting tasks. In this chapter, we have studied how LV can help predict if a student has guessed to answer a question in the quiz that is discussed later in the paper.

Our contributions can be summarized as:

1. We propose, define, and formulate a learning velocity metric that quantifies the student learning rate that helps instructors in an online setup get insights into student course learning.
2. We utilize LV to predict if a student guesses while answering an MCQ. This shows one of the applications of LV.
3. We perform comprehensive experiments for calculating students LV, categorize them in classes, and predict if a student has guessed a question in MCQ. We research with academicians and discover that the methodology and results are sufficiently realistic.

## 2 Related Work

Zohair et al. [15] describe student performance as achieving the educational goals defined by educational institutes, academicians, and government. Student performance is very subjective, and usually, academicians measure it in terms of final grades such as GPA [5]. However, other researchers have used attributes such as dropout rate, student knowledge, post-course outcomes, etc., to measure student performance [10]. Predicting a student's performance can be a tricky task because of an education program's diverse nature.

Machine learning models prove to be an efficient way of predicting student performance. Gary et al. [14] have predicted student performance in a test in terms of marks so that the faculty could have prior feedback on whether the set test was too difficult or easy for the student. Predicting a student's performance also gives us information about whether the student improved compared to his previous performance. Static machine learning models such as decision tree, support vector machine, K-means, etc., employed by researchers [4, 7, 9, 11, 13]. Several researchers [3, 6, 12] have used time-series data and therefore applied sequential machine models such as BILSTM, attention models, graph neural networks, etc., in their work.

However, these works on predicting student performance do not explicitly focus on capturing student's learning but rather consider an aspect of student learning such as his grades, dropout, etc. Our work focuses on capturing student learning considering educational data and utilizing this information in several downstream tasks such as categorizing students in classes, predicting if a student guessed an MCQ or not, and many other impressive tasks.

### 3 Dataset Description and Preprocessing

The database obtained from the online education company contained different SQL tables for tests taken by students for each chapter and the video content for each sitting. We have extracted relevant information from these SQL tables. We aim to keep only the columns that can act as attributes regarding a student's learning and performance. Each student had a unique student ID that could link the data across different SQL tables. We treated a student's performance in two separate chapters as independent data points.

The data has over a hundred thousand data points for each chapter for the student's content covered. We discarded entries where students had taken less than 0.4 times the total length of the video and entries where the student had taken more than 2.5 times the video length to cover the video content. The assumption was made after careful consultation with the teachers that have curated the content. We also discarded students that had covered less than 10% syllabus from a given chapter. It made a significant fraction of our data unusable.

The dataset contains information about grade eleventh students for the "Physics" subject. The data has records of their quiz attempts and the lectures that they attended. Their performances across three chapters are recorded, i.e., "Laws of Motion," "Work Power Energy," and "Rotation Motion." Each data entry has a student ID, and a content ID, along with the time a student spent on the content and their performance, the number of questions they solved correctly, and the total number of questions (in case the content was a quiz taken by the student). Tables 1 and 2 summarize the attributes along with their description, which are used in this chapter.

Almost 2340 students have attended "Laws of Motion," 1951 students have attended "Work Power Energy," and 1859 students have attended "Rotation Motion" with the online education institute; 3124 unique students have taken any course with the online education institute, and 1177 students have taken all three chapters with the online education institute.

Course videos are significant components of the course material in an online environment, and academicians consider a positive relationship between the number of videos watched and the course completion [2]. Furthermore, assignment and quiz performance positively relates to the course learning. A student's learning from an online educational platform is primarily based on "**Quiz performance**" and "**Video**

**Table 1** Quiz data attributes and description

Attribute	Description
Set ID	Unique quiz ID
User ID	Unique student ID
Set time	Maximum time allotted for quiz
User time taken	Time taken by student to attempt the quiz
Total marks	Maximum marks of quiz
Marks scored	Total marks scored by the student
Percentage marks	Percentage of marks scored by the student

**Table 2** Video data attributes and description

Attribute	Description
Content ID	Unique video ID
User ID	Unique student ID
Video time	Video length in seconds

**content”** covered. In a particular chapter, LV statistically captures these two aspects of course content coverage and test performance.

## 4 Methodology

### 4.1 Calculating Learning Velocity

The motivation behind defining learning velocity is to give the online education institutes a parameter that can directly segregate students into batches to provide different batches and adjust the pace of teaching them accordingly. We define a new value to evaluate a student’s learning, and we call it the learning velocity of the student. The learning velocity helps us gain insights into the learning power or capability of a student. It tells the academicians about how effectively a student can learn something new. For example, someone might be able to grasp concepts with minimal supervision of anyone and perform well. In contrast, another student might require additional management and must be taught 2–3 times before understanding the concepts. The time spent on a particular content helps us determine the speed at which the student learns. Their performance tells us whether what they knew was effectively learned or not.

To calculate the learning velocity of a student across a chapter, we first find his learning across the various topics of that chapter. To find the learning velocity at the topic level, we need to derive information from the quizzes and the lecture videos that the student has covered. After finding the learning velocity at the topic level, we calculate the learning velocity of the student at the chapter level. We now discuss the steps to calculate the LV:

1. **For video content:** We calculate LV for content as the ratio of “Expected time of Video” and “Total time across all sittings.” Its value equal to 1 represents a consistent student who has completed the video content.

There are multiple entries for particular video content in the dataset, representing that a student has taken multiple sittings to complete the video content. We have considered the first entry where the student completed the entire video content in such cases.

Sometimes time spent on video is much greater than the actual video length, and it is not practical to spend much time on video. In these cases, we discard the entry. We decided to discard any entries that had a time more than 2.5 times the length of the video. The expected time to watch a video was equal to the length of the video. The LV of content was then calculated by dividing the expected time by the time taken.

$$\text{Expected time video} = \text{Video Time} \quad (1)$$

$$LV_{\text{content}} = \frac{\text{Expected time video}}{\text{Time taken}}. \quad (2)$$

2. **For Quiz content:** LV for content is defined as the ratio of “Set Time” to “User Time Taken.” We assume that a student who has subject knowledge will need at least “Set Time” to complete the quiz and in a case the student does not know the subject could attempt it within the allotted time. We discard outlier data entries here as well.

$$\text{Expected time to attempt a quiz} = \text{Set Time} \quad (3)$$

$$LV_{\text{content}} = \frac{\text{Expected time to attempt a quiz}}{\text{User Time Taken}}. \quad (4)$$

3. **Performance:**

To measure the student's performance, their percentage marks and not their Quiz Score are considered. We intended to find whether a student has effectively learned a given topic or not.

From the quiz data, we have calculated the “Quiz Score” of a student in a particular quiz by dividing the number of questions that the student solved correctly by the total number of questions. The “Quiz Score” received is a value between 0 and 1 since the total number of questions will always be greater than or equal to the number of correctly solved.

$$\text{Quiz Score} = \frac{\text{Total correct attempted Questions}}{\text{Total questions in Quiz}} \quad (5)$$

Based on our discussions with academicians, we have decided to ignore how correctly one is attempting questions rather than how many can the student solve

correctly, given the student has covered the topic before attempting the questions. A student's performance for a particular chapter is calculated by taking the arithmetic mean of his scores across all the quizzes the student attempted. Since the quizzes are a graded component of the course, no student was allowed to re-attempt a particular quiz.

$$\text{Performance} = \frac{\text{Sum of Quiz Score of all quizzes}}{\text{Total number of quizzes}} \quad (6)$$

#### 4. Learning velocity across a topic:

Final LV across a topic is a summation of all LVs across that topic divided by the number of contents (video and quiz).

$$LV_{\text{topic}} = \frac{\sum LV_{\text{content}}}{\text{number of content}}. \quad (7)$$

5. **Learning velocity across a chapter:** We finally calculate the LV chapter by taking an arithmetic mean of the LV of the topics and multiplying it by the performance of the student.

$$LV_{\text{chapter}} = \text{Performance} \times \left( \frac{\sum LV_{\text{topic}}}{\text{number of content in}} \right). \quad (8)$$

In the initial experiment, we considered one chapter from a course and calculate the LV using the equations discussed above. The numerical value of LV does not present intuitive meaning to the academicians, and in order to handle this situation, three classes based on normalized LV values are created. These classes are named as "Poor," "Average," and "Good" students.

## 4.2 Classification Based on Learning Velocity

Students are different, and it is irrational to assume that if we randomly pick 100 students in a class and place them all together, they will all find the pace of the ongoing lecture to be okay. Hence, there is a need to segregate students based on metrics to be taught at a pace with which they are comfortable. For example, weaker students can attend extra classes focusing on fundamentals and practicing more straightforward problems. In contrast, fast learner students can be encouraged to solve more complex problems and focus less on the fundamental problems.

We now have the LV for a student in a particular chapter. We categorize the students based on their LV individually for each chapter by segregating them based on LV values. We consider that a particular topic can be more complex than the other, so we end up having one large group containing three separate entries for each student. The reason behind not treating each chapter differently is that we

also want to find the topics in which students are performing poorly so that the educational institute can focus more on those topics for the better learning of the students. There is a possibility that a given topic is slightly more complex in general than the other topics, in which case the institute gets to know that they have to prepare more materials for the students in that topic.

The segregation is done based on their relative performance and discussion with academicians. After the discussion, it is decided that the top 25% learning velocity values are treated “Good,” the following 50% learning velocity values represent “Average” category, and the last 25% corresponds to “Poor.” It enabled a student to be in multiple categories at the same time. For example, a student can be “Good” in a chapter such as “Laws of motion” and at the same time can be “Poor” in other chapter such as “Rotation motion.”

### **4.3 Guess Prediction**

One major problem that many education institutes face online is that students make many guesses instead of solving the questions while attempting MCQ questions. While it may allow them to score better sometimes, it hampers their learning and can be detrimental to them in the long run as most topics taught in school build up from previous topics. Hence, a method is needed to be devised to find whether the student had guessed the questions or not.

As mentioned earlier, the data consists of information about questions and corresponding quizzes. We use this information to find out the top five most popular questions. The popularity of a question is determined by the number of times students attempt a question. The reason behind picking the most popular questions is to ensure that the question is attempted by many students and not just a few, which could be misleading. We get the most popular questions for each chapter by performing a “group by count” query on the dataset, followed by a non-increasing sorting cover count value and selecting the first five occurrences of this chapter. Any question left unanswered by any student during a quiz is not counted in the “group by count” query. The reason behind excluding these entries is that we are not sure whether the student has gotten the question right or not. It might also be possible that the student did not get time to see the problem.

Moreover, by the nature of our problem, we want to find out whether the student guessed the problem or not. Hence, including entries where no answer has been given does not contribute to this problem. We want that any selected question has a significant number so that it contains many students from each student category defined by us. Now we used the following algorithm to determine whether a student guessed the answer or attempted it properly.

Calculating whether students guessed MCQ or not:

1. First, we find all the students from the “Good” category correctly attempting a given question. Here, we have disregarded the students who solved the question



incorrectly. The primary intuition for picking the “Good” category students for determining guess is that we assume “Good” category students must have solved the question rather than make a guess.

2. Next, we take the arithmetic mean of the time taken by each of these students to solve the question. We call this value the “Class Time.”
3. Lastly, we examine students who attempted the question (both correctly and incorrectly). We have differentiated between students from various categories. We came up with these thresholds after finding the 1st quartile, 2nd quartile, and the 3rd quartile for the overall distribution of threshold values. After careful consultation with academicians from the online educational institute, this decision was taken. The studies between them also vary from category to category based on the following decision rules:

$$\text{Thresh Value} = \frac{\text{Question solving time by a student}}{\text{Class Time}}. \quad (9)$$

$$\text{Lower Bound}(LB) = \begin{cases} 0.4, & \text{'Good' category} \\ 0.6, & \text{'Average' category} \\ 0.8, & \text{'Poor' category} \end{cases} \quad (10)$$

$$\text{Guess Prediction} = \begin{cases} \text{Guess,} & \text{Thresh Value} \geq \text{LB} \\ \text{Not Guess,} & \text{otherwise.} \end{cases} \quad (11)$$

If the thresh value is above the lower bound, the student did not guess the answer, else it is a guess and did not solve the problem completely.

The reason for having different lower bounds for each category of students is that the learning velocity for each student is different. The higher lower bound implies that a student needs more time to solve the question due to a lack of course coverage or poor past performance in this topic. We do not set an upper bound since each question has an upper limit time available in the database. The upper time allotted for solving the questions is more than the three times the time required to solve the question, provided someone knows the concept needed to solve this question. The upper limit and lower bound values of time are decided based on the consultation with the online education institute teachers.

We aim to help online education institutes find whether the content they provide is being effectively learned by the students and determine whether students are honestly attempting the quiz questions or just making random guesses. The information received from our study helps gain insights into both of these questions.

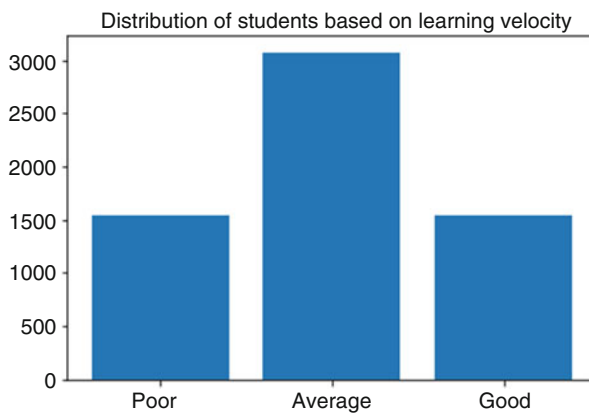
## 5 Results and Analysis

### 5.1 Learning Velocity Distribution

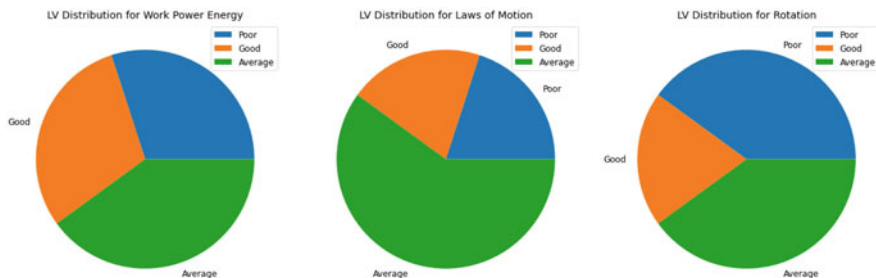
Out of the total 3124 students, we got 6150 entries regarding the learning velocities of students on individual chapters. Out of these, 2340 were in “Laws of Motion,” 1951 were in “Work Power Energy,” and 1859 were in “Rotation Motion.” We have treated each Student ID, Chapter ID pair as a separate data entry. This means that one student can belong to the “Poor” category for a particular topic and be in the “Average” or “Good” category for another topic. We aim to judge a student's performance in a particular chapter. A student might be great at learning the concepts in one chapter but not in another. We are, moreover, clubbing the performance across different chapters together.

Figure 1 summarizes the distribution of students based on LV value into three classes as “Poor,” “Average,” and “Good.” We had 1538 “Poor” category Students, 3075 “Average” category students, and 1537 “Good” category students. A majority of students are average, while a few are good. This scenario is common in most classes where most students have average learning capacity and a few students with poor and good learning capacity. Educational institutes can prepare plans for each class. “Good” students can be encouraged to practice hard and deeper problems to learn better. In contrast, the academically weaker students may be given extra attention in additional classes, low-difficulty-level material, and more examples to prevent their dropout. These steps help to enhance quality education.

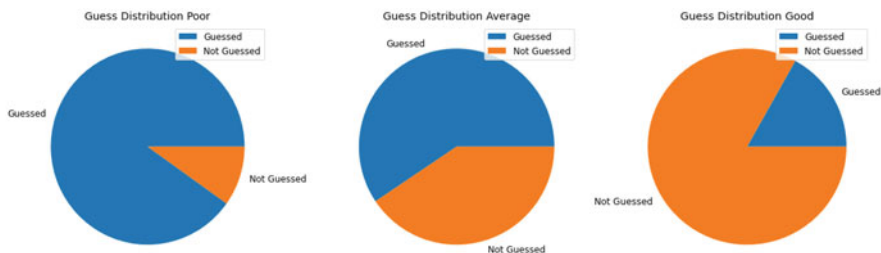
Furthermore, we do not normalize the learning velocity chapter-wise since we want to determine which chapters students struggle in and which chapters they find comparatively more straightforward. These insights can be beneficial for the



**Fig. 1** Categorization of students into “Poor,” “Average,” and “Good” classes based on learning velocity value



**Fig. 2** Learning velocity distribution across three chapters



**Fig. 3** Guess distribution across three categories (namely poor, average, and good) of students

Education Institutes since they know which topics need more focus. Moreover, it helps them find out whether their content regarding that chapter is too tricky. We also observe that out of all the three chapters, “Rotation Motion” has the highest amount of students belonging to the “Poor” category, followed by “Laws of Motion” and “Work Power Energy” as presented in Fig. 2.

### 5.2 Guess Prediction Analysis

We have only taken the top five most popular questions for conducting our analysis. Only the popular questions had sufficient data statistics, considering other samples with fewer data can be misleading. We observe that the maximum students who guessed the questions belonged to the “Poor” category, followed by the “Average” category and the “Good” category as mentioned in Fig. 3. In the “Good” Category, 17.3% of the students have guessed the answer, 40.6% in the “Average” category, and 87.7% in the “Poor” category. A total of 266 good students have guessed a question, 1248 for average students, and 1349 for poor students. Interestingly, even though the “Poor” category has half the number of students as the average category, the number of students who guessed the question is still higher in the poor category. A possible reason could be that these students have not covered the syllabus properly, and therefore, their concepts are weak, and they rely on guessing to score the marks.

## 6 Conclusion

The proposed methodology helps in quantifying student learning. This chapter opens a dimensional space for researchers and academicians to develop a metric that categorizes students based on their performance and the pace of course content covered so that educational institutes can do proper planning to help the students. Learning velocity helps categorize students based on their learning and performance, which can be better for their overall experience. The traditional way of segregating students based on their performance may seem lucrative at first but can be flawed, especially in the online mode. Some students might perform and learn well in class and might have had a bad day. On the other hand, other students might have made a few guesses that would have worked in their favor. Educational companies build software that aims to help academicians, and we find that this research project can create a high impact in academia. Creating software that can predict whether a student has guessed a question or not based on their learning can prove to be helpful in academia. The motivation behind segregated education is that it helps students to be taught at a higher pace and needs a little extra time to grasp the material. A rating instead of a ranking system is a better approach since it tells us where a student stands and whether the student belongs to “Good,” “Average,” or a “Poor” category. Ranking systems can result in students barely above-average getting allotted into a classroom with exceptional students, or vice versa, which can be detrimental for their learning.

The scope of LV is not confined to segregating students. However, it can also help in several downstream tasks such as predicting the answer guessing, recommendation of contents and learning pathway, revising specific topics, providing insights to instructors for personalized focus to some students for these topics, and many other exciting tasks.

In our dataset, we also find that students are not watching course videos, and the institute should develop plans that encourage students to cover the course efficiently. These explanations and recommendations, along with the accurate predictions, are our significant contributions. In the future, we look forward to incorporating personal information such as mother tongue language, family background, and gender while calculating the LV.

## References

1. Ahuja, R.; Jha, A.; Maurya, R.; and Srivastava, R. 2019. Analysis of educational data mining. In *Harmony Search and Nature Inspired Optimization Algorithms*, 897–907. Springer.
2. de Barba, P. G.; Kennedy, G. E.; and Ainley, M. 2016. The role of students' motivation and participation in predicting performance in a MOOC. *Journal of Computer Assisted Learning* 32(3): 218–231.
3. Doan, T.-N.; and Sahebi, S. 2019. Rank-based tensor factorization for student performance prediction. In *12th International Conference on Educational Data Mining (EDM)*.

4. Elbadrawy, A.; Studham, R. S.; and Karypis, G. 2015. Collaborative multi-regression models for predicting students' performance in course activities. In *Proceedings of the Fifth International Conference on Learning Analytics and Knowledge*, 103–107.
5. Hellas, A.; Ihantola, P.; Petersen, A.; Ajanovski, V. V.; Gutica, M.; Hynninen, T.; Knutas, A.; Leinonen, J.; Messom, C.; and Liao, S. N. 2018. Predicting academic performance: a systematic literature review. In *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 175–199.
6. Hu, Q.; and Rangwala, H. 2019. Academic performance estimation with attention-based graph convolutional networks. *arXiv preprint arXiv:2001.00632*.
7. Karimi, H.; Derr, T.; Huang, J.; and Tang, J. 2020. Online academic course performance prediction using relational graph convolutional neural network. In *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, 444–450.
8. Karimi, H.; Huang, J.; and Derr, T. 2020. A Deep Model for Predicting Online Course Performance.
9. Polyzou, A.; and Karypis, G. 2016. Grade prediction with models specific to students and courses. *International Journal of Data Science and Analytics* 2(3): 159–171.
10. Rastrollo-Guerrero, J. L.; Gomez-Pulido, J. A.; and Duran-Dominguez, A. 2020. Analyzing and predicting students' performance by means of machine learning: a review. *Applied Sciences* 10(3): 1042.
11. Umair, S.; and Sharif, M. M. 2018. Predicting students grades using artificial neural networks and support vector machine. In *Encyclopedia of Information Science and Technology, Fourth Edition*, 5169–5182. IGI Global.
12. Unal, D. S. 2019. Modeling Student Performance and Disengagement Using Decomposition of Response Time Data. In *EDM*.
13. Wang, R.; Harari, G.; Hao, P.; Zhou, X.; and Campbell, A. T. 2015. SmartGPA: how smartphones can assess and predict academic performance of college students. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 295–306.
14. Zhao, Y.; Xu, Q.; Chen, M.; and Weiss, G. M. 2020. Predicting Student Performance in a Master of Data Science Program Using Admissions Data. *International Educational Data Mining Society*.
15. Zohair, L. M. A. 2019. Prediction of Student's performance by modelling small dataset size. *International Journal of Educational Technology in Higher Education* 16(1): 1–18.

# Holistic Features and Deep-Guided Depth-Induced Mutual-Attention-Based Complex Salient Object Detection



Surya Kant Singh and Rajeev Srivastava

## 1 Introduction

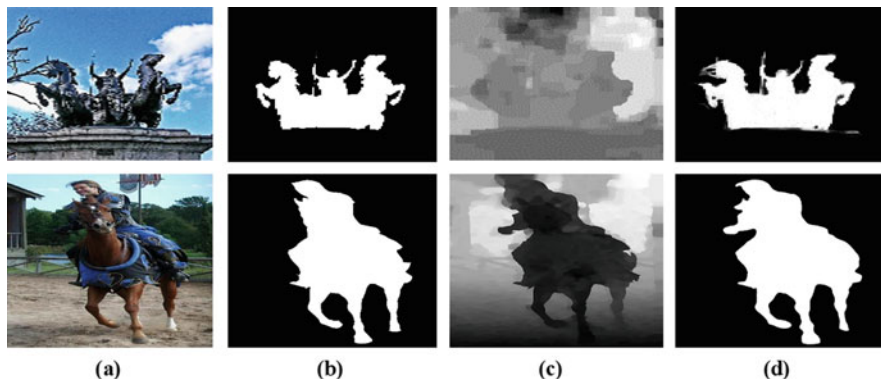
Visual salient object detection generates conspicuous and prominent objects in the complex and clutter background image, identified by the human visual system. The saliency computations are commonly used as an integral part in many vision-related applications, such as semantic segmentation [1], object classification [2], and content-based image editing [3].

The main motivation of the three-stream network is to explore all possible features to predict the salient objects correctly. Two networks ( $stream_1$ ) and ( $stream_3$ ) are based on color and depth modalities. These networks target non-complementary features such as purely color contrast-based features, depth contrast-based features, and regional color features.

The learning and discriminating features in CNN are essential for designing multi-stream models. Various two-stream network-based improved models [4–6] have been proposed and improved the performance. Similarly, the most recent, multi-stream network [7] has been utilized in complex scenarios to achieve the next level benchmark. These contemporary architectures improved the performance. However, these recent models discriminate of complementary and non-complementary features in successive integration in middle-level strategy. The proposed model uses a middle-level fusion strategy and mutual attention to address the aforementioned limitations. This model predicts saliency maps similar to human perception, illustrated in Fig. 1:

---

S. K. Singh (✉) · R. Srivastava  
Computer Science and Engineering, Indian Institute of Technology (Banaras Hindu University),  
Varanasi, India  
e-mail: [rajeev.cse@iitbhu.ac.in](mailto:rajeev.cse@iitbhu.ac.in)



**Fig. 1** Holistic feature space and proposed depth-induced mutual attention produce deep localized features for detecting salient objects in complex and clutter backgrounds. (a) Input Image. (b) Ground Truth. (c) Depth Map. (d) Our

- In this chapter, a three-stream network has been proposed. Two independent streams are dedicated to color and depth modalities. The third stream uses the features of the above two streams to produce all possible essential features, defined as *holistic features space*. It includes a non-complementary, cross-, and intra-complementary features. The non-complementary feature-based fusion preserves the modality-specific saliency.
- We designed a middle fusion strategy, *cross-complementary fusion (CF)*, and a deeply guided attention map, *depth-induced mutual attention map*. The fusion strategy fuses the cross- and intra-complementary features, driven by a deep localized enhanced feature.
- The comprehensive experiments on four complex datasets demonstrate remarkable improvement with the state-of-the-art methods.

The rest of the chapter is organized in the following sections. The comprehensive review of closely related deep-learning-based RGBD models is described in Sect. 2. Section 3 describes the proposed method in detail. Section 4 discusses the experiment setup and demonstrates the performance with other state-of-the-art saliency detection methods. Section 5 describes the conclusion and the future scope of improvements in this model.

## 2 Related Works

Ample saliency object detection (*SODs*) models have been developed over the last two decades. Early (*SODs*) models [8–10], based on low-level, handcrafted features without learning and testing. At the same time, the recent deep-learning-

based methods are based on high-level, contextual, and semantic features. The deep-learning-based model employed CNNs to improve performance and achieved encouraging benchmark results. Finding the salient object in a low-depth image is nearly impossible. Therefore, depth features were utilized first time in saliency computation by Niu et al. [11] to address the low-depth issue. Then, a large-scale RGBD dataset NLPR is constructed by Peng et al. [12] and proposed multi-stage fusion of RGB and depth feature for saliency computation.

The early fusion model utilized the low-level handcrafted features as inputs in deep CNN. In this DF [13] model, CNN integrated different low-level features with depth saliency to produce the salient object. But in these models, most saliency is lost in the handcrafted feature, which is irrecoverable in CNN. Therefore, CNN is not fully utilized in the early fusion model. An adaptive fusion *AF-Net* [5] is based on the late fusion strategy. But it failed in the complex and cluttered background because it does not address the complementary features in low-depth and complex images. Similarly, Han et al. [4] proposed a two-stream CNN to fuse RGB-D deep features. Another, most recent model using this strategy, D3NET [7] is proposed. It has a three-stream network such as RgbNet, RgbDNet, and DepthNet. This method has not explored the intra-complementary features between deep- and shallow-level saliency computation.

The middle-level fusion strategy is utilized to efficiently explore the cross- and intra-complementary features between color and depth images. Initially, these models [14, 15] were put forward by proposing a complementary-aware RGB-D saliency detection model. Another similar end-to-end RGBD framework CTMF [4] based on the generative adversarial network is proposed. Next, JL-DCF [15] model has been proposed. It uses a shared Siamese backbone network based on joint learning (JL) and densely cooperative fusion (DCF). Recent models introduced attention mechanisms to overcome the regional disparity to compute saliency. This mechanism assigned a different weight to provide a different importance to different regions. A classic model of attention mechanism S2MA [6] used CNN in a two-stream RGBD network. The improvements are achieved by using a selective attention mechanism for filtering out low saliency values. This model is our motivation to utilize attention mechanisms to improve the performance.

### 3 The Proposed Method

Three stream networks, based on RGB (*Stream<sub>1</sub>*), depth (*Stream<sub>3</sub>*), and cross-modality (*Stream<sub>2</sub>*), produce holistic features space. The most preferred backbone network, VGG-16 [16], is used to produce the holistic feature space. The three-stream architecture of the proposed method and learning module is visualized in Fig. 2.



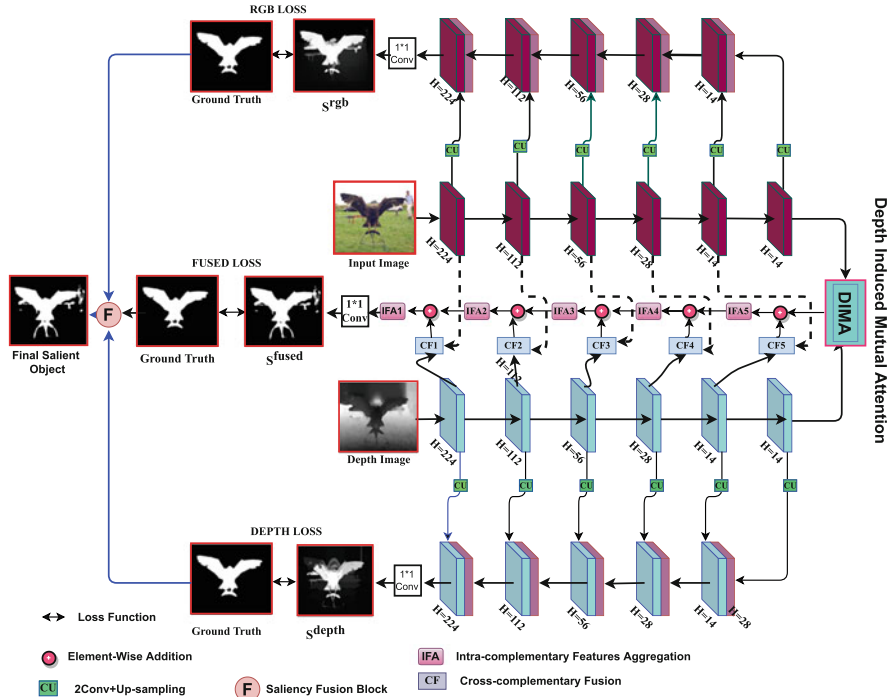


Fig. 2 The proposed model has a three-stream network. Two streams,  $Stream_1$  and  $Stream_3$ , have color and depth input image of  $224 \times 224 \times 3$

### 3.1 Non-Complementary Feature Aggregation

The non-complementary features generated by the  $VGG-16$  in the color and depth streams are produced separately and simultaneously, demonstrated in  $stream_1$  and  $stream_3$ , respectively, in Fig. 2. Let us use five convolution blocks  $Conv1_2$ ,  $Conv2_2$ ,  $Conv3_3$ ,  $Conv4_3$ , and  $Conv5_3$ , in backbone feature generation. The outputs produced by these blocks are denoted as  $C_i$ . Their corresponding saliency features are  $S_i$ . We add a convolution block and upsampling layer with resolution  $224 \times 224$  at the end of the RGB and depth stream. The backward aggregation of these features fused the feature  $S_i$  with the deep-fused feature  $S_{i+1}$  at  $i$  and  $i + 1$  scales, respectively. Finally, these two streams produce their saliency maps  $S^{rgb}$  and  $S^{depth}$  separately and simultaneously. Let us define modalities  $m = (RGB, Depth)$  in following Eq. (1). The formulation of features generation and saliency prediction is defined in Eq. (1) as follows:

$$S_i = \psi(\varphi(\varphi(C_i))) \quad 1 \leq i \leq 5$$

$$\tilde{S}_i = \begin{cases} \varphi(i + 1, \tilde{S}_i) & 1 \leq i < 5 \\ S_i & i = 5 \end{cases} \quad (1)$$

$$S^m = \text{Sig}(k_s * \tilde{S}_i + \text{bias}).$$

$\psi(\dots)$  denotes the upsampling function to make raw saliency with the same resolution that uses bilinear interpolation.  $\varphi(\dots)$  describes convolution operation with 64 channels. It is followed by a non-linear activation function. In this convolution operation, the kernel size is  $3 \times 3$  and the stride size is 1.  $k_s$  is  $1 \times 1$  kernel and  $\text{bias}$  is the bias parameter.  $\text{Sig}(\dots)$  is the Sigmoid function, while  $*$  represents the convolution operation.  $|\dots|$  represents a channel-wise concatenation. These features preserve the modality-specific saliency and produce cross- and intra-complementary fusion, which is guided by the proposed attention map and defined in the following sections.

### 3.2 Depth-Induced Mutual Attention—DIMA

The RGB and depth maps have cross-complementary features, which are essential to detect a complete salient object. This objective has been achieved by a deeply guided depth-induced mutual attention map. The depth- and color-based deep features produce the enhanced localized feature using spatial-, mutual-, and feature-level attention mechanisms. The spatial mask ( $7 \times 7$ ) is followed by another spatial window ( $7 \times 7$ ). It is suggested by Xu et al. [17] where two ( $7 \times 7$ ) spatial masks are used to enhance the geometrical features.

The fused modality-based mutual attention map is proposed to improve the deepest features and enhance deep localized features to start the fusion process.

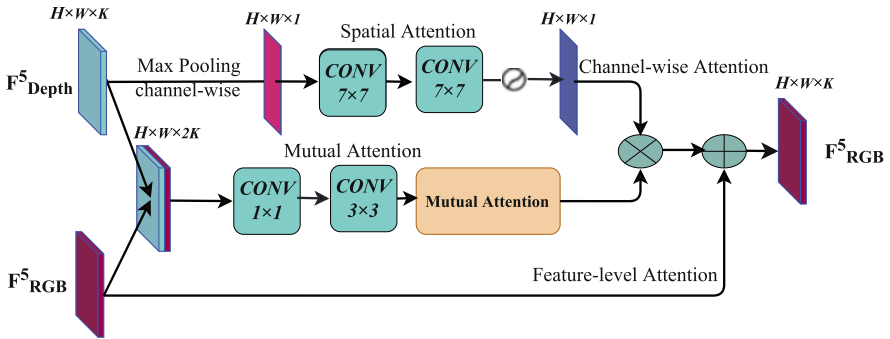


Fig. 3 The proposed depth-induced mutual attention map DIMA to enhance deep localized features and start the fusion process

First, reduce the channels in concatenated features from both modalities by applying  $1 \times 1$  convolution mask. Then after  $3 \times 3$  convolution mask, enhance the more details in combined features. The mutual attention feature is achieved by using both deep features, which is shown in Fig. 3.

### 3.3 Complementary Features Fusion Model

The process of  $stream_2$  is described in the following two stages.

#### Cross-Complementary Fusion (CF)

The CNN backbone network in color  $stream_1$  and depth  $stream_3$  produces saliency features as in [18] (side output). The saliency features from depth and color stream from each stage (total six stages from  $CF1$  to  $CF6$ ) are fused into the CF model. In this model, the varied resolution features are compressed into smaller (fixed size equal to  $k$ ) and exact sizes. This compressed feature contains both depth and RGB information separately. The processed saliency features in  $RGB$  and  $depth$  modality are defined as follows:  $Sf_{rgb}$ ,  $Sf_{depth}$ , each with equal  $k$  channels. The output of the  $CF$  module is defined in Eq. (2) as

$$C^k(Sf_{rgb}, Sf_{depth}) = (Sf_{rgb}^k \otimes Sf_{depth}^k) \oplus (Sf_{rgb}^k \oplus Sf_{depth}^k). \quad (2)$$

In this cross-view fusion, “ $\otimes$ ” and “ $\oplus$ ” are defined as element-wise multiplication and addition, respectively. The output of the  $CF$  model from each stage is successively fused from deep localized by proposed  $DIMA$ . The fused features  $CF6$   $CF1$  are fed into a dense decoder [19] that has a dense connection to purify the saliency. These dense connections are used to unify the multiscale features at multi-stages.

#### Intra-Complementary Features Aggregation (IFA)

In this fusion model, the feature maps of all preceding layers are used as inputs for each layer to find the intra-complimentary features. The fusion process is supervised by a deeply guided depth-induced mutual attention map.

This model is similar to the original inception module [16], with one difference. In this method, the same channel number  $k$  is maintained in input and output. Finally, all three saliencies are fused with simple element-wise addition and normalized to generate the final salient object.

### 3.4 Loss Function

The total loss function comprises stream-wise saliency loss defined in Eq. (3). The loss function is the standard cross-entropy loss, which is defined in [5]. The color, depth, and fused saliency loss functions are computed with their respective saliency maps  $S^{rgb}$ ,  $S^{depth}$ ,  $S^{fused}$  and ground truth map  $Gt$ . The total loss function is defined as

$$\xi_{total}(S, Gt) = \sum_{i \in (rgb, depth, fused)} \xi(S^i, Gt). \quad (3)$$

## 4 Experiment and Result Analysis

The related parameters, experimental setup, dataset, implementations detail, and network architecture are described in detail in the following sections.

### 4.1 Dataset and Evaluation Metrics

We have conducted extensive experiments on publicly available RGB-D benchmarking datasets: STEREO-1000, NJUD-2000, and RGBD-135 [7]. Most of the contemporary models have used the data pattern for training and testing, which is used in [5]. We use recent evaluation metrics widely used in recent comparisons. These metrics are (1) S-measure, (2) F-measure, (3) mean absolute error (MAE), and (4) E-measure ( $E_{\psi}$ ). All these parameters are recent and adequately defined in [15].

### 4.2 Implementation Details and Training Details

The backbone network of this model is the VGG-16 model [16], JL-DCF [15]. The existing pre-trained parameters of mostly preferred method DSS [18] are used to initialize the backbone network. The size of convolution layers in all CFF modules is  $(3 \times 3)$  and filter size is  $k = 64$ . We have used the stride of 1 in *Max - Pools* to enhance the resolution of the coarsest feature maps. Upsampling operation by multiple factors is performed in each stage-wise saliency feature to maintain the same resolution. The Caffe has been used to implement the proposed model. This optimization has batch size 8 and a learning rate of 0.0001. The approximate training time is around 18 h/16 h, which contains 40 epochs in the VGG-16 configuration.

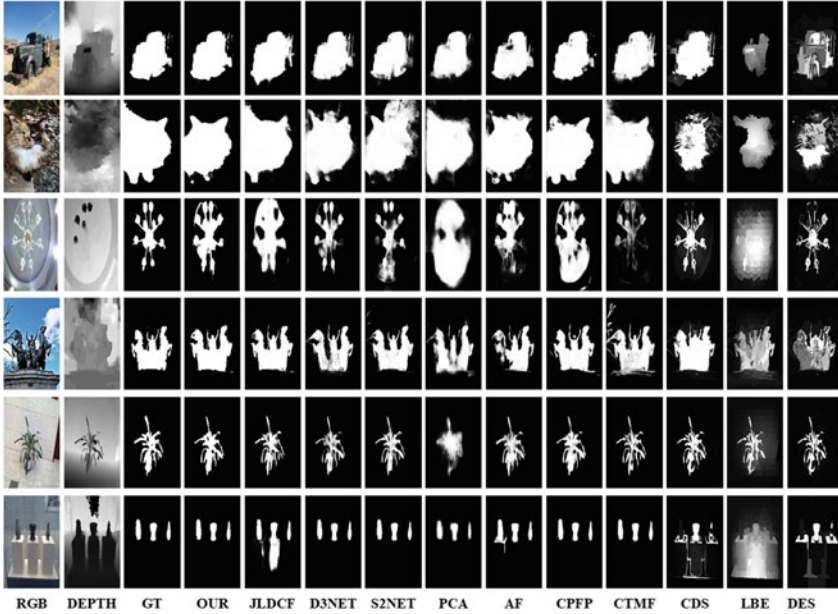


Fig. 4 The visual demonstration of the proposed model with other state-of-the-art methods

### 4.3 Comparison and Result Analysis

We compare the proposed model results with the following twelve state-of-the-art methods: JL-DCF [15], S2NET [6], D3NET [7], CPFP [20], AFNet [5], CTMF [4], PCANet [14], DF [13] are closely related deep-learning-based RGBD methods, while CDS [9], MDSF [10], DES [8], and LBE [21] are the traditional approaches. Note that the above-used saliency maps are either produced by running source codes or pre-computed and publicly posted by corresponding authors. The result analysis is demonstrated through visual comparison and quantitative comparison. Table 2 illustrates that the proposed model achieves a significant improvement. The visual demonstration is shown in Fig. 4 to show the visual superiority of the proposed model.

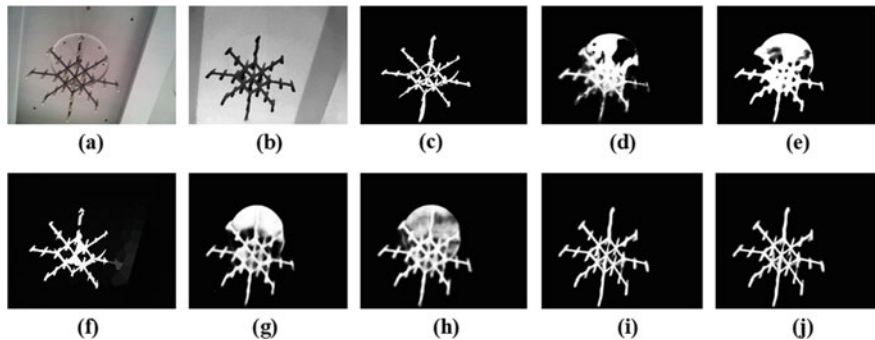
### 4.4 Ablation Analysis

#### Validation of Three-Stream Network

The validation is essential to demonstrate the contribution of each stream in the final saliency contributions. Any three streams cannot distinguish the salient regions individually in the complex background. The gradual improvements in three-stream saliencies are visible in Table 1 and Fig. 5.

**Table 1** The quantitative comparison of the proposed framework on four benchmark RGBD datasets is shown here

*	Metric	NLPR					NJU2K					RGBD-135					STERE				
		$F_{\beta}^m \uparrow$	$S_{\alpha} \uparrow$	MAE $\downarrow$	$E_{\psi}^m \uparrow$	$F_{\beta}^m \uparrow$	$S_{\alpha} \uparrow$	MAE $\downarrow$	$E_{\psi}^m \uparrow$	$F_{\beta}^m \uparrow$	$S_{\alpha} \uparrow$	MAE $\downarrow$	$E_{\psi}^m \uparrow$	$F_{\beta}^m \uparrow$	$S_{\alpha} \uparrow$	MAE $\downarrow$	$E_{\psi}^m \uparrow$	$F_{\beta}^m \uparrow$	$S_{\alpha} \uparrow$	MAE $\downarrow$	$E_{\psi}^m \uparrow$
	<b>OUR</b>	0.917	0.925	0.021	0.965	0.910	0.908	0.041	0.955	0.924	0.924	0.021	0.970	0.906	0.907	0.041	0.952	0.906	0.907	0.041	0.952
	JL-DCF [15]	0.916	0.925	0.022	0.962	0.903	0.903	0.043	0.944	0.919	0.929	0.022	0.968	0.901	0.905	0.042	0.946	0.901	0.905	0.042	0.946
	S2NET [6]	0.902	0.915	0.030	0.953	0.849	0.899	0.053	0.941	0.935	0.973	0.021	0.961	0.882	0.890	0.051	0.932	0.882	0.890	0.051	0.932
	D3NET [7]	0.897	0.912	0.030	0.953	0.900	0.900	0.041	0.950	0.885	0.898	0.031	0.946	0.891	0.899	0.046	0.938	0.891	0.899	0.046	0.938
	CPEP [20]	0.867	0.888	0.036	0.932	0.877	0.879	0.053	0.926	0.846	0.872	0.038	0.923	0.874	0.879	0.051	0.925	0.874	0.879	0.051	0.925
	PCFNet [14]	0.841	0.874	0.044	0.925	0.872	0.877	0.059	0.924	0.804	0.842	0.049	0.893	0.860	0.875	0.064	0.925	0.860	0.875	0.064	0.925
	CTMF [4]	0.825	0.860	0.056	0.929	0.845	0.849	0.085	0.913	0.844	0.863	0.055	0.932	0.831	0.848	0.086	0.912	0.831	0.848	0.086	0.912
	AFNet [5]	0.771	0.799	0.058	0.879	0.775	0.772	0.100	0.853	0.728	0.770	0.068	0.881	0.823	0.825	0.075	0.887	0.823	0.825	0.075	0.887
	DF [13]	0.778	0.802	0.085	0.880	0.804	0.763	0.141	0.864	0.766	0.752	0.093	0.870	0.757	0.757	0.141	0.847	0.757	0.757	0.141	0.847
	MDSF [10]	0.793	0.805	0.095	0.885	0.775	0.748	0.157	0.838	0.746	0.741	0.122	0.851	0.728	0.719	0.176	0.809	0.728	0.719	0.176	0.809
	CDS [9]	0.768	0.782	0.098	0.824	0.779	0.744	0.160	0.803	0.786	0.791	0.129	0.832	0.746	0.741	0.122	0.851	0.746	0.741	0.122	0.851
	LBE [21]	0.745	0.762	0.081	0.855	0.748	0.695	0.153	0.803	0.788	0.703	0.208	0.890	0.633	0.660	0.250	0.787	0.633	0.660	0.250	0.787
	DES [8]	0.681	0.702	0.125	0.700	0.704	0.713	0.189	0.754	0.666	0.682	0.143	0.770	0.566	0.582	0.193	0.670	0.566	0.582	0.193	0.670



**Fig. 5** Visual demonstration of the contribution of three-stream network in complex image having inferior and low-depth image. We define  $s^{DC} = (\omega_1 \otimes (s^{Depth} \oplus s^{s^{RGB}}))$ ,  $s^{DF} = (\omega_1 \otimes (s^{fused} \oplus s^{depth}))$ , and  $s^{CF} = (\omega_2 \otimes (s^{rgb} \oplus s^{fused}))$ . (a) Input Image. (b) Depth Map. (c) Ground Truth. (d)  $S^{Depth}$ . (e)  $S^{rgb}$ . (f)  $S^{Fused}$ . (g)  $S^{DC}$ . (h)  $S^{DF}$ . (i)  $S^{CF}$ . (j)  $S$

**Table 2** The validation of effectiveness of three-stream network is shown here using mean absolute error (MAE) in the proposed model

<i>Dataset</i>	$s^{Depth}$	$s^{rgb}$	$s^{Fused}$	$s^{DC}$	$s^{DF}$	$s^{CF}$	$S$
NLPR	0.0389	0.0349	0.0248	0.0355	0.0238	0.0226	0.0216
NJUD2K	0.0558	0.0495	0.0402	0.0500	0.0407	0.0389	0.0411
RGBD-135	0.0400	0.0373	0.0205	0.0366	0.0235	0.0236	0.0217
STERE	0.0588	0.0548	0.0450	0.0570	0.0433	0.0418	0.0418

## Effectiveness of Holistic Feature Space

The validation of this ample feature space is necessary, which is strongly supported through the results of Table 2. The significant improvements in the results shown in Table 2 are observed in all datasets with all parameters validating the importance of the proposed model. The successive contributions in saliency computations are shown in Table 1 and visually shown in Fig. 4, which validate the effectiveness of each stream of the proposed model on complex RGBD datasets.

## 5 Conclusion

The proposed model defined a holistic feature space. This feature space includes all essential features produced by the three-stream network guided by DIMA. Further, these features have been used by naval proposed fusion strategy, using middle-level fusion strategies to explore relevant features from holistic space. An innovative, deeply guided depth-induced mutual attention map has efficiently located the salient object with the exact object border and removed the background. The creative, progressive learning is designed to generate these features into three-stream

networks, and the joint fusion strategy remarkably improves the performance. Our model has generalization capability, which can further incorporate some other model to enhance saliency computation.

## References

1. Jerripothula, K. R., Cai, J. & Yuan, J. Image co-segmentation via saliency co-fusion. *IEEE Transactions on Multimedia* **18** (9), 1896–1909 (2016).
2. Durand, T., Mordan, T., Thome, N. & Cord, M. *WILDCAT: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation* (2017).
3. Wang, W. & Shen, J. *Deep cropping via attention box prediction and aesthetics assessment*, 2186–2194 (2017).
4. Han, J., Chen, H., Liu, N., Yan, C. & Li, X. CNNs-based RGB-D saliency detection via cross-view transfer and multiview fusion. *IEEE Transactions on Cybernetics* **48** (11), 3171–3183 (2017).
5. Wang, N. & Gong, X. Adaptive fusion for RGB-D salient object detection. *IEEE Access* **7**, 55277–55284 (2019).
6. Liu, N., Zhang, N. & Han, J. *Learning selective self-mutual attention for RGB-D saliency detection*, 13756–13765 (2020).
7. Fan, D.-P., Lin, Z., Zhang, Z., Zhu, M. & Cheng, M.-M. Rethinking RGB-D salient object detection: Models, data sets, and large-scale benchmarks. *IEEE Transactions on Neural Networks and Learning Systems* (2020).
8. Cheng, Y., Fu, H., Wei, X., Xiao, J. & Cao, X. *Depth enhanced saliency detection method*, 23 (ACM, 2014).
9. Zhu, C., Li, G., Wang, W. & Wang, R. *An innovative salient object detection using center-dark channel prior [c]* (2017).
10. Song, H. et al. Depth-aware salient object detection and segmentation via multiscale discriminative saliency fusion and bootstrap learning. *IEEE Transactions on Image Processing* **26** (9), 4204–4216 (2017).
11. Niu, Y., Geng, Y., Li, X. & Liu, F. *Leveraging stereopsis for saliency analysis*, 454–461 (IEEE, 2012).
12. Peng, H., Li, B., Xiong, W., Hu, W. & Ji, R. *RGB-D salient object detection: a benchmark and algorithms*, 92–109 (Springer, 2014).
13. Qu, L. et al. RGB-D salient object detection via deep fusion. *IEEE Transactions on Image Processing* **26** (5), 2274–2285 (2017).
14. Chen, H. & Li, Y. *Progressively complementarity-aware fusion network for RGB-D salient object detection*, 3051–3060 (2018).
15. Fu, K., Fan, D.-P., Ji, G.-P. & Zhao, Q. *JL-DCF: Joint learning and densely cooperative fusion framework for RGB-D salient object detection*, 3052–3062 (2020).
16. Szegedy, C. et al. *Going deeper with convolutions*, 1–9 (2015).
17. Xu, K. et al. *Show, attend and tell: Neural image caption generation with visual attention*, 2048–2057 (PMLR, 2015).
18. Hou, Q. et al. *Deeply supervised salient object detection with short connections*, 3203–3212 (2017).
19. Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. *Densely connected convolutional networks*, 4700–4708 (2017).
20. Zhao, J.-X. et al. *Contrast prior and fluid pyramid integration for RGB-D salient object detection*, 3927–3936 (2019).
21. Feng, D., Barnes, N., You, S. & McCarthy, C. *Local background enclosure for RGB-D salient object detection*, 2343–2350 (2016).



# Machine Learning Based Decision Support System for Resilient Supplier Selection



Saurav Kumar, Anoop Kumar Dixit, and Milind Akarte 

## 1 Introduction

The supply chain has always been affected by unpredictable and unforeseeable events, and it hampers both productivity and profitability. There can be many reasons responsible for uncertainties, including natural disasters and manmade actions leading to accidents and strikes [1]. Therefore, supply chain risk management is one of the key areas of research [2]. Researchers proposed strategies to assess, mitigate and monitor these risks [3]. In the past, numerous mathematical programming methods have been employed to tackle uncertainties like stochastic programming, MCDM methods, fuzzy methods, optimizations, and simulation techniques. However, with the advent of big data in recent times, many approaches are being developed under the umbrella of data analytics, like data mining, process mining, and artificial intelligence. This has become possible because of the development of data collection and storage capabilities. Machine learning techniques allow a computer program to learn from input data, which must be labelled in case of supervised learning and unlabeled in case of unsupervised learning [3].

Machine learning models influence supply chain risk decision making, and these risks can be broadly divided into four categories: (i) *Supplier risk*: selection of supplier based on past transactional data (e.g., on-time delivery rate of the supplier), (ii) *Demand risk*: estimating the uncertainties in consumer demand in terms of order quantities or dates and assisting in the peak-and-trough analysis. (iii) *Capacity risk*: addressing fluctuating needs because of seasonal patterns or overlapping customer order books addressing, and (iv) *Process/product risk*: determining product com-

---

S. Kumar · A. K. Dixit (✉) · M. Akarte  
National Institute of Industrial Engineering (NITIE), Mumbai, India  
e-mail: [anoop.2004005@nitie.ac.in](mailto:anoop.2004005@nitie.ac.in); [milind@nitie.ac.in](mailto:milind@nitie.ac.in)

plexity (e.g., chances of getting it right the first time) and developing product profiles (e.g., runners, repeaters or strangers) [4].

Different machine learning techniques are employed by the researchers for managing the supply risk, but the most used are classification algorithms like Support Vector Machine, Logistic Regression, Artificial Neural Network. But apart from these, there are few other techniques also whose usefulness in the supply risk domain is being explored, like Natural language Processing, Bayesian Networks, and Reinforced Learning. Natural Language Processing based models are mostly used to extract supply chain maps from textual data, thus increasing the supply chain visibility [5]. Reinforced learning models consist of intelligent agents who get penalized for wrong actions & rewards for the correct ones, and thus model learns how to interact with the environment; these models are used to select appropriate demand forecasting techniques [6].

## 2 Literature Review

There is voluminous work carried out to reduce the supply risk and improve delivery reliability and ascertain the resilient supplier to make the supply chain robust. The literature review is briefly summarized below in three sections: supplier selection, resilient supply chain network, and supply chain mapping.

Tayaran H. et al. has put forward a hybrid technique combining simulation & machine learning for resilient supplier selection in digital manufacturing using KNN, Logistic Regression [7]. Hosseini S. et al. analyzed main contributors to the resilience of supplier selection & suggested a hybrid ensemble and AHP approach for resilient supplier selection using Logistic Regression, CART, ANN [8]. El-Hiri M. brings forward a framework using ANN for supplier selection & monitoring their performance [9]. Hamdi F. et al. reviewed the published literature regarding supplier selection under supply chain risk management [10]. Abdollahnejadbarough H et al. developed an analytics toolset to perform supplier rationalization for Verizon strategic sourcing teams & supplier negotiations using RNN and NLP [11].

Wang W et al. reviewed the literature on resilient supply chain performance & proposed a fuzzy gain-loss computational approach to evaluate resilient suppliers [12]. Tordecilla, R.D. et al. optimized methods for designing resilient supply chain network & also identified research opportunities for hybrid approaches for uncertainty & dynamic conditions modeling [13]. Nezamoddini N. et al. presented a framework for optimizing supply chain network design & planning considering the risk perspective of decision-makers using ANN and GA ML models [14].

Wichmann P. et al. proposed the ANN, NLP based model to automatically extract the supply chain maps using news articles [5]. Handfield R. et al. brought an approach that uses newsfeed data to assess regional supply base risk for the apparel sector [15]. Brintrup A. et al. presented a case study harnessing historical data to predict supply chain disruption using different machine learning techniques like Random Forest, SVM, KNN, Logistic Regression, Linear Regression [16].

It is evident from the literature that researchers have used Machine Learning (ML) models for supply risk management. The most used algorithms include Support Vector Machine (SVM), Artificial Neural Networks (ANN), Logistic Regression & CART. Apart from these, other machine learning techniques which are being explored are Natural Language Processing and Reinforcement Learning. It is well known that ML model require large dataset for training and prediction. Most ML models proposed in the literature, especially deep learning, are demonstrated when a large dataset is available. However, there is a paucity of research, especially in the supply chain domain, where the availability of a smaller dataset can also be leveraged to apply the ML model by synthetically generating the data. The research proposes a generic framework for resilient supplier selection through the CTGAN model to harness the limitation of adequate data availability by using ML.

### 3 Methodology

Figure 1 shows the overall research methodology. Initially, “Anylogic” based simulation model was developed to generate a dataset that includes a transactional dataset for order placing and product delivery between two immediate tiers of the supply chain. In the next step, the CTGAN model has been developed to generate a synthetic dataset. CTGAN allows the effective generation of a realistic dataset consisting of multi-modal continuous columns and discrete columns. In the next stage, ML steps are followed that include data preprocessing, feature engineering, evaluation metrics selection, and hyperparameter tuning. In the last phase, the expected date of delivery for every order and on-time delivery probability is determined through ANN. Finally, the model performance is evaluated using various metrics.

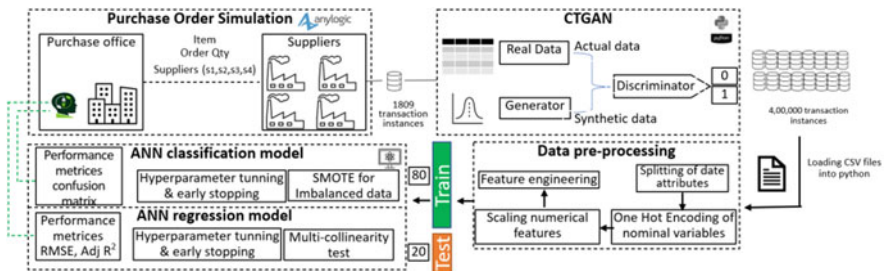


Fig. 1 Overall research methodology

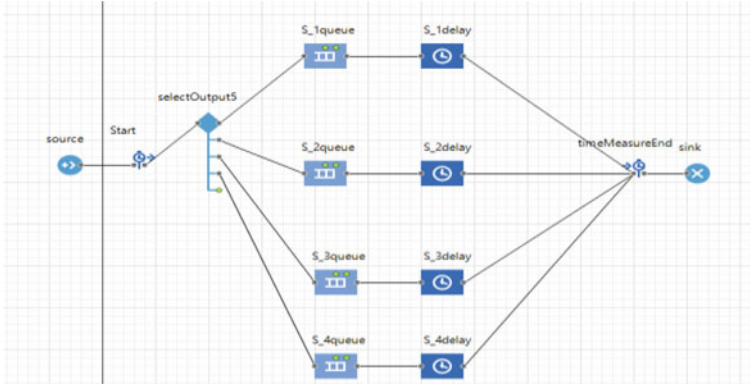


Fig. 2 Simulation model for purchase order

### 3.1 Modeling Purchase Orders Using Simulation

Simulation modeling is the process of creating a digital prototype of a real-world scenario. It can be used to approximate the behaviour in the real system. Hence, it can be used for testing scenarios and constructing the model, which can be helpful in achieving a greater understanding of the system. Thus, a simulation model has been used to generate the dataset with desired attributes. A simulation model of a single product and four possible suppliers is developed using Anylogic software. It is a multimethod simulation modeling tool that supports agent-based, discrete event, and system dynamics simulation methodologies.

Figure 2 depicts the flow diagram of agents in the simulation model. The model was used to mimic a real-world purchasing scenario and generate procurement-related attributes like date of purchase order (PO) release, the date on which the vendor started processing the order, and date of arrival at the buyer end. Few assumptions have been considered while developing the simulation model for purchase order data (i) Purchase order release and processing time for each supplier are normally distributed (ii) Allocation of PO is based on a pre-defined percentage (iii) FIFO is followed for processing the order at supplier end (iv) Single order can be processed at a time.

The model is used to simulate purchase order-related transactions over 20 years; then also it could generate only 1809 instances. In many practical scenarios, the number of total purchase transactions would be limited. But building a Deep Learning model requires a huge amount of data. To solve this problem of scarcity of data points, the research employed Generative Adversarial Networks (GAN).

**Table 1** CTGAN data

Order ID	Supplier ID	PO release	Processing start	Arrival	Quantity rejected (%)
1	B	18-09-2004	20-09-2004	28-09-2004	3.900
2	D	10-09-2019	10-09-2019	18-09-2019	4.725
3	C	19-11-2011	04-12-2011	15-12-2011	3.019

### 3.2 Synthetic Data Generation Using CTGAN

Generative Adversarial Networks (GAN) consists of two models which are trained simultaneously: generative model and discriminative model. The generative model captures the data distribution, whereas the discriminative model tries to estimate the probability that a sample came from training data rather than the generative model. The goal of the generative model is to maximize the probability of the discriminative model making mistakes [17]. GANs performs very well for image data, but the same is not true while dealing with tabular data due to some of the challenges associated with the tabular dataset, such as: Real-world tabular data consists of various data types (integer, decimals, categories, time, text), Continuous variables have different shapes of distribution (multi-modal, long tail, Non-Gaussian), Sparse one-hot-encoded vectors and highly imbalanced categorical columns.

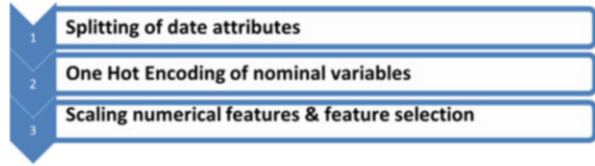
For tabular data, the most used variants of GANs are TGAN and CTGAN. CTGAN outperforms the TGAN models as it overcomes the challenges associated with tabular data by employing the below-mentioned techniques [18]: For continuous variables, a variational Gaussian mixture model (VGM) is used. It first estimates the number of modes and then fits a Gaussian mixture. After it normalizes the initial vector within each mode, the mode is represented as a one-hot vector. Conditional vectors are used to represent the concatenated one-hot vectors of all discrete columns but with the specification of only one category, which was selected. During training by sampling, the goal is to resample efficiently in a way that all the categories from discrete attributes are sampled evenly during the training process; as a result, to get real data distribution during the test.

So, for this study, CTGAN was employed. CTGAN was trained using 1809 instances generated by the simulation model, and 4,00,000 new data instances (Table 1) were created. The quality of the synthetic data was judged using statistical methods: chi-square test and Kolmogorov-Smirnov test.

### 3.3 Data Preprocessing

Figure 3 depicts the steps of preprocessing and feature engineering. Since the data was generated using the simulation model and CTGAN, thus there were no missing values or outliers in the data. Machine learning models cannot be trained directly over the date variables, so purchase order release date, order processing start date,

**Fig. 3** Data preprocessing & feature engineering steps



and order arrival date were split into three components: day of the month, month of the year, and year.

Since all the data attributes were needed to be in numerical form, we have converted the Supplier ID to equivalent numerical form. But Supplied ID is a nominal variable, so label encoding would have induced ranking in the data, which could have altered the performance of the regression problem. Onehot encoding from Sklearn library was used to transform the Supplier ID in 4xn matrix form to avoid this situation.

Regression models cannot have a target variable in the date format; thus instead of arrival date, the duration between the purchase order release and order arrival was used as the target variable. For a classification problem, deliveries were divided into two categories, on-time delivery, and late delivery, depending on whether the order was delivered within the requested delivery date or not. Since the different independent variables were of highly varying magnitude, the scaling of the attributes is needed. Minmax scaler from the Sklearn library was used to scale all the feature values between 0 and 1.

### 3.4 Feature Engineering

To add more features in the data, new attributes were introduced, like for each purchase order release date and order processing start date day of the week, week of the year, and day of the year. Apart from these attributes, the time duration between these dates was also introduced as a feature. The next step was to identify the useful features and discard the redundant features before training regression and classification models. The feature selection library of Sklearn was used to find the relevant features based on the statistical test, including chi-square, correlation coefficient, mutual information gain, and tree classifier. Further, it is important to address the challenge of an imbalanced dataset.

Dataset is termed to be imbalanced when the number of data instances of any category is much higher than the other categories. In such cases, the accuracy score shows higher values even for a zero-rule classifier. Most of the practical case which involves risk belongs to the minority class. There are three methods to overcome the challenge of an imbalanced dataset. These are resampling dataset, weighted loss function, and Synthetic Minority Oversampling Technique (SMOTE).

In the dataset, late deliveries consisted of around 27% of the data instances; thus, it was a case of an imbalanced dataset. Therefore, SMOTE was employed on the

training dataset to overcome the challenges imposed by the imbalanced dataset. SMOTE generates synthetic data using the k-nearest neighbour technique. SMOTE begins by selecting random data from the minority class, after which the data's k-nearest neighbours are determined. The random data and the randomly chosen k-nearest neighbour would then be combined to create synthetic data [19].

### 3.5 ANN Model Training & Hyperparameter Tuning

**ANN Model for Predicting Probability of On-time Delivery** After data preprocessing, CSV files containing independent variables and output class were imported to the google colab for further model building. For the classification problem, the time duration between purchase order release and order arrival date was converted into two classes, on time or late delivery. The data set was divided into two groups. One for training the model (80%) and another for testing the model (20%) using a test train split of the sklearn library.

Artificial Neural Network (ANN) consists of input, output, and intermediate hidden layers. These layers consist of nodes that are connected with the nodes of preceding and succeeding layers; this interconnection has weights, biases, and nonlinear activation functions. The neural networks learn the values of weights by a series of sequential steps: forward propagation, cost calculation, differentiation of loss function, backward pass, and weight update.

The nonlinear activation functions are also of various types. The most used functions include sigmoid and hyperbolic tangent functions. However, these functions suffer from gradient vanishing problems, as for most of the domain, their derivative value is nearly zero, making it difficult for the model to learn weights. To overcome this problem Rectified Linear Unit (ReLU) activation function is used, it does not suffer from gradient vanishing problems, but its derivative is not defined for the negative domain. The solution to this problem is to use the Leaky ReLU activation function. It overcomes the zero-gradient issue from ReLU by assigning a small value for the negative domain. For this study, the Leaky ReLU activation function was used for input and hidden layers. For the output layer softmax activation function was used to predict the probability of output class.

**Hyperparameter Tuning** Neural Network learns the model parameters while training, but hyperparameters need to be tuned for the better performance of the model. For hyperparameter tuning, we used the Keras tuner library to determine the optimum number of layers, the number of nodes in each layer, and the learning rate for the optimizer. Keras tuner determined the hyperparameters for optimum model performance by employing random search in the given space of hyperparameter values. Then ANN model was built as per the output of Keras tuner, Adam was used as an optimizer, and the loss function was sparse categorical cross-entropy. From the training dataset, 20% of the data instances were used for model validation purpose.

Overfitting is one of the common challenges faced during the training of neural networks. To overcome this problem, we employed a callback function from the Keras library. The callback function monitors the accuracy score on the training dataset as well as the validation dataset, and it stops further model training when it discovers a sudden decrease in the accuracy score of the validation dataset.

**ANN Model for Predicting Actual Date of Delivery** For predicting the actual date of delivery, a regression model was built, using the duration between purchase order release and order arrival date as the target variable. After importing CSV files of independent and dependent variables, we checked for multicollinearity. Multicollinearity occurs when there are two or more independent variables in a regression model, which have a high correlation among themselves. Multicollinearity leads to instability/high variance/overfitting of coefficients of the model. Even a small change in the training data would lead to a dramatic change in the coefficients/predictions.

Commonly used techniques to detect multicollinearity is to inspect correlation coefficient or Variance Inflation Factor (VIF). However, not all collinearity issues can be identified by looking at the correlation matrix. This is due to the fact that a correlation matrix only reveals a correlation between two variables. Multicollinearity occurs when there is collinearity between three or more variables, which can be found by looking at the VIF. A  $VIF > 10$  implies a multicollinearity problem in general. From our model, we individually removed all the variables having a VIF value greater than 10. This resulted in 11 independent variables for model training. The dataset for our study consisted of 400,000 data instances, among which 20% of the data was separated for testing purpose, and the remaining 80% was used for training purpose. It was done using a test train split of the sklearn library.

The rest of the ANN model building process was similar to the classification model for predicting probability of on-time delivery. For hyperparameter tuning, we used the Keras tuner library to determine the optimum number of layers, the number of nodes in each layer, and the learning rate for the optimizer. Keras tuner determined the hyperparameters for optimum model performance by employing random search in the given space of hyperparameter values. Unlike the classification model, we kept only one node in the last layer and used linear function as activation function. The loss function was the mean square loss, and the early stopping from callback function was used to check the overfitting of the model.

## 4 Results and Discussion

We apply the proposed framework for selection of resilient suppliers on historical transaction data between two immediate tiers of supply chain, using simulation modeling. It created 1809 instances of attributes like purchase order release date,



**Fig. 4** Hyperparameters values for classification model

```
Objective(name='val_accuracy', direction='max')
Trial summary
Hyperparameters:
num_layers: 5
units_0: 8
learning_rate: 0.01
units_1: 8
units_2: 12
units_3: 2
units_4: 2
Score: 0.993989497423172
```

order processing start date, and date of order arrival. But ANN models require large volume of training data and in real scenarios also it can be a challenge to have a large number of transactional instances. To overcome this problem of the inadequate dataset, further 400,000 new transaction instances were created using CTGAN to train ANN model and new attributes were added in dataset using feature engineering. We divide this to train ANN models to predict probability of on time delivery and actual date of delivery.

#### ***4.1 Model for Predicting Probability of On-time Delivery***

For the classification model, SMOTE was employed for producing a balanced dataset. To determine the hyperparameters such as the number of hidden layers, node in each layer, learning rate for the optimizer, we implemented Keras tuner. The following specifications for the model are depicted in Fig. 4. The classification model was built as per the specifications with a softmax function in the output layer, which predicted the probability of on-time delivery. The model performance metrics calculated include Accuracy Score (0.9769), Precision (0.9886), Recall (0.9796), and F1 Score (0.9844).

#### ***4.2 Model for Predicting Actual Date of Delivery***

Steps similar to the classification model were used to develop the model, and a Keras tuner was employed for hyperparameter tuning. The output is shown in Fig. 5. A regression model was built as per the above specifications with linear function in the output layer, which predicted the actual date of delivery. The model performance metrics include Mean Square Error (5.3577, R2 Square (0.9915), and Adjusted R2 Square (0.9915).

**Fig. 5** Hyperparameters values for regression model

```

Trial summary
Hyperparameters:
num_layers: 6
units_0: 14
learning_rate: 0.001
units_1: 2
units_2: 2
units_3: 2
units_4: 2
units_5: 2
Score: 5.2377331256866455

```

### 4.3 Discussion

The above result shows that the score achieved by evaluation matrices are reasonably high. The reason behind this may be the use of exact distribution for different attributes in the simulation model to generate the input dataset. In real-life scenarios, these attributes may not follow exact distributions; Rather, we try to fit distribution over the dataset and learn its distribution parameters. Similar models can be used in real-life scenarios to predict the probability of on-time delivery and the actual delivery date for available suppliers and thus reduce the supply risk.

The proposed framework can predict probability of on-time delivery & date of delivery for each order given input attributes i.e., supplier ID, PO date, quantity, quality criteria etc. This information can help in assessment of risks associated regarding delivery reliability with individual suppliers and thus supplement decision making process in selection of suppliers with reduced supply risk.

## 5 Conclusion and Future Work

The research proposed a generic framework for resilient supplier selection using ML. It also portrayed how CTGAN models can be harnessed in a case where there is a limited number of data instances. With increasing risks and uncertainty in the supply chain, these models are very much useful in predicting the risk. Although the study used a synthetic dataset generated through a simulation model, the framework can be used to apply the ML models in real-life industry problems with a limited dataset.

In the future, (i) model performance can be studied for real-life datasets, (ii) similar model for more than one product and many suppliers, (iii) performance of the neural network-based models and models based on bagging & boosting can be compared as these ensembled models are also highly capable at the same time they can be computationally cheaper.

## References

1. Snyder, L.: OR/MS Models for Supply Chain Disruptions: A Review. *IIE Transactions* (Institute of Industrial Engineers) 48 (2): 89–109(2016).
2. Behzadi, G.: Agribusiness Supply Chain Risk Management: A Review of Quantitative Decision Models. *Omega* 79: 21–42(2018).
3. Baryannis, G.: Predicting supply chain risks using machine learning: The trade-off between performance and interpretability. *Future Generation Computer Systems*, 101, 993–1004(2019).
4. Baryannis, G.: Supply chain risk management and artificial intelligence: State of the art and future research directions. *International Journal of Production Research*, 57(7), 2179–2202(2019).
5. Wichmann, P.: Extracting supply chain maps from news articles using deep neural networks. *International Journal of Production Research*, 58(17), 5320–5336(2020).
6. Chien, C. -: Deep reinforcement learning for selecting demand forecast models to empower industry 3.5 and an empirical study for a semiconductor component distributor. *International Journal of Production Research*, 58(9), 2784–2804(2020).
7. Tayaran, H.: A framework for online reverse auction based on market maker learning with a risk-averse buyer. *Mathematical Problems in Engineering*, (2020).
8. Hosseini, S.: A hybrid ensemble and AHP approach for resilient supplier selection. *Journal of Intelligent Manufacturing*, 30(1), 207–228(2019).
9. El-Hiri, M.: Suppliers selection in consideration of risks by a neural network. *International Journal of Engineering, Transactions A: Basics*, 32(10), 1454–1463(2019).
10. Hamdi, F.: Optimization of a supply portfolio in the context of supply chain risk management: Literature review. *Journal of Intelligent Manufacturing*, 29(4), 763–788 (2018).
11. Abdollahnejadbarough, H.: Verizon uses advanced analytics to rationalize its tail spend suppliers. *Interfaces*, 50(3), 197–211(2020).
12. Wang, W.: Decision support system toward evaluation of resilient supplier: A novel fuzzy gain-loss computational approach. *Kybernetes*, 49(6), 1741–1765(2019).
13. Tordecilla, R. D.: Simulation-optimization methods for designing and assessing resilient supply chain networks under uncertainty scenarios: A review. *Simulation Modelling Practice and Theory* (2021).
14. Nezamoddini, N.: A risk-based optimization framework for integrated supply chains using genetic algorithm and artificial neural networks. *International Journal of Production Economics*, 225(2020).
15. Handfield, R.: Assessing supply chain risk for apparel production in low cost countries using newsfeed analysis. *Supply Chain Management*, 25(6), 803–821(2020).
16. Brintrup, A.: Supply chain data analytics for predicting supplier disruptions: A case study in complex asset manufacturing. *International Journal of Production Research*, 58(11), 3330–3341(2020).
17. Goodfellow, I.: Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).
18. Xu, L.: Modeling tabular data using conditional GAN. *Advances in Neural Information Processing Systems* 32 (2019).
19. Chawla, V.: SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16: 321–357(2002).

# An Adaptive Task Offloading Framework for Mobile Edge Computing Environment: Towards Achieving Seamless Energy-Efficient Processing



Mohammad Ashique E. Rasool, Anoop Kumar Bhola, Asharul Islam, and Khalid Mohiuddin

## 1 Introduction

Nowadays mobile phones are used in every walk of our life, communication, entertainment, education, healthcare business, etc. With the unprecedented growth in the variety of mobile applications and multimedia information sharing, the processing requirement and power consumption became a point of concern. These mobile devices have limited processing capability and battery life due to their limited size. The process-intensive tasks cannot be completed on time or can even be blocked if completely processed locally [1]. So, processing resource-intensive tasks on resource-limited devices has become a problem. This problem could be solved through Mobile Cloud Computing (MCC), where the process-intensive task can be offloaded [2] to the cloud server but the latency due to the remote location of the cloud server creates the bottleneck. Mobile Edge Computing (MEC) has emerged as a responsive potential solution to this bottleneck.

In the MEC environment, the miniature resourceful cloud servers, also referred to as Edge Servers or Edge Nodes, are placed at the edge of the network nearer to the mobile and IoT devices. In the new edge computing architecture, computation offloading task placement is possible on any edge node along with the option of local processing or remote processing at the cloud server [3]. In the edge computing

---

M. A. E. Rasool (✉) · A. K. Bhola

Department of Computer Science, Banasthali Vidyapith, Radha Kishnpura, Rajasthan, India

A. Islam

Department of Information System, King Khalid University, Abha, Saudi Arabia

K. Mohiuddin

Department of Management Information System, King Khalid University, Abha, Saudi Arabia

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023

R. Misra et al. (eds.), *Advances in Data Science and Artificial Intelligence*,

Springer Proceedings in Mathematics & Statistics 403,

[https://doi.org/10.1007/978-3-031-16178-0\\_5](https://doi.org/10.1007/978-3-031-16178-0_5)

environment, the devices can determine which data should be stored and processed locally and offloaded to the edge node or the cloud server for further processing [4].

Offloading is one of the promising solutions to overcome the limitations of resource-limited mobile devices, especially battery life [5]. The advanced features of mobile devices like 4G and 5G network access help quickly connect edge node's resources. In the mobile cloud computing environment, smartphones can access resources  $24 \times 7$ . In the last few years, Mobile Cloud Computing (MCC) has been highly demanding. It helped reduce the processing load on mobile devices by offloading process-intensive and specific tasks to a remote cloud server [6]. But offloading the process-intensive applications to distant cloud servers adds latency and security issues. This led the researcher to consider other possible options such as MEC [7] and cooperate computing [8]. One of the landmarks in this direction is to shift some of the applications from the mobile device and execute them on a resource-rich node/server at the network edge near the mobile device with distant cloud-server-like processing capabilities and service offerings [7]. This technique is termed Mobile Edge Computing (MEC). MEC brings various advantages over MCC, such as lower latency, better privacy and security of mobile devices, saving energy/battery of the mobile devices and supporting context-aware computing [9]. Hence, MEC is a promising technology for expanding mobile devices' resources and capability.

The offloading mechanism can be understood with flowchart in Fig. 1. It gives an abstract view of offloading decision based on different criteria, for example, application and data size (computation requirement), nature of application (real time or non-real time) and could be partitioned or not.

## 2 Motivation

Limited size, battery and processing capacity are central issues with mobile and IoT devices. This research identified the following questions: Can mobile devices efficiently perform process-intensive computation seamlessly with negligible latency without draining too much battery? Or can we minimize power consumption and maximize throughput while executing computer-intensive applications on mobile devices?

1. **Limited Computing Capacity of the Small Device:** Computing capacity is the most important factor for efficiency. So many innovative applications are being developed and incorporated promptly on smartphones, but we cannot deny that these devices are still considered resource-limited devices. On the other hand, the users and developers want to execute more computation-intensive applications on their smartphones.
2. **Limited Battery Life of Mobile Devices due to Its Smaller Size:** Executing process-intensive smartphone applications degrades performance and reduces battery life span. Therefore, this is a crucial issue to provide the required

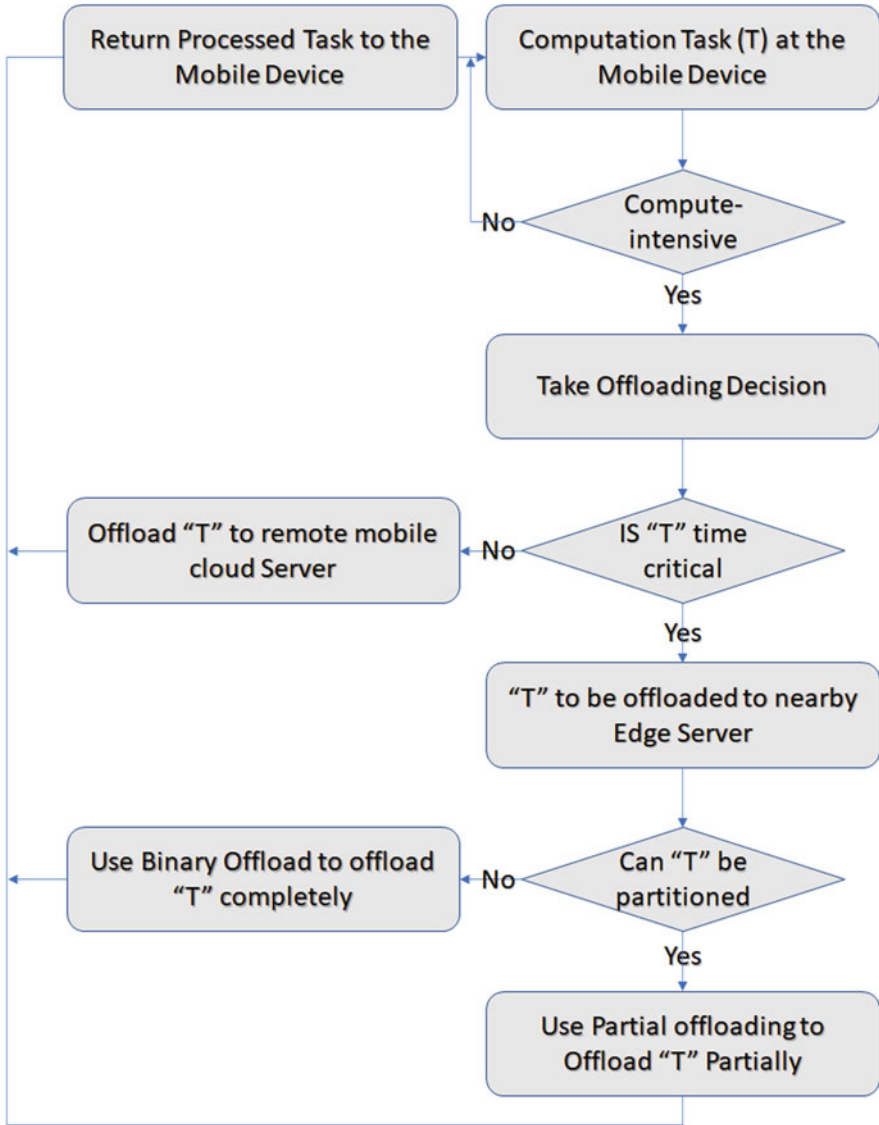


Fig. 1 An abstract view of task offloading mechanism

performance capability and enhanced battery life for these devices. This issue has become serious nowadays to unexpected usages and trends of using smartphones in many real-time applications. In addition, the usage of these devices for accessing the Internet has imposed massive traffic on the network, the web and the application servers, increasing power consumption. Thus, there is a keen need for all possible improvements in resource allocation to enhance efficiency and improve energy consumption trends.

3. **Offloading Decision and Access Latency:** To improve performance and reduce power consumption, there is a need for an optimal offloading decision, minimizing latency and maximizing throughput. There is still a huge scope for optimizing the offloading decision. When the device is mobile, there is a fair chance of encountering a barrier. At this point, a revised offloading decision or resumption should significantly improve performance and reduce power consumption at the local device.

Mobile Edge Computing has evolved as one of the most promising solutions. The process-intensive tasks are offloaded to the nearby edge nodes to reduce the latency in these real-time applications. However, Mobile Edge Computing infrastructure supports processing vast amounts of information through offloading and on-demand access.

### 3 Contributions

In this research study, our main objective is to minimize battery consumption and latency while processing real-time applications on mobile devices by employing an optimal offloading decision and revised offloading decision while encountering a barrier when the device is on the go.

The proposed contributions are listed below:

1. To analyse currently available offloading frameworks and parameters that affect performance, battery life and latency in the MEC by an intensive study of the related work.
2. To propose a service-oriented framework for an optimal adaptive offloading decision that delivers a strategy for saving energy and enhancing the performance of the mobile devices in the MEC environment.

### 4 Related Work

Our focus is to optimize offloading decisions to achieve maximum energy efficiency with minimum latency. The offloading problem is also described as cyber foraging or remote execution on a third-party device. In this section discussed the related scholarly works.

In [10], Y. Mao et al. have shown the offloading decision optimization, job scheduling and power provisioning for single-user MEC architecture with multiple autonomous jobs to minimize the execution latency and power consumption.

In [11], Liu et al. considered the social relations of the trusted mobile devices having energy harvesting abilities to model a dynamic task offloading mechanism based on game theory for the fog computing environment to minimize overall

trusted social group computation cost. This study assumes homogeneous fog computing nodes for demonstrating the offloading problem.

In [12], Zhang et al. proposed an iterative search algorithm for jointly optimizing the local computation resources, transmission power and channel allocation for achieving an optimum trade-off between the energy consumption and the execution latency.

In [13], A. Yousefpour et al. discussed the task offloading mechanism where the fog computing servers share the required task completion load in a distributed fashion to minimize service latency. This architecture does not contemplate any centralized unit for disseminating the execution load among the fog computing nodes. Though, it utilizes the information of the neighbouring node for allocating the tasks among the fog network nodes.

In [14], Guo et al. proposed a task offloading mechanism for ultra-dense IoT networks. The mobile devices can offload their service needs to the mobile edge nodes installed at the network provider's base stations. The objective of this research is to minimize the computation throughput. This research further proposes a two-tier game-theoretic greedy task offloading scheme for dynamically varying availability of computing resources at the MEC servers.

In [15], F. Shan et al. proposed a two-step approach to minimize the power requirements of tiny IoT devices built on transparent computing by offloading the delay-sensitive tasks to the Transparent Computing Server. The first step analyses the code block, including the data associated with the task, to determine if offloading is required. The second step schedules the offloading tasks to minimize the IoT device's energy consumption further.

In [16], G. Lee et al. proposed a dynamic approach to choose a group of neighbouring fog nodes to create a network under ambiguity on the arrival of the fog computing nodes. This research work aims to minimize the computing latency by optimizing the task allocation in the fog-cloud network.

In [17], T. Yang et al. proposed an offloading mechanism for the maritime mobile cloud system for ships and container terminals. Further, in this work, the researchers proposed an enhanced Hungarian algorithm for selecting optimal task execution nodes to reduce the energy consumption in the container terminals and the computing task's execution latency.

In [18], G. Zhang et al. considered a heterogeneous fog computing network. They proposed a fair and energy-minimized task offloading (FEMTO) algorithm to effectively offload the computation tasks from IoT devices to the fog nodes.

In [19], Zhang et al. considered a block chain-enabled MEC environment to work on the joint computation offloading model and coin loaning model problem, similar to the cost of execution of tasks on the remote cloud servers. This problem was devised as a non-cooperative game for minimizing the actual cost of the mobile device.

In [20], Farhan et al. took advantage of the Jackson network to model a load-sharing distributed Mobile Edge Computing (MEC) network by developing a complex multi-objective optimization problem to optimize execution delay, energy consumption and actual financial cost of the smart mobile devices.



In [21], Zhaolong Ning et al. designed a MEC-enabled 5G health monitoring system for IoMT. This research aims to minimize the patient data processing cost, which depends significantly on the energy consumption for processing the health monitoring data. They considered intra-WBANs, gateways to regulate the data transmission rates of the body sensors by appropriate bandwidth allocation to minimize the data processing cost. A cooperative game theory was proposed to achieve optimal resource allocation. The patients can analyse the medical data by local devices or edge servers.

Table 1 compares the recent state-of-the-art research on offloading models for the MEC environment. The comparison table clearly shows that most of the distributed task offloading models do not provide load-sharing mechanism and ignore the support from the resource-rich cloud infrastructure. Furthermore, these research works only consider energy consumption minimization or execution latency but ignore the network barriers and signal strength, which may play a significant role in offloading decisions and overall throughput. The limitations of this research work motivated us to study the computation further offloading in a multi-user and distributed load-sharing MEC environment in association with the resource-rich cloud, improving the mobile device's performance and enhancing the performance battery life.

## 5 Proposed Framework

Figure 2 shows the proposed framework for efficient offloading. This section explains the proposed service-oriented framework for optimal offloading decisions. This framework will estimate the mobile device's execution time and power consumption. It will enhance the use of mobile device resources by offloading the applications. This framework will also give a revised offloading decision if a communication barrier is encountered. Most of the previous work ignored the situation when a barrier is encountered. The barriers could be physical or logical, like a high-rise building, basement, tunnel, etc., or the hollow area between the two base stations.

The proposed framework has a layered architecture with virtualization layer, service layer, decision layer and communication layer. In the decision layer, we proposed two novel components, a signal strength monitor and a barrier analyser. These two components together will make a decision when a barrier is encountered. The framework will be implemented at the edge server or the base stations from where the decision will be made.

**Table 1** Compression of the state-of-the-art research on offloading models

Research work	Offloading models		Computing platform				Optimization parameters			
	Framework	MCC	MCC	MEC	Execution latency	Power consumption	Network barriers			
[10] Mao et al. (2017)	Centralized	X	✓	✓	✓	✓	X			
[11] Liu et al. (2018)	Centralized	✓	✓	✓	✓	✓	X			
[12] Zhang et al. (2018)	Centralized	X	✓	✓	✓	✓	X			
[13] Yousefpour et al. (2018)	Distributed	✓	✓	✓	✓	X	X			
[14] Guo et al. (2018)	Distributed	X	✓	✓	✓	✓	X			
[15] Shan et al. (2019)	Centralized	X	✓	✓	✓	✓	X			
[16] Lee et al. (2019)	Distributed	✓	✓	✓	✓	X	X			
[17] Yang et al. (2019)	Distributed	✓	✓	✓	✓	✓	X			
[18] Zhang et al. (2019)	Distributed	✓	✓	✓	X	✓	X			
[19] Zhang et al. (2020)	Distributed	✓	✓	✓	X	X	X			
[20] Farhan et al. (2020)	Distributed	✓	✓	✓	✓	✓	X			
[21] Zhaolong et al. (2021)	Distributed	✓	✓	✓	✓	✓	X			
Our study	Distributed	✓	✓	✓	✓	✓	✓			

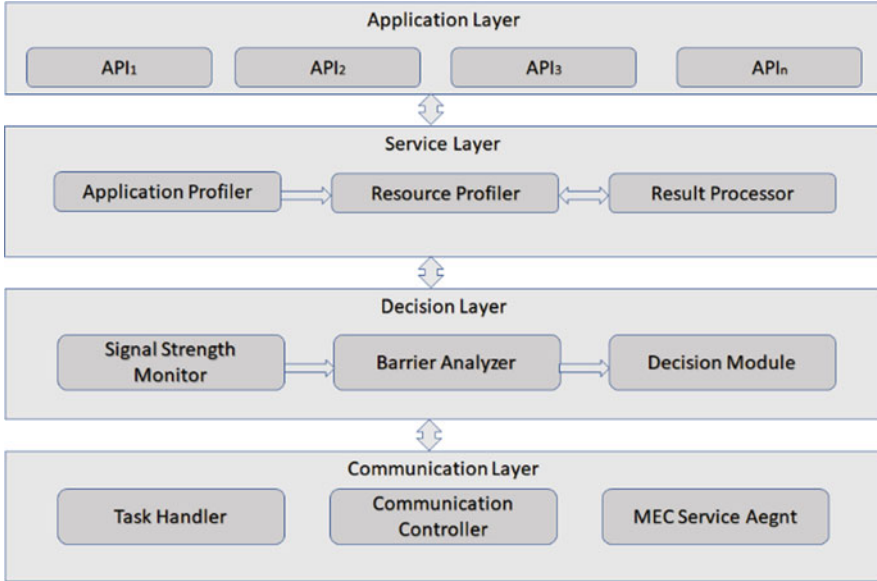


Fig. 2 An abstract view of task offloading mechanism

## 5.1 Communication Layer

The communication controller is a core component in the proposed framework. It handles connection requests for communicating its offset component (of remote cloud) across the architecture. It is a stimulant to the architecture efficiency and performance based on effective coordination between communication controller and MEC collaborator. This layer dynamically receives the offloading decision from the decision layer and offloads the task based on the decision.

The MEC service agent analyses architecture's resources such as edge nodes, intermediate fog nodes or the remote cloud for resource deployment. Further, this agent updates the available resource information and stores the updated resource information in the resource profiler. The current resource status information, such as computing capacity, IP address, connection congestion level, the barrier and signal strength, are critical for the framework service efficiency. This agent also coordinates with a signal strength monitor to minimize resource hunts such as edge nodes in the framework.

When the decision layer decides to offload a task, the task handler wraps the required information (code, input data, needed libraries) to make an offloading package. When the offloading package gets ready, it is dispatched to the specified edge or cloud node for processing. The node address (edge or cloud server) is determined by communication with the task handler. The task handler at the edge or cloud node unpacks the received package and synchronizes the libraries that need

to be executed on the edge server. Once the execution is completed, the result will be sent back to the client's mobile device. Effective and efficient communications between the handlers on both sides will contribute to performance efficiency.

## ***5.2 Decision Layer***

This layer will decide on whether to offload a task or not. The decision module plays a significant role in performance efficiency. As much as the decision will be efficient, the device's overall performance will be efficient. The running information received from the service layer, the signal strength monitor and the barrier analyser offloading decision will be taken based on MCDM method. The signal strength monitor continuously monitors the signal strength. If the signal strength is below the given limit, then offloading may take longer and have high latency. So, to take offloading decisions, the data received from signal strength monitor plays a significant role. Mobile signal frequency is identified by the signal strength monitor, and the barrier analyser decides whether a barrier is encountered or not. It assumes a barrier of signal degradation and a huge fluctuation in the signal strength. In case of a barrier, the offloaded processing task will continue over the edge server until the device gets out of the barrier. Once the barrier ends and the previous edge server is not in the range decision module will send its location information to the nearest edge server. The offloaded task will be shifted through an edge-to-edge collaborative network to this edge server.

## ***5.3 Service Layer***

This layer is responsible for application profiling, resource profiling and results possessing. The application profiler contributes to decision making by providing application information to the decision layer. The application profiler does application profiling based on the criteria such as application size, data size and application priority. If the application size is big or the data size is large, the application will be profiled as process-intensive and queued for the decision layer. If the application and data size is smaller, the task will be profiled as non-process-intensive, and it will be queued for local processing to the resource profiler. The resulting processor is responsible for receiving the processed task from the edge server and combining the results to produce the outcome.

## ***5.4 Application Layer***

The application layer is responsible for executing APIs and sending and receiving data from concerned applications to render the outcome on the user's screen. This

layer also sends a request to the service layer for application profiling. Application profiling involves classifying the applications based on application size, data size, application type (real time or non-real time), partitionable or non-partitionable. It focuses on end-user services and facilitates process-to-process connections.

## 6 Conclusion and Future Work

The previous research only considers energy consumption minimization or execution latency but ignored the network barriers and signal strength, which may play a significant role in offloading decisions and overall throughput. The limitations of these research work motivated us to study the computation further offloading in a multi-user and distributed load-sharing MEC environment in association with the resource-rich cloud, improving the mobile device's performance and enhancing the performance battery life. Our framework also considers network barrier and signal strength as significant parameters in offloading decisions, and optimizes. With the optimized offloading decision, the performance efficiency will be enhanced, and the mobile device's battery life will be improved.

1. In the future we have a plan to propose an algorithm considering various offloading criteria/parameters. We will use Multi-Criteria Decision Making (MCDM), specifically Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) for optimal decision making. It compares a set of alternatives based on a pre-specified criterion.
2. For validating the algorithm and conducting the experiments, we will set up the experimental environment using cloud and edge simulators that can simulate the processing, power consumption and latency for MEC environment.
3. For validating the experimental results, first, we will set a benchmark and use regression analysis. Further we will cross validate the results of the statistical analysis to generalize it on independent data set.

## References

1. Shakarami, A. Shahidinejad, and M. Ghobaei-Arani, "An autonomous computation offloading strategy in Mobile Edge Computing: a deep learning-based hybrid approach," *Journal of Network and Computer Applications*, vol. 178, 2021.
2. Asharul Islam, Anoop Kumar, Khalid Mohiuddin, Sadaf Yasmin, Mohammed Abdul Khaleel & Mohammad Rashid Hussain "Efficient resourceful mobile cloud architecture (mRARSA) for resource-demanding applications" *Journal of Cloud Computing* volume 9, Article number: 9 (2020)
3. Li Lin, Xiaofei Liao, Hai Jin, Peng Li, "Computation Offloading Toward Edge Computing", *Proceedings of the IEEE July 2019*, <https://doi.org/10.1109/JPROC.2019.2922285>
4. Edge Computing benefits and considerations for IoT and beyond, VEXCHANGE White paper 2020.

5. Asharul Islam, Anoop Kumar, Sadaf Yasmin “Computation-intensive offloading to cloud: concepts and challenges”, 2019 JETIR May 2019, Volume 6, Issue 5, [www.jetir.org](http://www.jetir.org) (ISSN-2349-5162).
6. Luobing Dong, Meghana N. Satpute, Junyuan Shan, Baoqi Liu, Yang Yu, Tihua Yan, “Computation Offloading for Mobile-Edge Computing with Multi-user”, 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)
7. Yun Chao Hu et al. “Mobile edge computing—A key technology towards 5G”. In: ETSI white paper 11.11 (2015), pp. 1–16.
8. Shanhe Yi, Cheng Li, and Qun Li. “A survey of fog computing: concepts, applications and issues”. In: Proceedings of the 2015 workshop on mobile big data. ACM. 2015, pp. 37–42.
9. Yuyi Mao et al. “A Survey on Mobile Edge Computing: The Communication Perspective”. In: IEEE Communications Surveys Tutorials PP.99 (2017), pp. 1–1.
10. Y. Mao, J. Zhang, and K. B. Letaief, “Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems,” in Proc. IEEE Wireless Commun. Netw. Conf. (WCNC), Mar. 2017, pp. 1–6.
11. L. Liu, Z. Chang, and X. Guo, “Socially aware dynamic computation offloading scheme for fog computing system with energy harvesting devices,” IEEE Internet Things J., vol. 5, no. 3, pp. 1869–1879, Jun. 2018.
12. J. Zhang, X. Hu, Z. Ning, E. C.-H. Ngai, L. Zhou, J. Wei, J. Cheng, and B. Hu, “Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks,” IEEE Internet Things J., vol. 5, no. 4, pp. 2633–2645, Aug. 2018.
13. A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, “On reducing IoT service delay via fog offloading,” IEEE Internet Things J., vol. 5, no. 2, pp. 998–1010, Apr. 2018.
14. H. Guo, J. Liu, J. Zhang, W. Sun, and N. Kato, “Mobile-edge computation offloading for ultradense IoT networks,” IEEE Internet Things J., vol. 5, no. 6, pp. 4977–4988, Dec. 2018.
15. F. Shan, J. Luo, J. Jin, and W. Wu, “Offloading delay constrained transparent computing tasks with energy-efficient transmission power scheduling in wireless IoT environment,” IEEE Internet Things J., vol. 6, no. 3, pp. 4411–4422, Jun. 2019.
16. G. Lee, W. Saad, and M. Bennis, “An online optimization framework for distributed fog network formation with minimal latency,” IEEE Trans. Wireless Commun., vol. 18, no. 4, pp. 2244–2258, Apr. 2019.
17. T. Yang, H. Feng, C. Yang, Y. Wang, J. Dong, and M. Xia, “Multivessel computation offloading in maritime mobile edge computing network,” IEEE Internet Things J., vol. 6, no. 3, pp. 4063–4073, Jun. 2019.
18. G. Zhang, F. Shen, Z. Liu, Y. Yang, K. Wang, and M.-T. Zhou, “FEMTO: Fair and energy-minimized task offloading for fog-enabled IoT networks,” IEEE Internet Things J., vol. 6, no. 3, pp. 4388–4400, Jun. 2019.
19. Z. Zhang, Z. Hong, W. Chen, Z. Zheng, and X. Chen, “Joint computation offloading and coin loaning for blockchain-empowered mobile edge computing,” IEEE Internet Things J., vol. 6, no. 6, pp. 9934–9950, Dec. 2019–Jan 2020
20. Farhan Sufyan, Amit Banerjee, “Computation Offloading for Distributed Mobile Edge Computing Network: A Multiobjective Approach”, IEEE Open Access Journal, August 2020.
21. Zhaolong Ning, Bin Hu, Xiaojie Wang, Tie Qiu, “Mobile Edge Computing Enabled 5G Health Monitoring for Internet of Medical Things: A Decentralized Game Theoretic Approach”, Article in IEEE Journal on Selected Areas in Communications February 2021.

# Road Surface Classification and Obstacle Detection for Visually Impaired People



Shripad Bhatlawande, Yash Aney, Aatreya Gaikwad, Vedant Anantwar, Swati Shilaskar, and Jyoti Madake

## 1 Introduction

Experts use the term “visual impairment” to describe any type of vision loss, from total blindness to partial blindness. Many individuals are legally blind, but some are completely blind. A decrease in visual acuity, in which the eye sees objects less clearly than usual, can cause vision impairment. It might also be caused by a loss of visual field, which occurs when the eye cannot see as much without moving the eyes or rotating the head as it used to.

All around the world, nearly more than two billion people are affected with near or far vision impairment [1]. Vision impairment may have been avoided or managed in at least one billion – or nearly half – of these cases.

On further research, the major cause of blindness (partial/complete) is cataracts or uncorrected refractive indexes. The major population above the age of 50 is affected with vision impairment (VI), but that does not mean that only the older generation is the one affected. The loss of vision can affect any age groups some of which may be well below 50 years of age. This also affects the global costs of productivity losses. Nearly two hundred billion USD are lost solely because of the issues related to impairment in vision, namely, myopia. The prevalence of blindness and MSVI (moderate to severe impairment) is substantially higher in older age groups since the risk of most eye disorders increases with age [2]. For visually challenged people, navigating around cities may be difficult and mentally exhausting, particularly when visiting places that are unappealing or unfamiliar.

---

S. Bhatlawande · Y. Aney · A. Gaikwad · V. Anantwar · S. Shilaskar (✉) · J. Madake  
Department of Electronics and Telecommunication, Vishwakarma Institute of Technology, Pune,  
India  
e-mail: [shripad.bhatlawande@vit.edu](mailto:shripad.bhatlawande@vit.edu); [yash.aney19@vit.edu](mailto:yash.aney19@vit.edu); [aatreya.gaikwad19@vit.edu](mailto:aatreya.gaikwad19@vit.edu);  
[vedant.anantwar19@vit.edu](mailto:vedant.anantwar19@vit.edu); [swati.shilaskar@vit.edu](mailto:swati.shilaskar@vit.edu); [jyoti.madake@vit.edu](mailto:jyoti.madake@vit.edu)

Despite the growing number of helpful technologies that help people with vision loss improve their consciousness and navigation skills while on the move, only a few systems provide a high level of independence and accuracy outside of familiar environments, allowing blind people to achieve significant mobility and integrate into daily active life [3]. To increase the accuracy and dependability of these systems, much research and study have been conducted. Numerous research has been developed to efficiently comprehend how individuals with visual acuity failure perceive and engage with the city area, as portrayed in their operations of cognitive load and stress, by placing the visually impaired at the center of attention and utilizing the latest events in physiological computing and smart wearable sensor embedded systems. There have been several blind aids/solutions invented and researched. Some are acoustic, while others are tactile, with a handful of non-wounded vision substitutes tossed in for good measure. Blindness causes perceptual compensation, which manifests as an over-performance of hearing in terms of physiology. In reality, the hearing of a blind individual is very sensitive than that of a normal human being. As per related physiologic studies, a highly skilled blind commuter can approach a crossroad, listen to the traffic, and evaluate the spatial structure of intersecting streets, the street width, the number of traffic lanes in each direction, and the existence of pedestrian islands or medians based exclusively on auditory stimuli. vOICe was a system made by a Dutch physicist Meijer. It worked in real time and was called vOICe [4] which meant “Oh, I can see.” It had a video camera which used to take images of the surroundings and convert them to a digital form which later would be given out as sound after conversion to the VI [5]. It was a sonic imaging system for the VI. It is a wearable device. The voice has proved to be very useful to the VI (visually impaired). The only limitation to this though is the fact that the time span to learn and master the device was very long. It is almost as if it is a new language [6] because the output of this device was a different sound pattern for every object [7]. Photodiode cane [8] was a device proposed by Joselin Villanueva which was a handheld device used to detect the correct path for the VI to walk through. It would find a protected path that was free of all the obstacles in the way and would provide vibratory feedback when the VI came close to any object. Similar canes which worked on Laser were also proposed as an aid for the blind, for instance, Teletact [9] was one such device. Sonic Torch [5, 10] and KASPA [11] were devices worked on by Leslie Kay. His work done is considerable when it came to solar mobility which would act as blind aid for the VI. The above utilize FM signals which would calculate/generate the distance of the object by the generated sound's pitch [5].

Leslie Kay's KASPA [5] invention had ultrasound FM emitters (sweep) and three sensors displaced laterally [12]. To create audible noises, the message received from the echo is compared to the signal sent out. The information about the reflected qualities of the item was transmitted by the timbre [11], which is inversely related to the frequency of the sound. The user, on the other hand, had to struggle for days to comprehend the thing. For better understanding, the Sonic Torch made by Leslie Kay was like what bats use to gather information about the object in their sight. The disadvantages though remained similar to the vOICe system. It also had a very long learning period. There were various other devices that worked on similar principles



of using ultrasounds; to name a few there was CyARM [13] and miniguide [14]. The ultrasonic cane [15, 16] was a handheld device that used ultrasound sensors to detect things on the ground and in the air. It was a relatively easy device to use as it would give out information through the speakers which were very easily understandable. Navigation Assistant for Visually Impaired [17] was a system made by R. Nagarajan and his fellow researchers which was a major advancement over vOICE. The VS (vision sensor) would capture information in the sight of the VI. After that, the picture was analyzed in order to determine the item in front of the individual. At the same time, the item was being identified using a real-time image processing approach based on fuzzy algorithms. It was a wearable device that had a better and improved image processing algorithm. Similar technology had been seen in other aids as well; all of them were wearables, some belts, and some bracelets while some were hats or fixed on a cane. For example, an electronic bracelet [18] was a similar technology in the form of a bracelet.

Navigation and narration system for the blind [19] was a device that had an ImageNet dataset that was trained and worked on a neural network that was convolution in nature. It would narrate detected objects and send the information to the visually impaired person. This was possible for any device having a camera like a smartphone and tablet. Just like the auditory senses of the blind, the haptic or touch sense of VIs also is heightened. Just like reading braille, which is the language for the blind, there are various haptic feedbacks generated on the displays which would give the visually impaired an idea about the image [12]. A very common 2D display was made by Kaczmarek which worked on closely packed pulsating electrodes giving the VI an idea about the image. It was named after his name Kaczmarek's electro-tactile display which had simply packed black and white pixels onto a matrix [20]. A wearable navigation system based on Arduino platform [19] which was developed by Saurav and Niketa Gandhi was a wearable that helped to assist the person with indoor surroundings. It used Arduino microcontroller as the brain and a very cheap camera as a CV (computer vision) camera along with several vibration motors to give out feedback. The main issue with this system was that the error would increase when the number of objects in the frame increased making it less accurate. All the previous devices mentioned either worked on an auditory or haptic feedback system which would convert images into other forms and given to the VI as a haptic or auditory signal, but according to today's trends, the latest VIDs (visually impaired devices) are a combination of both auditory and haptic senses which would not only improve the quality of the aid but also could be a significant improvement for the blind as their vision substitution devices [12]. This was later researched and carried on by a researcher named Mahdi Safaa who made a device that gave out both auditory and haptic feedback [8]. The proposed novel system uses computer vision for object detection and road surface recognition. It is easy to deploy and computationally in-expensive as compared to existing solutions. It is an easy to carry low-powered embedded system that can be carried in a bag pack. The proposed system detects obstacles in the range of up to 4 m to avoid information overloading. It is a method of assisting visually impaired individuals by guiding and notifying them to walk on the pavement rather than walking on a road, namely, tar and cement. It will guide him using voice commands through a Bluetooth earplug.

**Table 1** Comparison of technology in use

Name	Technique	Form	Feedback
vOICe [4]	Sonic imaging	Wearable	Auditory
Photodiode cane [8]	Photodiode	Cane	Tactile
Teletact [9]	Laser technology	Handheld	Tactile
CyARM [13], Miniguide [14]	Ultrasonic sensor	Wearable/handheld device	Auditory
NAVI [17]	Image processing and fuzzy algorithms	Wearable	Auditory
Navigation and narration system [19]	Convolutional neural network	Handheld	Auditory
2D display [20]	Closely packed pulsating electrodes	Micro actuator-based tactile graphic display	Electro-tactile
Proposed system	Computer vision	Cane	Auditory

The comparison table of existing electronic aids for the visually impaired is shown in Table 1.

The proposed novel system uses computer vision for object detection and road surface recognition. It is easy to deploy and computationally in-expensive as compared to existing solutions. It is a method of assisting visually impaired individuals by guiding and notifying them to walk on the pavement rather than walking on a road namely tar and cement.

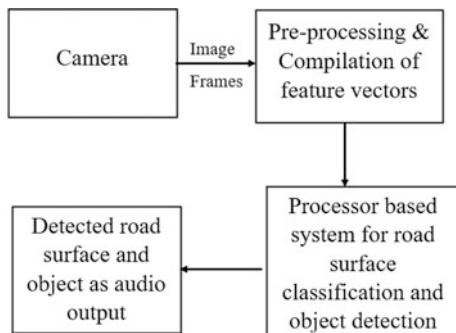
## 2 Methodology

The proposed system detects two types of roads (tar and cement) with and without obstacles. The block diagram of system is shown in Fig. 1. It consists of a camera, a processor-based system, and an earplug.

The camera acquires the information on the road and provides it as input to the portable computing system. It categorizes the detected road scene in (i) tar road without obstacles, (ii) tar road with obstacles, (iii) cement road without obstacles, and (iv) cement road with obstacles. It converts the detected scene into audio feedback and conveys it to the visually impaired via earplug.

### 2.1 Dataset and Preprocessing

The dataset used in the system was made by the authors. The dataset consists of 8000 images of which 50% of the images were captured using an iPhone 11 (13MP) camera, and the rest of the images were taken from the Internet [21]. The distribution of images in the dataset is shown in Table 2.

**Fig. 1** Road and obstacle detection system**Table 2** Number of images in each class

Type	No. of images	Class no.
Cement road with obstacle	2000	0
Cement road without obstacle	2000	1
Tar road with obstacle	2000	2
Tar road without obstacle	2000	3

The images were resized to  $400 \times 200$  pixels and were then further processed using Algorithm 1.

#### Algorithm 1: Algorithm for Image Preprocessing

```

input Image frames
output Preprocessed images
for every image in the directory do
  Gray scaling the image
  Resizing the images to 400 x 200 pixels
  Histogram equalization of images
  Cropping from 50 px to 400 px row wise
  Sharpening the images
  Apply global thresholding
end for
return preprocessed images
  
```

The scale-invariant feature transform (SIFT) was used on preprocessed images for feature extraction.

## 2.2 Feature Extraction and Dimensionality Reduction

SIFT descriptor is the best feature extraction algorithm for identifying and coordinating neighborhood features in the image. SIFT descriptor was used because it proves to be robust with issues like occlusion while being invariant to scaling [22, 23]. It identifies local features which are then utilized for model training. The rotation invariance of feature descriptors is achieved by assigning an orientation

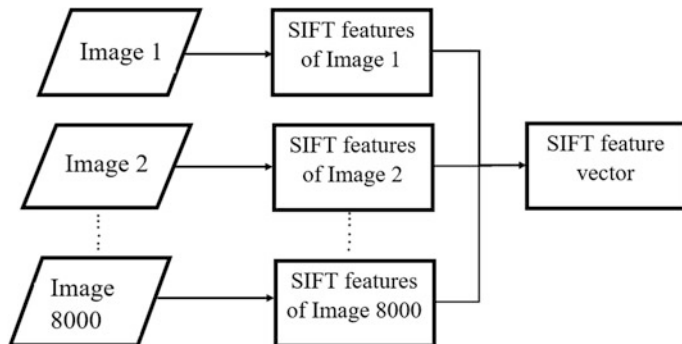


Fig. 2 SIFT feature extraction

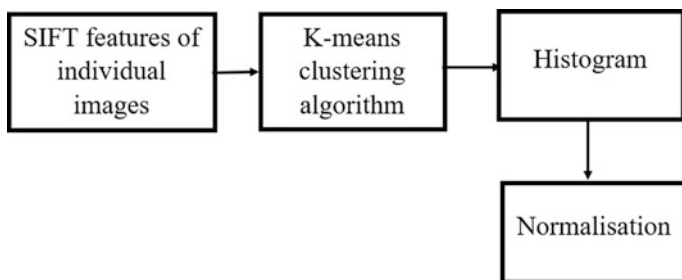


Fig. 3 Block diagram of k-means clustering

to key points. The amplitude and orientation of a pixel's gradient are defined as follows:

$$n(a, b) = \sqrt{((a+1, b) - L(a-1, b))^2 + (L(a, b+1) - L(a, b-1))^2}$$

$$\theta(a, b) = \tan^{-1} \left( \frac{L(a, b+1) - L(a, b-1)}{L(a+1, b) - L(a-1, b)} \right)$$
(1)

SIFT feature extraction was implemented on all 8000 images in the dataset as shown in Fig. 2, and an output feature vector was obtained. It was very large dimensions  $4,895,612 \times 128$ .

The number of rows was optimized by using k-means clustering [24] as shown in Fig. 3. The elbow technique was used to calculate the value of  $K$ . This yielded a result of 26. The acquired feature vector was then transformed into a 26-bin histogram.

**Algorithm 2: Algorithm for Image Preprocessing**

```

input Feature vector of size 4,895,612x128
output Optimized feature vector 8000x20

k-means clustering on SIFT feature vector where k =26
for every image in a specified path do
    Perform prediction on pre-trained k-means model
    Normalize data
end for
    Standardization using standard scalar
    Perform PCA with n_components= 20
    Perform PCA transform
return optimized feature vector of size 8000 x 20

```

This histogram was used to divide the 4,895,612 descriptors into 26 clusters. The SIFT features of every image were then predicted and grounded using a pre-trained k-means model to obtain a feature vector of  $8000 \times 26$ . The input feature vector's dimensionality was reduced using principal component analysis (PCA) [25, 26]. This approach optimized the feature vector from size  $8000 \times 26$  to a final feature vector of size  $8000 \times 20$ . The entire dimensionality reduction algorithm is shown in Algorithm 2. The final feature vector was then given as input to an array of four supervised machine learning classifiers to predict all four classes in the proposed novel system.

### 2.3 Classification

The resultant feature vector was given as input to four supervised machine learning classifiers: (i) SVM, (ii) decision tree, (iii) KNN, and (iv) random forest as shown in Fig.4.

The first classifier used was the support vector machine (SVM). It is a supervised machine learning algorithm used to find a hyperplane in an N-dimensional space that categorizes data points clearly. The hyperplane's size is determined by the number of features. It improves the ability to generalize [27]. The equation to maximize the margin between the classes is given in Eq. 2.

$$L(w) = \sum_{i=1} \max \left( 0, 1 - y_i \left( w^T x_i + b \right) \right) + \lambda \|w\|_2^2 \quad (2)$$

whereas  $\lambda = 1/c$ .  $c$  is a regularization parameter.  $w$  is a weight vector and  $b$  is bias.

The second classifier used was decision tree (DT). It is a hierarchical classification system for items based on a set of principles [28]. When provided data about objects with attributes and classes, the DT technique generates rules to forecast any unseen data.

The third classifier used was k-nearest neighbors (KNN). This is a method to classify a data point "p," its k closest neighbors are retrieved by finding Euclidean distance given by the formula (3), resulting in a "p" neighborhood [29, 30].

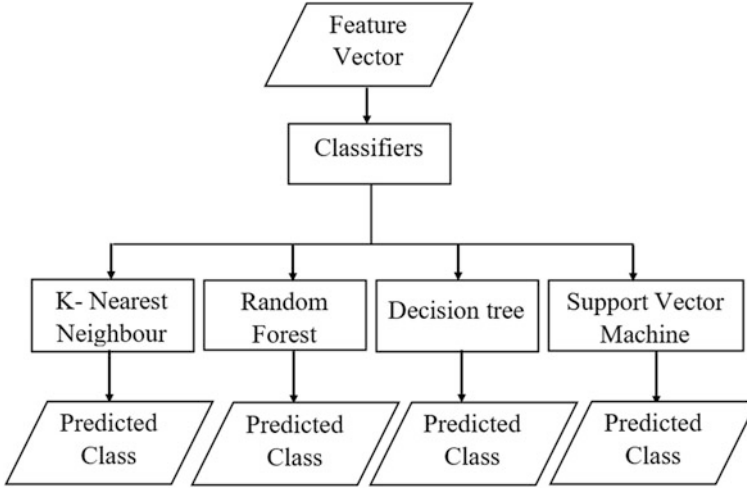


Fig. 4 Different classifiers used to predict classes in the proposed model

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3)$$

The fourth classifier used was random forest, in which the output class is defined by the mode of the response class provided by the trees. The input is processed by a forest of decision trees [28, 31]. The prediction  $Y$  of  $n$  trees, each with its own weight function  $M_i$ , averaged in a random forest is given as (4) [32].

Euclidean distance between  $p(x_1, y_1)$  and the  $k$ -neighbors =

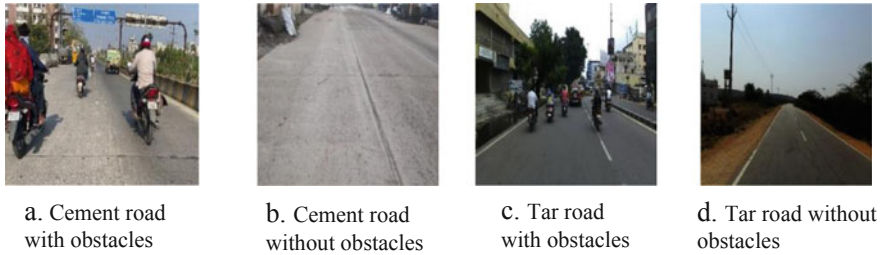
$$Y = 1/n \sum_{i=1}^n \sum_{j=1}^m M_i(x_j, x') y_j \quad (4)$$

The performance of all the abovementioned classifiers was recorded. The accuracy, precision score, and F1 score are mentioned in the following section.

### 3 Results

Four supervised machine learning models were used for classification, namely, KNN, SVM, decision tree, and random forest. These four classifiers were able to detect accurately the obstacles from the image frame. Their accuracies were then compared in Table 2. Figure 5 shows the images classified as classes described in Table 1. The results are discussed below.

KNN was used with different values of  $k$  such as  $k = 5$ ,  $k = 20$ ,  $k = 1$ , and  $k = 11$ , and accuracies were noted down. With the value of  $k = 1$ , the accuracy came out



**Fig. 5** Road surface classes. (a) Cement road with obstacles (b) Cement road without obstacles (c) Tar road with obstacles (d) Tar road without obstacles

**Table 3** Classifier performance

Classifier	Accuracy (%)	Precision	Recall	F1 score
KNN	83.93	0.839	0.834	0.832
Random forest	88.54	0.885	0.883	0.883
Decision tree	83.03	0.830	0.827	0.823
SVM	85.14	0.851	0.851	0.847

to be equal to 88.54%, and the model has achieved maximum accuracy. Similarly, for other values of  $k$ , the accuracy was slightly less than the accuracy value of the classifier when  $k = 5$ . So, when the optimal value of  $k$  is equal to 5, the accuracy of the algorithm is nearly 83.93%. The precision of the model was nearly equal to 0.839 when  $k$  was equal to 5, and the F1 score was equal to 0.832 and recall is 0.834. Further, the decision tree classifier was used, and the training accuracy was equal to 83.03% with  $\text{max\_depth} = 35$ . The precision value was equal to 0.830, while the recall value was equal to 0.827. Therefore, after using the values of recall and precision, the F1 score was equal to 0.823. Random forest was then used as the classifier, and the accuracy score was 88.54%. Experimentation with different  $n\_estimator$  values had been conducted and found that the results were inconsistent. The number of trees in the random forest, or  $n\_estimators$ , is proportional to the number of rows in the dataset. As a result, the value of  $n\_estimator$  was chosen, with the highest accuracy achieved when the value of the  $n\_estimator$  was equal to 31 and the accuracy score was 88.54%. Precision was nearly equal to 0.885, recall equal to 0.883, and F1 score equal to 0.883 which is close to 1 which means the model is showing good results. The accuracy score obtained after applying SVM to the model with the RBF kernel was 85.14%. Experimentation with all three kernels, namely, linear, sigmoid, and Gaussian, was conducted. However, when the accuracy score of SVM with RBF kernel was compared to other kernels, the accuracy score of SVM with RBF kernel was higher. The precision, recall, and F1 scores for the RBF kernel after selecting it for classification in SVM were 0.851, 0.848, and 0.847, respectively (Table 3).

After evaluating the performances of all four classifiers, random forest had the highest accuracy. Therefore, random forest with  $n\_estimator$  value equal to 31 is

the best classifier for the proposed model with training accuracy equal to 88.54%. Differentiating between tar and cement road is the major challenge while building the model. Though the accuracy score of random forest is nearly 89%, this score can be improved by improving the quality of the dataset.

## 4 Conclusion

The proposed novel system is a method of assisting visually impaired people by continuously guiding and notifying them to keep them walking on a sidewalk rather than a tar or cement road. The proposed system can be mounted on the cane/cap of the blind person which would guide him with the help of voice commands using the Bluetooth earplug. The model was trained with a huge amount of data on different machine learning algorithms and was tested with four classifiers, namely, random forest, KNN, and support vector machine. Out of which random forest has the highest accuracy and F1 score. All the gadgets discussed in the literature are either costly, unreliable, or require blind people to learn an entirely new language, making them more difficult and time-consuming to operate. Our prototype solves all the concerns described above, making it the most dependable and go-to tool for the visually handicapped. The work proposed in this paper has a training accuracy score of 88.54%. This accuracy is quite satisfactory, considering the lack of visual distinction between the tar and cement surface of the road. The accuracy can be increased if the images of the dataset are collected under very ideal climatic conditions. Ideally, the images should be captured during a clear and sunny day and during the time of mid-day. Special care should be taken while considering the angle of the camera, as after a certain angle both cement and tar roads tend to reflect the sunlight, thus diminishing the distinction between both surfaces of the roads. Another important consideration is the presence of liquid on the road surface. The presence of the water makes the concrete road look similar to the asphalt roads. The most eminent problem regarding the roads found in India and other developing countries is that a prominent number of roads are made up of a combination of asphalt and bitumen. This makes it difficult to find the visual distinction between the road surfaces. Therefore, if the conditions and factors are taken into consideration while improving the quality of the dataset, the result would improve.

## References

1. World Health Organization. (2018). Vision impairment and blindness. [online]. <http://www.who.int/newsroom/factsheets/detail/blindness-and-visual-impairment>
2. Manduchi, Roberto, and James Coughlan. "(Computer) vision without sight." *Communications of the ACM* 55, no. 1 (2012): 96–104.
3. Takeshi Yashiro, Shinsuke Kobayashi, Noboru Koshizuka and Ken Sakamura, "An Internet of Things (IoT) Architecture for Embedded Appliances", Electrical and Control Engineering (ICECE), 2011 International Conference, Yichang, IEEE, 2011, pp. 2578–2581.



4. Meers, Simon, and Koren Ward. "A vision system for providing 3D perception of the environment via transcutaneous electro-neural stimulation." In *Proceedings. Eighth International Conference on Information Visualisation, 2004. IV 2004.*, pp. 546–552. IEEE, 2004.
5. <http://sonicvision.co.nz/> May.
6. Dakopoulos, Dimitrios, and Nikolaos G. Bourbakis. "Wearable obstacle avoidance electronic travel aids for blind: a survey." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 40, no. 1 (2009): 25–35.
7. Kuc, Roman. "Binaural sonar electronic travel aid provides vibrotactile cues for landmark, reflector motion and surface texture classification." *IEEE transactions on biomedical engineering* 49, no. 10 (2002): 1173–1180.
8. Villanueva, Joselin, and René Farcy. "Optical device indicating a safe free path to blind people." *IEEE transactions on instrumentation and measurement* 61, no. 1 (2011): 170–177.
9. Farcy, René, Roger Leroux, Alain Jucha, Roland Damaschini, Colette Grégoire, and Aziz Zogaghi. "Electronic travel aids and electronic orientation aids for blind people: technical, rehabilitation and everyday life points of view." In *Proceedings of the Conference and Workshop on Assistive Technology for Vision and Hearing Impairment*. 2006.
10. Kay, L. "Electronic aids for blind persons: an interdisciplinary subject." *IEE Proceedings A (Physical Science, Measurement and Instrumentation, Management and Education, Reviews)* 131, no. 7 (1984): 559–576.
11. Kay, Leslie. "Auditory perception of objects by blind persons, using a bioacoustic high resolution air sonar." *The Journal of the Acoustical Society of America* 107, no. 6 (2000): 3266–3275.
12. Liu, Jihong, and Xiaoye Sun. "A survey of vision aids for the blind." In *2006 6th World Congress on Intelligent Control and Automation*, vol. 1, pp. 4312–4316. IEEE, 2006.
13. Ito, Kiyohide, Makoto Okamoto, Junichi Akita, Tetsuo Ono, Ikuko Gyobu, Tomohito Takagi, Takahiro Hoshi, and Yu Mishima. "CyARM: an alternative aid device for blind persons." In *CHI'05 Extended Abstracts on Human Factors in Computing Systems*, pp. 1483–1488. 2005.
14. Dakopoulos, Dimitrios, and Nikolaos G. Bourbakis. "Wearable obstacle avoidance electronic travel aids for blind: a survey." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 40, no. 1 (2009): 25–35.
15. Kumar, Krishna, Biswajeet Champaty, K. Uvanesh, Ripunjay Chachan, Kunal Pal, and Arfat Anis. "Development of an ultrasonic cane as a navigation aid for the blind people." In *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, pp. 475–479. IEEE, 2014.
16. Mahdi Safaa, A., H. Muhsin Asaad, and I. Al-Mosawi Ali. "Using ultrasonic sensor for blind and deaf persons combines voice alert and vibration properties." *Research Journal of Recent Sciences*
17. Nagarajan, R., Sazali Yaacob, and G. Sainarayanan. "Role of object identification in sonification system for visually impaired." In *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region*, vol. 2, pp. 735–739. IEEE, 2003.
18. Strakowski, Marcin R., Bogdan B. Kosmowski, Ryszard Kowalik, and Pawel Wierzba. "An ultrasonic obstacle detector based on phase beamforming principles." *IEEE Sensors Journal* 6, no. 1 (2006): 179–186.
19. Gandhi, Saurav, and Niketa Gandhi. "A CMUcam5 computer vision-based Arduino wearable navigation system for the visually impaired." In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1768–1774. IEEE, 2018.
20. Kacznaarek, KURT A., and Paul Bach-Y-Rita. "Tactile displays." In *Virtual environments and advanced interface design*, pp. 349–414. Oxford University Press, 1995.
21. Rateke, Thiago, Karla Aparecida Justen and Aldo von Wangenheim. "Road Surface Classification with Images Captured From Low-cost Camera – Road Traversing Knowledge (RTK) Dataset." *RITA* 26 (2019): 50–64.
22. M. Wei and P. Xiwei, "WLIB-SIFT: A Distinctive Local Image Feature Descriptor," *2019 IEEE 2nd International Conference on Information Communication and Signal Processing (ICICSP)*, 2019, pp. 379–383, <https://doi.org/10.1109/ICICSP48821.2019.8958587>.

23. Z. -S. Ni, "B-SIFT: A Binary SIFT Based Local Image Feature Descriptor," *2012 Fourth International Conference on Digital Home*, 2012, pp. 117–121, <https://doi.org/10.1109/ICDH.2012.69>
24. I. El rube', "Image Color Reduction Using Progressive Histogram Quantization and Kmeans Clustering," *2019 International Conference on Mechatronics, Remote Sensing, Information Systems and Industrial Information Technologies (ICMRSISIT)*, 2019, pp. 1–5, <https://doi.org/10.1109/ICMRSISIT46373.2020.9405957>.
25. A. Alkandari and S. J. Aljaber, "Principle Component Analysis algorithm (PCA) for image recognition," *2015 Second International Conference on Computing Technology and Information Management (ICCTIM)*, 2015, pp. 76–80, <https://doi.org/10.1109/ICCTIM.2015.7224596>.
26. S. Sehgal, H. Singh, M. Agarwal, V. Bhasker and Shantanu, "Data analysis using principal component analysis," *2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom)*, 2014, pp. 45–48, <https://doi.org/10.1109/MedCom.2014.7005973>.
27. Byun, Hyeran, and Seong-Wan Lee. "Applications of support vector machines for pattern recognition: A survey." In *International workshop on support vector machines*, pp. 213–236. Springer, Berlin, Heidelberg, 2002.
28. Le, Tuan Minh, Ly Van Tran, and Son Vu Truong Dao. "A Feature Selection Approach for Fall Detection Using Various Machine Learning Classifiers." *IEEE Access* 9 (2021): 115895–115908.
29. Chowdhury, Shovan, and Marco P. Schoen. "Research Paper Classification using Supervised Machine Learning Techniques." In *2020 Intermountain Engineering, Technology and Computing (IETC)*, pp. 1–6. IEEE, 2020.
30. M. Agarwal, K. K. Rao, K. Vaidya and S. Bhattacharya, "ML-MOC: Machine Learning (kNN and GMM) based Membership determination for Open Clusters," in *Monthly Notices of the Royal Astronomical Society*, vol. 502, no. 2, pp. 2582–2599, Jan. 2021, <https://doi.org/10.1093/mnras/stab118>.
31. B. Wang, L. Gao and Z. Juan, "Travel Mode Detection Using GPS Data and Socioeconomic Attributes Based on a Random Forest Classifier," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1547–1558, May 2018, <https://doi.org/10.1109/TITS.2017.2723523>.
32. J. K. Jaiswal and R. Samikannu, "Application of Random Forest Algorithm on Feature Subset Selection and Classification and Regression," *2017 World Congress on Computing and Communication Technologies (WCCCT)*, 2017, pp. 65–68, <https://doi.org/10.1109/WCCCT.2016.25>.

# A Survey on Semantic Segmentation Models for Underwater Images



Sai Krishna Anand, Pranav Vigneshwar Kumar, Rohith Saji, Akhilraj V. Gadagkar, and B R Chandavarkar

## 1 Introduction

Semantic segmentation of images involves a pixel-level classification of an image [1]. It strives to classify each pixel to a certain class from the dataset. If there are multiple objects from the same class in the image, then each pixel of these objects is classified and output in the same color. It has been a key field and a very important research direction of computer vision. This concept has been widely used in different fields and is the topic of extensive research, with applications in medicine [2], autonomous vehicles, and even agricultural science. However, studies on the use of semantic segmentation for underwater applications have been far less when compared to its applications in other fields. Improving the models used for terrestrial images is important as this allows real-time applications such as exploring probes and underwater vehicles to work that much more efficiently and smoother. The performance of traditional segmentation techniques often suffers due to issues such as poor lighting and contrast and hence needs to be adapted to specific requirements of underwater images by making certain image enhancements [3].

The ocean is rich in diverse biological resources, energy, and metals. The main goal of underwater expeditions is to explore and research the huge amount of marine biological resources to fulfill various demands. Semantic segmentation [1] can be extremely beneficial to several underwater applications, especially in underwater exploration missions with probes and robots. Segmentation of underwater images is important since it can aid in the automation of various processes that allow more efficient exploration of the vast expanses of the ocean [4].

---

S. K. Anand (✉) · P. V. Kumar · R. Saji · A. V. Gadagkar · B. R. Chandavarkar  
Department of Computer Science and Engineering, National Institute of Technology Karnataka,  
Surathkal, India

In the context of semantic segmentation, numerous models have been explored starting from simple, fully connected networks to encoder–decoder networks such as the U-Net. Faster-region-based convolutional neural network (R-CNN) is another popular model that has previously been used for semantic segmentation successfully. However, for underwater images, some of these models perform very poorly, and the SUIM paper shows that 6 models—SUIM-Net [5], SegNet [6], U-Net [2], DNN-VGG (Deep Neural Network-VGG) [7], DeepLabv3+ [8], and PSP-Net (Pyramid Scene Parsing Network) [9]—work better than other elementary models. With this as the basis, the details of these 6 models are explored with respect to semantic segmentation of underwater images. The primary aim of this survey is to gain insights on each model, understand their differences and novelties, and recognize the working of these models for underwater scenes.

The remainder of this chapter is organized as follows. Section 2 contains a technical overview of the different models beginning with their architectures, unique characteristics, and finally exploring how these models would work for underwater images. Section 3 presents a comparative analysis of the models that include the advantages and disadvantages followed by a comparison of the model characteristics. Finally, Sect. 4 includes the conclusions drawn.

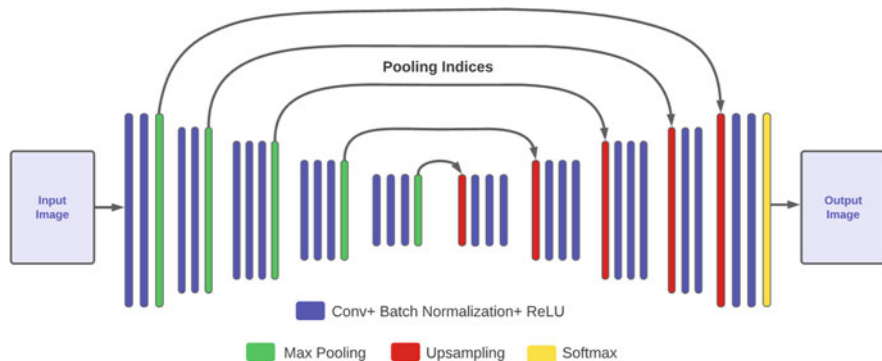
## 2 Semantic Segmentation Models

In this section, the technical overview of SegNet, PSP-Net, U-Net, DNN-VGG, DeepLabv3+, and SUIM-Net is presented. The details of their architectures along with their unique features are explored. Finally, the use of these models specifically in underwater scenarios is also discussed.

### 2.1 *SegNet*

SegNet includes an encoder network and a decoder network, as well as a pixel-by-pixel classification layer [6]. The encoder network comprises 13 convolutional layers, as seen in Fig. 1, which corresponds to the VGG-16 network's first 13 convolutional layers for object categorization [10]. The entirely connected layers at the deepest encoder output are discarded in favor of keeping higher resolution feature maps. Since each encoder layer has a corresponding decoder layer, the decoder network comprises 13 layers. The output of the decoder is then forwarded to a multiclass soft-max [11] layer that does the pixel-wise classification.

The novelty of SegNet is in the subsampling [12] stage, where max-pooling [13] is utilized to achieve translation invariance [14] over tiny spatial shifts in the image, and when it is combined with subsampling, each pixel governs a larger input image context (spatial window) [15]. These methods improve classification accuracy while reducing feature map size, resulting in a lossy image representation with



**Fig. 1** SegNet Architecture [6]

blurring boundaries, which is ineffective for segmentation. These methods improve classification accuracy while reducing feature map size, resulting in a lossy image representation with blurring boundaries, which is ineffective for segmentation.

SegNet performs upsampling [16] in its decoder, in order to ensure that the input and output images have the same resolution, and to achieve this in a space-efficient manner, it stores the max-pooling indices from the encoder network [15]. This does cause a slight loss in precision [17].

Although the first encoder's input has three channels, the decoder that corresponds to it creates a multi-channel feature map (RGB). This is different from other decoders in the network that produce feature maps with the same number of channels as their encoder inputs. A trainable soft-max [11] classifier is given the high-dimensional feature representation at the output of the final decoder. The soft-max classifier produces a "K" channel image of probabilities, where "K" is the number of classes. The class with the highest probability at each pixel corresponds to the expected segmentation.

When comparing SegNet with other models, it was found that models that save all of the encoder network feature maps in full perform the best, but they use more memory during inference [6]. SegNet, on the other hand, is more efficient since it just saves the feature maps' max-pooling indices and uses these in its decoder network to achieve good performance. SegNet performs competitively on huge and well-known datasets.

A drawback with SegNet is that due to not having fully connected layers, it may not learn as many feature mappings as other models. This drops the SegNet's performance below others in the same category. Hence, in order to apply SegNet on underwater images, it can be coupled with ResNet, which results in better segmentation but increases the computational cost [5]. Other work [18] includes adding a feature pyramid [19] to the encoder network that uses edge detection operators [20] to obtain boundary information and combines it with the multiscale features [21] to improve identification of smaller objects in the image.

## 2.2 PSP-Net

The PSP-Net [9] architecture considers the image's global context when predicting local-level predictions, resulting in improved performance on benchmark datasets such as PASCAL Visual Object Classes (VOC) 2012 [22] and Cityscapes [23]. The development of PSPNet was driven by the fact that fully convolutional network (FCN) [24]-based pixel classifiers were unable to capture the context of the entire image.

The PSP-Net model is nothing more than an encoder. A convolutional neural network (CNN) backbone is included, as well as the pyramid pooling module. The standard convolutional layers are substituted with dilated convolutional layers [25] in the backbone's last layers, which help to increase the receptive field [26, 27]. The dilated convolution layers are found in the backbone's last two blocks. As a result, the feature at the end of the backbone has more features. When doing convolution, the value of dilation indicates the sparsity. When compared to ordinary convolution, the receptive field of dilated convolution is greater. The amount of context information used is determined by the size of the receptive field. The dilation values of the last two blocks of the backbone in PSP-Net are 2 and 4, respectively.

The pyramid pooling module [9] is the most important aspect of this model since it allows the model to collect the image's global context, which allows it to classify pixels based on the image's global information. The feature map from the backbone is pooled at various sizes, then passed through a convolution layer, and then upsampled to make the pooled features the same size as the original feature map. The upsampled maps are then concatenated with the original feature map before being sent to the decoder. This method combines features from many scales, hence aggregating the total context.

For example, the four colors in Fig. 2 from the study [9] represent distinct scales: 6, 3, 2, and 1 for green, blue, orange, and red, respectively. To minimize the feature depth, the feature map is pooled at these sizes and then convolved with  $1 \times 1$  filters. After that, all of these features are concatenated and upsampled to the size of the

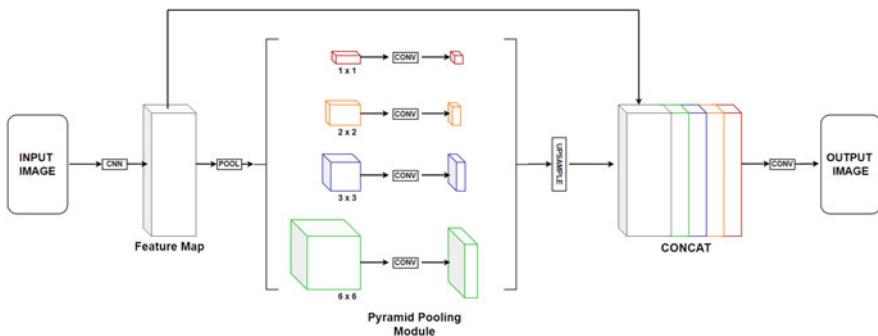


Fig. 2 PSP-Net Architecture [9]

feature map. The  $1 \times 1$  feature captures all of the information in a single  $1 \times 1$  spatial position; however, when the spatial resolution increases, high-resolution features are also taken into account, such as the  $6 \times 6$  pyramid size.

Figure 2 represents the working of the PSP-Net Model. The input image is initially passed through a CNN that helps to obtain the feature map of the last convolutional layer. Then the pyramid pooling module is used to extract the various sub-region representations, which are then upsampled and concatenated to generate the final feature representation, which includes both local and global context information. This feature representation is finally fed into a convolutional layer that does the pixel-wise classification.

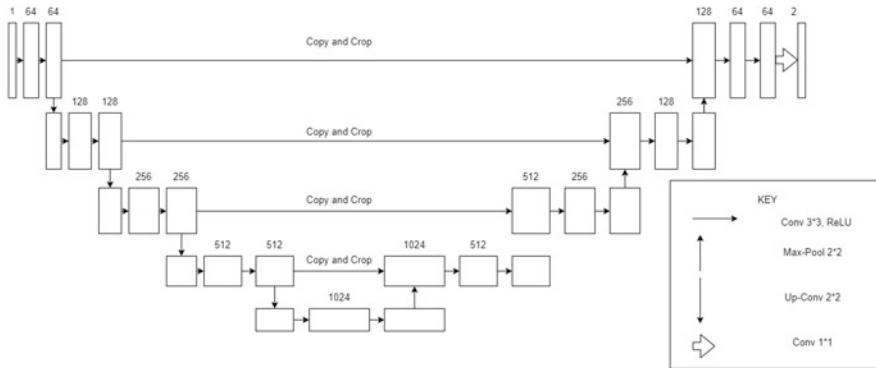
Since the feature map is pooled in different sizes in the pyramid pooling module, both high- and low-resolution features are taken into account that can be advantageous for underwater applications. The concatenation operations in pyramid pooling module can sometimes introduce noise and redundancy in the output. PSP-Net is also quite expensive computationally and requires a lot longer to process images than other models such as U-Net. An improvement performed on the PSP-Net included fusing the multi-layer image features extracted from the pyramid image features [28]. This retained the original network's ability to improve the effect of the receptive field as well as the detail features of the image through the fusion of multi-layer data.

### 2.3 U-Net

Fischer P et al. [2] proposed the U-Net architecture. It was originally constructed for biomedical images, but its application now lies far beyond just the medical field. The U-Net model has 2 parts—the encoder and decoder. Unlike other encoder–decoder networks, U-Net architecture uses skip connections to get additional information from the encoder network. Due to its symmetrical “U” shape, it is called U-Net. Skip connections were introduced to solve different problems in different architectures [2, 29, 30].

By using a skip connection, an alternative path is provided for the gradient (with backpropagation). It is experimentally validated that these additional paths are often beneficial for the model convergence. Skip connections in deep architectures, as the name suggests, skip some layer in the neural network and feed the output of one layer as the input to the next layers (instead of only the next one) (Fig. 3).

U-Net uses long skip connections [30]. Long skip connections often exist in architectures that are symmetrical, where the spatial dimensionality [16] is reduced in the encoder part and is gradually increased in the decoder part as illustrated above. In the decoder part, one can increase the dimensionality of a feature map via transpose convolutional layers [31]. The transposed convolution operation forms the same connectivity as the normal convolution but in the backward direction. By introducing skip connections in the encoder–decoder architecture, fine-grained details can be recovered in the prediction. Even though there is no theoretical



**Fig. 3** U-Net Architecture [2]

justification, symmetrical long skip connections work incredibly effectively in dense prediction tasks (medical image segmentation), where a pixel-wise classification is performed.

The encoder network consists of 4 blocks of convolution and max-pooling layers [32, 33]. Each of these layers produces more feature maps and gains information about the image on what are the objects/instances in the images. However, the shape of the layers will be completely changed, and hence, the model loses the spatial location of the extracted features. To recover these spatial features, i.e., “where” the instances are, the decoder network spatially upsamples the feature maps.

Each block in the decoder consists of convolutional and upsampling layers to recover the decoded spatial information. The decoder network is symmetrical to the encoder network and consists of skip connections from the encoder layer at the same level by concatenation of the 2 maps [32, 33]. This helps the decoder in learning better precise locations with respect to the initial feature maps of the encoder. Once the concatenation is finished, they are passed through a double convolution and an upsampling layer to get a more precise output.

As mentioned earlier, the skip connections are what provides the U-Net its good performance, and they are equally important whether the data is terrestrial or underwater images. The U-Net was tested on the Fish4Knowledge dataset, consisting of underwater images of fish and their segmentation masks [34]. It performs with an average F1 score of 0.92. Clearly, the U-Net is able to adapt well to underwater images, owing to the skip connections providing additional spatial information in the decoder network. Other enhancements such as histogram equalization [35, 36], pixel transformations [37, 38], etc., can be made directly to the images, to improve their contrast, brightness, and other image characteristics.

Base U-Nets can be improved by using dilated convolutions, as mentioned under PSP-Net. This can expand the receptive field without affecting the image resolution. Another drawback of U-Nets is that their optimal depth can only be found by repeated trial and error, with skip connections only between mirroring layers creating a restrictive framework. This was dealt with in U-Net++ proposed



by Zhou et al. [39], where an efficient ensemble of networks partially sharing an encoder was created to calculate the optimal depth. The U-Net++ paper [39] accumulated different features of varying semantic scales at different decoder sub-networks, leading to a highly flexible feature fusion scheme.

### 2.4 DNN-VGG

Zhou et al. [7] proposed a deep neural network architecture that consists of encoder and decoder networks to specifically perform the task of underwater semantic segmentation. This model, when applied to real-time videos, has better generalization ability when compared to SegNet, leading to better performance on unseen data. It is also less intensive on the memory during the training phase when compared to U-Net [2]. The data used to train the model was collected by Witted Srl, an underwater research company based in Italy.

This model, similar to SegNet [6] and U-Net [2], eliminates the requirement of fixed-size inputs by removing the fully connected layers from the architecture and including the encoder–decoder structure. The encoder network borrows 13 convolutional layers from VGG-16 [40] with weights that have been pre-trained using the ImageNet [41] dataset. A batch normalization [42] and ReLU (Rectified Linear Unit) layer [43] is also added after each convolutional layer in order to improve the efficiency and speed up the training of the model. The 13 convolutional layers in the encoder structure are split into 5 different blocks, as shown in Fig. 4, with a max-pooling [13] layer at the end of each block. This max-pooling layer serves the function of reducing the size of the feature maps.

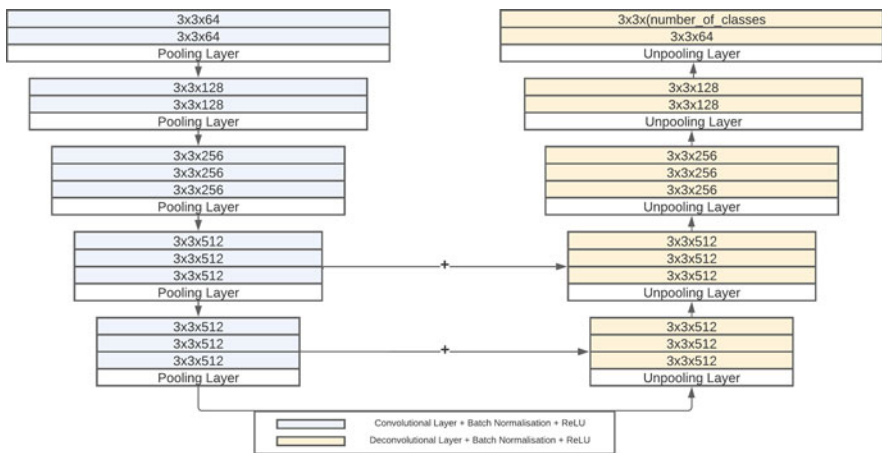


Fig. 4 DNN-VGG Architecture [7]

The decoder network, just like the encoder network, is made up of 13 deconvolutional [24, 44] layers divided into 5 blocks. It is the mirror image of the encoder structure. Each block begins with an un-pooling [6] layer followed by the deconvolutional layers. The max un-pooling operation works by using the stored indices of the max values during the pooling stage to map the input to the output positions. The undefined positions are mapped to zero in this process. No learning actually occurs in the un-pooling layers due to the lack of any trainable parameters. Research has indicated that the lack of learning does not negatively impact performance when compared to using filters with a large number of parameters that can be updated [6]. The primary purpose of using the un-pooling operation here is to ensure that the spatial information of the feature maps is maintained.

This architecture also borrows a feature from U-Net [2] wherein the feature maps are transferred from the encoder to the decoder structure using the concatenation operation. Concatenation is a useful feature as it allows the decoder network to learn directly from the feature maps using the information learned by the encoder. The concatenations are limited to the last two blocks of the encoder network in order to reduce the memory requirements during the training phase.

## 2.5 *DeepLabv3+*

DeepLabv3+ [8] is a state-of-the-art semantic segmentation model developed by a group of researchers from Google terrestrial applications in mind. DeepLabv3+ extends DeepLabv3 [45] by employing an encoder–decoder structure where DeepLabv3 is used as the encoder module and a simple decoder module is added to improve the sharpness and refine the object boundaries in the segmentation result. The results of this model have been extremely promising with it attaining test set performances of 82.1% and 89.0% on the Cityscapes [23] and PASCAL VOC 2012 [22] datasets.

The encoder module, as shown in Fig. 5, is structured in such a way that a backbone network, such as ResNet [46], Xception [47], or VGG [40], is first used to extract features from the input image. The feature maps that are obtained from the backbone network are then passed onto the Atrous Spatial Pyramid Pooling (ASPP) [27] module.

The ASPP module uses the concept of atrous convolution [26, 48], which was first presented in DeepLab [27] as a way to adjust the resolution of extracted features as well as modify the effective field of view of the convolution. The field of view can be adjusted by varying a parameter known as “atrous/dilation rate,” which defines the spacing between the values in a kernel. It is a basic yet effective method to make the field of view of filters larger, without negatively affecting computation times or the number of parameters. It is particularly useful for semantic segmentation since it preserves spatial resolution and allows us to build a deeper network by capturing features at different scales [49].

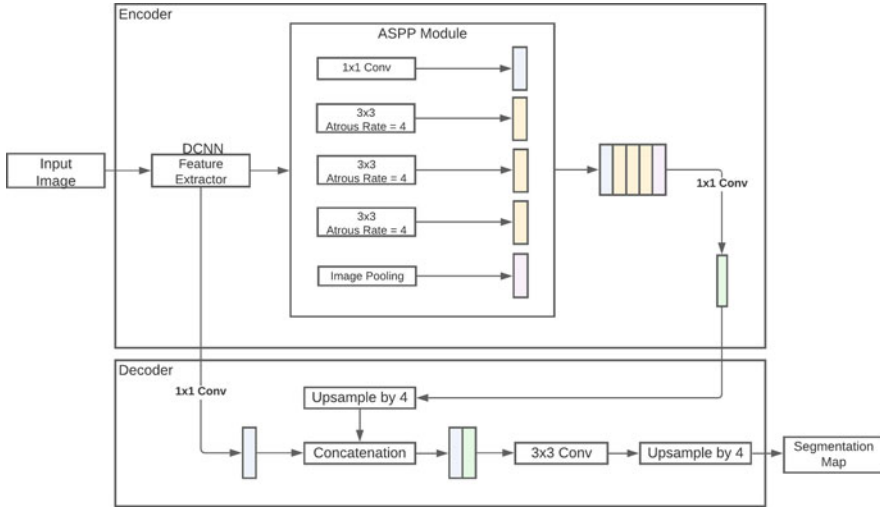
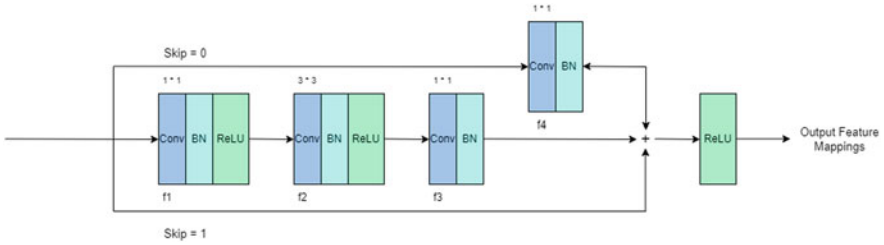


Fig. 5 DeepLabv3+ Architecture [8]

The primary motive behind the ASPP module is to obtain multiscale contextual information. When the feature map from the backbone network is input into the ASPP module, four parallel atrous convolutions with different dilation rates are applied to handle segmenting the object at different scales. The final step in the encoder network is to apply a  $1 \times 1$  convolution on the output from the ASPP module.

In the decoder part of the model, the first step is to bilinearly upsample the output from the encoder by a factor of 4. Then this upsampled output is concatenated with the  $1 \times 1$  convolution of the low-level features. After the concatenation, a few  $3 \times 3$  convolutions are applied to refine the features followed by another simple bilinear upsampling by a factor of 4. This will give us the prediction for the semantic segmentation [8].

As DeepLabv3+ was developed with terrestrial images as its primary aim, the model underperforms when directly used for underwater images. It suffers from poor segmentation of the target boundary, lack of refined edges, and classification errors. In line with this, Liu et al. [4] proposed an extension to the DeepLabv3+ network to specifically adapt the model for underwater images. The decoder module of DeepLabv3+ was modified to include two additional upsampling layers where the inputs are upsampled by a factor of 2 instead of 4. Reducing the factor of upsampling to 2 decreases the amount of feature information lost. Connections between the low-level features from the backbone network and high-level feature information were also added at each upsampling stage to tackle the issue of lack of contour and boundary information. These modifications address the drawbacks of the original decoder module with respect to feature retention and poor boundary segmentation of target objects. Further, the images were also pre-processed using the unsupervised



**Fig. 6** Residual skip block of SUIM-Net [5]

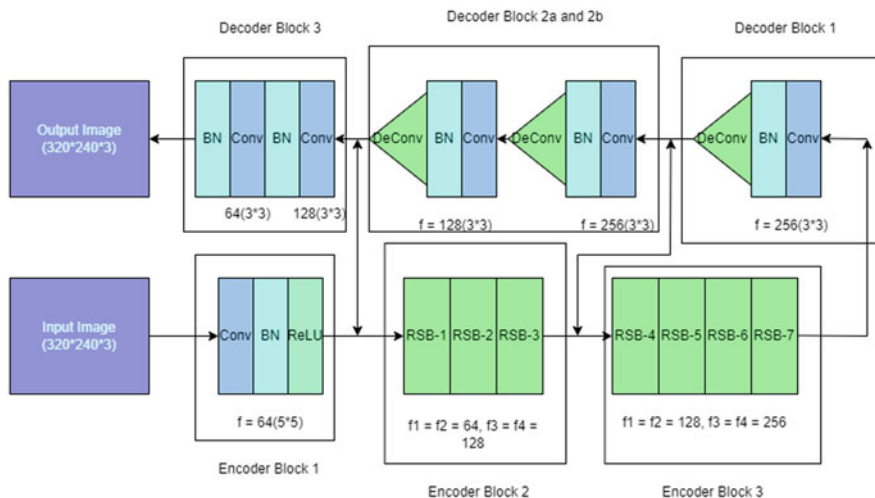
color correction method (UCM) [50] as the images enhanced with this method allow for higher feature extraction and ultimately lead to better segmentation maps.

## 2.6 SUIM-Net

Islam et al. [5] proposed the SUIM (Semantic Segmentation for Underwater Imagery)-Net, which is one of the newer models created specifically for underwater image segmentation. It is so named because it is trained on the SUIM dataset [51], which consists of around 1500 images with 8 different classes and has produced better results than the U-Net, SegNet, and DeepLabv3+ for the SUIM underwater dataset.

The SUIM-Net aims to incorporate visual modeling [52] with computation of spatial saliency [53, 54], which is understanding and predicting the importance of each pixel in the image. The salient regions of the image can be explored to detect and track known objects or discover new objects during underwater exploration. While such a saliency map provides interesting foreground regions, the semantic saliency map embeds additional information about the spatial distribution and interaction among the objects in the scene. This can be later used for spatial-temporal exploration and visual question answering [55], which aims to find the relevant parts of the image in response to a posed question (Fig. 6).

The SUIM-Net model consists of an encoder-decoder structure with skip connections [29, 30] connecting the respective mirrored layers. The model also consists of an optional skip layer called residual skip block or RSB. Each of these blocks has 3 convolutional layers, batch normalization [42] following each of them, followed by ReLU nonlinear activation [43]. Two sets of RSBs are used in the second and third encoder layers sequentially. The number of filters and their respective dimensions is as shown in Fig. 7. The encoder extracts 256 feature maps from the input images, and these maps are used by the decoder network to produce the segmentation output. These feature maps are unique because the model aims to combine the pros of skip connections with those of residual learning [56], which has not been performed in previous models.



**Fig. 7** Complete SUIM-Net RSB model [5]

The decoder network consists of 3 sequential blocks. Each decoder block consists of a convolutional layer, which receives input from the previous layer as well as its respective conjugate encoder layer. This is followed by a batch normalization layer and a deconvolutional layer for spatial upsampling [57]. The final convolutional layer generates the binary pixel labels for each of the classes present in the dataset. These per-channel binary masks can be later combined into an RGB image with each class represented by a single color. Another model trained by the SUIM authors utilizes 12 encoding layers of a pre-trained VGG-16 network [40]. Since this involves pre-trained weights, further training on the SUIM dataset aims to improve its performance. It consists of 4 encoder blocks from the pre-trained VGG followed by 3 decoder layers with the RSBs.

The SUIM-RSB model has slightly poor generalization ability compared to the best models for the same application. This is because it involves a high number of residual and skip connections, leading to a very low number of parameters (computationally cheap). The VGG model is slightly more expensive but can perform much better and hence can be improved on much more. This is because the VGG encoder is already pre-trained on a large corpus of images and hence already has more generalization capability during the encoding phase. Further improvements may be made by adding pre-trained layers to the decoder phase alongside the RSBs to maintain the uniqueness of the model as well as obtain a performance boost.

Table 1 presents some of the advantages and drawbacks of the models in reference to underwater images. In general, it is seen that SUIM-Net performs the best in most categories, with U-Net, VGGNet, and DeepLabv3+ close behind. SegNet

**Table 1** Pros and Cons of segmentation models

Model	Advantages	Disadvantages
SegNet [6]	<ul style="list-style-type: none"> <li>– Without pre-training, SegNet performs better than DeepLabv3+ in underwater images [58].</li> <li>– Since SegNet does not save all encoder feature maps, it uses comparatively less memory during inference; hence, it uses less resources while segmenting underwater images [6, 15]</li> </ul>	<ul style="list-style-type: none"> <li>– Recent architectures that save all encoder feature maps perform better than SegNet in most cases [6, 15].</li> <li>– SegNet lacks in size and complexity to achieve a competitive performance in underwater images [58]</li> </ul>
PSP-Net [9]	<ul style="list-style-type: none"> <li>– The final feature representation generated by the pyramid pooling module includes both local and global context information, thereby obtaining more complex underwater feature maps [9, 25]</li> <li>– Dilated convolutional layers allow for the output of the pyramid pooling module to have more underwater features compared to normal convolution.</li> </ul>	<ul style="list-style-type: none"> <li>– PSP-Net shows multi-pixel blending problems in underwater images [59].</li> <li>– PSP-Net takes relatively longer time to process underwater images compared to other models such as U-Net [9, 25].</li> </ul>
U-Net [2]	<ul style="list-style-type: none"> <li>– The U-Net's skip connections allow the global context and location of the underwater images to be fully learnt.</li> <li>– The skip connections allow the spatial information of certain vague underwater elements to be captured by the decoder [2, 29].</li> <li>– It does not need multiple runs to perform segmentation and can learn with few labelled images as well [32].</li> </ul>	<ul style="list-style-type: none"> <li>– In deeper U-Nets, the learning may slow down to the fact that the model learns to ignore layers with representations of abstract features [33], more so in underwater images.</li> <li>– The U-Net sometimes predicts very fine-grained contours due to poor quality of underwater images, owing to a higher cross-entropy loss, even if not by much [33].</li> </ul>
DNN-VGG [60]	<ul style="list-style-type: none"> <li>– Amplifies the generalization ability when applied to real-time underwater videos.</li> <li>– Achieves a standard real-time frame rate of around 25 frames per second when tested on practical underwater videos [7, 60]</li> </ul>	<ul style="list-style-type: none"> <li>– Higher memory requirements when compared to SegNet [40].</li> <li>– A lack of enhancement on the underwater image can lead to poor training convergence rate [40].</li> </ul>

(continued)

**Table 1** (continued)

DeepLabv3+ [8]	<ul style="list-style-type: none"> <li>– The concept of atrous convolution allows arbitrary control over the resolution of the encoder features extracted in order to achieve optimum precision and runtime for underwater applications [8, 45].</li> <li>– Depthwise separable convolution is applied resulting in a more efficient and stronger encoder–decoder network.</li> <li>– Relatively a small number of learnable parameters [45].</li> </ul>	<ul style="list-style-type: none"> <li>– Bilinear upsampling can result in poor results for underwater images [4, 8].</li> <li>– Faulty segmentation of target edges and a lack of refined edges for underwater images.</li> </ul>
SUIM-Net [5]	<ul style="list-style-type: none"> <li>– It is specifically crafted for underwater images and hence performs well when coupled with the pre-trained VGG encoder.</li> <li>– Like the U-Net, it uses skip connections to enhance learning with spatial context in the decoder network [5, 29].</li> <li>– The residual connections not only address the vanishing gradient problem, but also act as skip connections to learn complex mappings [5, 56], even if the image quality is poor.</li> </ul>	<ul style="list-style-type: none"> <li>– The computational advantage comes at the cost of slightly poor generalization performance, specifically for underwater images.</li> <li>– The SUIM-RSB encoder model does not perform as well as the VGG encoder model with certain class misidentifications occurring due to poorer generalization [5].</li> </ul>

performs competitively but fails to perform consistently across all applications. The PSP-Net performs much better, but at the cost of much higher computation, consisting of around 63M trainable parameters. U-Net’s performance shows that skip connections are very important and performance enhancing in image-to-image translation models. SUIM Net model performs the best overall in region similarity as well as contour accuracy, with its structure utilizing both residual learning alongside skip connections. Although the pure RSB model performs at an average level, the model combined with the pre-trained VGG net should outperform all the other models.

### 3 Comparative Analysis

In this section, the various characteristics of the aforementioned models have been discussed. Table 2 presents this data.

Table 2 presents the different model characteristics including the number of trainable parameters and the number of layers in the model, which are a good measure of how much computation is required by these models. The “Input Size” column in Table 2 represents the dimensions of the input image being passed into each of these models. The higher the dimensions, the more the number of parameters required in each layer to obtain feature mappings. Hence, the “Trainable Parameters” values in Table 2 are directly proportional to the input dimensions of the images. The “Underwater Images’ Suitability” column shows the models that were specifically created for underwater imagery. The asterisk mark signifies that those models were not originally created for underwater images but can be used for underwater applications with decent results.

PSP-Net is clearly the most computationally expensive, with it containing around 63.9M parameters. Even though it contains the least number of layers when compared to the rest, the number of parameters are more than twice that of U-Net and about 4 times that of SegNet. It can also be seen that SUIM-Net on its own consists of a meagre 3.8M parameters, but when paired with VGG, still has only around 12M parameters. This is much better than some of the other models as computation is a major bottleneck in real-time applications. Faster computation can sometimes result in poor results and hence is a trade-off to be maintained as per the application requirement. Underwater applications such as autonomous underwater vehicles (AUV) require faster and more efficient models in order for it to detect and avoid malicious objects in real time. On the other hand, there are some applications such as underwater resource mining where quick computation may not be a constraint, but it may require more accurate segmentation results.

**Table 2** Model Characteristics This table displays the various characteristics of the models [2, 5–9]

Model	Input size	Trainable parameters	Encoder layers	Decoder layers	Underwater images’ suitability
SegNet-ResNet	$320 \times 256$	15.012M	13	13	✓*
PSP-Net	$384 \times 384$	63.9M	3	5	✓*
U-Net	$320 \times 240$	31M	9	10	✓*
DNN-VGG	$640 \times 360$	14.7M	13	13	✓
DeepLabv3+	$320 \times 320$	41.25M	6	3	✓*
SUIM-RSB	$320 \times 240$	3.86M	22	5	✓
SUIM-VGG	$320 \times 256$	12.22M	12	3	✓

\* These models are not specifically created for underwater images.



## 4 Conclusion

Semantic segmentation plays a significant role in underwater applications, especially for unmanned or autonomous underwater vehicles. Poor segmentation results can render the vehicles ineffective in their goal of exploring and gathering data from the depths of the ocean. In this chapter, various different state-of-the-art semantic segmentation techniques were explored, some catered specifically to the underwater scene while others, more traditional models developed for terrestrial images. The architectures, unique features, and novelties of the different models were examined and presented. The encoder–decoder structure, which forms the foundation for all the models explored, produces good results even for underwater images. Further, since the traditional models, such as SegNet, U-Net, PSP-Net, were all developed with terrestrial use cases in mind, the adaptations required to achieve good segmentation results in underwater environments were also explored. The advantages and disadvantages of all the explored models were also presented alongside some model characteristics for each model.

## References

1. J. Jordan, “An overview of semantic image segmentation.” <https://www.jeremyjordan.me/semantic-segmentation/>, Nov 2020.
2. O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” 2015.
3. Z. D. Xiaolong Liu and Y. Yang, “Recent progress in semantic image segmentation,” 2018. <https://arxiv.org/ftp/arxiv/papers/1809/1809.10198.pdf>
4. F. Liu and M. Fang, “Semantic segmentation of underwater images based on improved DeepLab,” *Journal of Marine Science and Engineering*, vol. 8, no. 3, 2020. <https://www.mdpi.com/2077-1312/8/3/188>
5. M. J. Islam, C. Edge, Y. Xiao, P. Luo, M. Mehtaz, C. Morse, S. S. Enan, and J. Sattar, “Semantic segmentation of underwater imagery: Dataset and benchmark,” *CoRR*, vol. abs/2004.01241, 2020. <https://arxiv.org/abs/2004.01241>
6. V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A deep convolutional encoder-decoder architecture for image segmentation,” 2016.
7. Y. Zhou, J. Wang, B. Li, Q. Meng, E. Rocco, and A. Saiani, “Underwater scene segmentation by deep neural network,” Jan 2019. <https://hdl.handle.net/2134/37229>
8. L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” 2018.
9. H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” *CoRR*, vol. abs/1612.01105, 2016. <http://arxiv.org/abs/1612.01105>
10. T. J. Perumanoor, “What is VGG16?—introduction to VGG16,” <https://medium.com/mygreatlearning/what-is-vgg16-introduction-to-vgg16-f2d63849f615>, Sept 2021.
11. C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation functions: Comparison of trends in practice and research for deep learning,” 2018.
12. S. Pathical and G. Serpen, “Comparison of subsampling techniques for random subspace ensembles,” vol. 1, 08 2010, pp. 380–385.

13. D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *Artificial Neural Networks – ICANN 2010*, K. Diamantaras, W. Duch, and L. S. Iliadis, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 92–101.
14. V. Biscione and J. Bowers, "Learning translation invariance in CNNs," 2020.
15. Fezan, "Understanding of semantic segmentation & how SegNet model work to perform semantic segmentation," <https://medium.com/@fezancs/understanding-of-semantic-segmentation-how-segnet-model-work-to-perform-semantic-segmentation-5c426112e499>, Oct 2019.
16. Dumitrescu and C.-A. Boiangiu, "A study of image upsampling and downsampling filters," *Computers*, vol. 8, p. 30, 04 2019.
17. J. Jordan, "Evaluating image segmentation models." <https://www.jeremyjordan.me/evaluating-image-segmentation-models/>, May 2018.
18. Ai, Xinbo, Xie, Yunhao, He, Yanan, and Zhou, Yi, "Improve SegNet with feature pyramid for road scene parsing," *E3S Web Conf.*, vol. 260, p. 03012, 2021. <https://doi.org/10.1051/e3sconf/202126003012>
19. T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, Jul 2017, pp. 936–944. <https://doi.org/10.1109/CVPR.2017.106>
20. A. Jose, D. Merlin, N. Joseph, E. George, and A. Vadukoot, "Performance study of edge detection operators," 07 2014, pp. 7–11.
21. M. Sanatkar, "Analysis and applications of multi-scale CNN feature maps," <https://towardsdatascience.com/analysis-and-applications-of-multi-scale-cnn-feature-maps-a6804bbac8>, Apr 2020.
22. M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jan. 2015.
23. M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," *CoRR*, vol. abs/1604.01685, 2016. <http://arxiv.org/abs/1604.01685>
24. J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CoRR*, vol. abs/1411.4038, 2014. <http://arxiv.org/abs/1411.4038>
25. S.-H. Tsang, "Review: DilatedNet—dilated convolution," <https://towardsdatascience.com/review-dilated-convolution-semantic-segmentation-9d5a5bd768f5>, Nov 2018.
26. F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 05 2016.
27. L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *CoRR*, vol. abs/1606.00915, 2016. <http://arxiv.org/abs/1606.00915>
28. Z. Deng, K. Zhang, B. Su, and X. Pei, "Classification of breast cancer based on improved PSPNet," in *2021 IEEE/ACIS 6th International Conference on Big Data, Cloud Computing, and Data Science (BCD)*, 2021, pp. 86–90.
29. N. Adaloglou, "Intuitive explanation of skip connections in deep learning," <https://theaisummer.com/skip-connections/>, Mar 2020.
30. M. Drozdal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, "The importance of skip connections in biomedical image segmentation," 2016.
31. E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," 2016.
32. H. Lamba, "Understanding semantic segmentation with UNet," <https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47>, Feb 2019.
33. J. Zhang, "UNet explanation," <https://towardsdatascience.com/unet-line-by-line-explanation-9b191c76baf5>, Oct 2019.
34. N. N a, M. H. T P, and S. M H, "Semantic segmentation of underwater images using UNet architecture based deep convolutional encoder decoder model," 03 2021, pp. 28–33.

35. O. Patel, Y. Maravi, and S. Sharma, "A comparative study of histogram equalization based image enhancement techniques for brightness preservation and contrast enhancement," *Signal Image Processing: An International Journal*, vol. 4, 11 2013.
36. W. A. Mustafa and M. M. M. A. Kader, "A review of histogram equalization techniques in image enhancement application," *Journal of Physics: Conference Series*, vol. 1019, p. 012026, Jun 2018. <https://doi.org/10.1088/1742-6596/1019/1/012026>
37. R. de Lutio, S. D'Arconco, J. D. Wegner, and K. Schindler, "Guided super-resolution as pixel-to-pixel transformation," 2019.
38. V. Mottl, A. Kopylov, A. Kostin, A. Yermakov, and J. Kittler, "Elastic transformation of the image pixel grid for similarity based face identification," vol. 3, 02 2002, pp. 549–552 vol.3.
39. Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested U-Net architecture for medical image segmentation," *CoRR*, vol. abs/1807.10165, 2018. <http://arxiv.org/abs/1807.10165>
40. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.
41. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.
42. S. I. and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. <http://arxiv.org/abs/1502.03167>
43. V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning*, ser. ICML'10. Madison, WI, USA: Omnipress, 2010, p. 807–814.
44. M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *CoRR*, vol. abs/1311.2901, 2013. <http://arxiv.org/abs/1311.2901>
45. L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017.
46. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. <http://arxiv.org/abs/1512.03385>
47. F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *CoRR*, vol. abs/1610.02357, 2016. <http://arxiv.org/abs/1610.02357>
48. G. Papandreou, I. Kokkinos, and P.-A. Savalle, "Modeling local and global deformations in deep learning: Epitomic convolution, multiple instance learning, and sliding window detection," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 390–399.
49. "How DeepLabv3 works," <https://developers.arcgis.com/python/guide/how-deeplabv3-works/>.
50. K. Iqbal, M. Odetayo, A. James, R. A. Salam, and A. Z. H. Talib, "Enhancing the low quality images using unsupervised colour correction method," in *2010 IEEE International Conference on Systems, Man and Cybernetics*, 2010, pp. 1703–1709.
51. I. Robotics and U. o. M. Vision Lab, "SUIM dataset," <http://irvlab.cs.umn.edu/resources/suim-dataset>.
52. A. Perini and A. Susi, "Developing tools for agent-oriented visual modeling," 09 2004, pp. 169–182.
53. L. Duan, C. Wu, J. Miao, L. Qing, and Y. Fu, "Visual saliency detection by spatially weighted dissimilarity," 07 2011, pp. 473–480.
54. W. Cui, Q. Zhang, and B. Zuo, "Deep saliency detection via spatial-wise dilated convolutional attention," *Neurocomputing*, vol. 445, pp. 35–49, 2021. <https://www.sciencedirect.com/science/article/pii/S0925231221003179>
55. L. Bazzani, H. Larochelle, and L. Torresani, "Recurrent mixture density network for spatiotemporal visual attention," *CoRR*, vol. abs/1603.08199, 2016. <http://arxiv.org/abs/1603.08199>
56. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

57. M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2528–2535, 2010.
58. P. Drews-Jr, I. d. Souza, I. P. Maurell, E. V. Protas, and S. S. C. Botelho, "Underwater image segmentation in the wild using deep learning," *Journal of the Brazilian Computer Society*, vol. 27, no. 1, p. 12, Oct 2021. <https://doi.org/10.1186/s13173-021-00117-7>
59. F. Liu and M. Fang, "Semantic segmentation of underwater images based on improved DeepLab," *Journal of Marine Science and Engineering*, vol. 8, p. 188, 03 2020.
60. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

# An Interactive Dashboard for Intrusion Detection in Internet of Things



Monika Vishwakarma and Nishtha Kesswani

## 1 Introduction

With each passing year, the Internet evolves and expands, reaching every nook and corner of the world and helping communities to connect. The Internet of Things (IoT) has played an important role in recent years, with various devices sharing data via the Internet. According to a Cisco report, [1], by 2023, the number of Internet-connected appliances will overextend three times that of the global population. Today's time, most people spend seven to 8 h on social sites. However, most of them fall prey to cyberattacks due to a lack of knowledge about dark sites on the Internet. We need a robust system to protect it from being attacked to avoid this.

Many existing intrusion detection systems (IDSs) perform exceptionally well for detecting anomalous events, but some drawbacks have also been reported in related work. Many IDSs have been developed using machine learning and deep learning techniques. In order to achieve a system, the data has to pass through various stages such as data cleaning, preprocessing, and normalizing.

Data is great, but if we cannot communicate the story, it is not helpful. We need ways to organize the data that help us turn it into information. There are two primary ways to represent the data analysis, reports, and dashboards. Both reports and dashboards are used for data visualization. But each of them has some pros and cons. A report is a fixed collection of data. On the other hand, the dashboard can monitor live incoming data. Reports come with many benefits too. They can be created and sent out periodically, often weekly or monthly, as organized and easy to reference information. They are quick to design and easy to use as long as you continuously maintain them. Finally, because reports use static data or data that does

---

M. Vishwakarma (✉) · N. Kesswani  
Central University of Rajasthan, Ajmer, India

not change once it has been recorded, they reflect data that has already been cleaned and sorted. There are some downsides too. Reports need periodic supervision and are not very visually appealing.

Because reports are not automatic or dynamic, they do not show live, evolving data. We need to design a dashboard for a live reflection of incoming data. Dashboards are great for many reasons; they give more access to recorded information that can allow interaction through data by playing with filters. Because they are dynamic, they have long-term value.

We have discussed the existing techniques and issues in Sect. 2.2, which motivated us to build this model to address these issues. The main contributions of this chapter are summarized as follows:

- We have designed an interactive intelligent dashboard that provides data selection and ML algorithm selection using a visualization effect.
- Internally, we designed an intelligent model that performs data cleaning and preprocessing and selects the best features using the voting method.
- The dashboard also provides the visual features of the performance of the selected data on the different ML algorithms.

The rest of the chapter is systematized as follows. In Sect. 2, an analysis is conferred of publicly available IDS datasets and the related work. In Sect. 3, the proposed layered architecture is introduced, explaining the data preprocessing setup, feature selection using votes, and the classification algorithms. Section 4 explains the performance of the dashboard in IDS datasets and the results. Concluding statements and work for future work are explained in Sect. 5.

## 2 Background

In the era of IoT, a considerable number of heterogeneous devices are linked to the network to share information. However, we all know that everything has some negative factors. Similarly, due to the resource constraints of the IoT device, it is unable to provide high security on the network, and the conventional security mechanisms are not applicable to IoT.

### 2.1 Intrusion Datasets

Many intrusion datasets are publicly available, but some are benchmark datasets mainly used to test the IDS model. We have used all the latest standard datasets widely used by researchers. These datasets are summarized below.

**NSL-KDD Dataset** This dataset was proposed to solve the intrinsic issues of the KDD'99 dataset, which are mentioned in [23]. The dataset is primarily divided

into three distinct types: train, test, and test20%. Forty-one different characteristics explain these records.

**UNSW\_NB-15 Dataset** This dataset [19] was prepared by the IXIA PerfectStorm tool at the Australian Center for Cyber Security Lab in 2015. The dataset has been separated into two distinct categories: train and test. It has forty-nine characteristics and a combination of normal and attacked scenarios.

**ToN\_IoT Dataset** The [18] dataset was created at UNSW Canberra's Internet of Things lab. Edge, fog, and cloud are the three stages of the testbed. IoT and network devices make up the edge layer. The fog layer includes virtual machines and gateways. Finally, cloud services such as data analytics and visualization are included in the cloud layer, according to [7].

**MQTT-IoT-IDS2020** Using a simulated MQTT network architecture, the dataset [12] was created. `Uniflow_features`, `biflow_features`, and `packet_features` are the three abstraction layers for features. The routine operation, Sparta SSH brute-force, offensive scan, UDP scan, and MQTT brute-force assault were recorded.

## 2.2 *Related Work*

In this section, we have discussed the existing machine learning (ML)-based IDS and also discussed the problem that motivated us to create this model.

In the era of artificial intelligence, techniques such as ML and deep learning are very much in vogue. In algorithms, several techniques such as K-nearest neighbor, Naive Bayes, random forest, and AdaBoost grab the limelight owing to their performance. Anthi et al. [6] developed an IDS for smart home IoT devices in three layers. They deployed eight IoT equipment on the network and continuously monitored the network traffic to perform a trial run. They made several attacks on the network and kept the log files. After data collection, all attacks are classified into four main attack types: DoS, man in the middle, reconnaissance, and replay attack. The first layer determines IoT devices linked through a scanning network. The second layer classifies packets as vicious or generic. The third layer distributes adversary packets in one of four attacks using nine ML classification schemes and selects the best. However, the data features have not been clearly described. Choudhary et al. [8] presented correlation- and regression-based IDS for smart home. Wenjuan Li et al. [16] presented a disagreement-based semi-supervised tool that works for both labeled and unlabeled data. They used older DARPA (KDD99) data for evaluation.

Eesa et al. [11] presented a feature separation strategy based on the cuttlefish algorithm. To classify the intrusion, they employed a decision tree classification. Pajouh et al. [20] created a model in which they use PCA, LDA, and Naive Bayes to do feature removal and CF-KNN to accomplish classification. They identified four types of attacks: probe, denial of service, and largely targeted U2R and R2L,

and the overall accuracy is not up to the mark. Lee et al. [15] used a software-defined technique to accomplish two-stage AI-based intrusion detection. They used a random forest to classify the flows using the weighted voting approach. Mohamed et al. [17] proposed the highest win-based feature selection algorithm. A Naive Bayes classification algorithm is used to evaluate the highest winning features. Choudhary et al. [9] suggested IDS using a routing protocol that may only be able to detect sinkholes and selective forwarding attacks.

A feature selection system called CorrAUC has been proposed by Shafiq et al. [22] to separate features and preferred appropriate features for ML algorithms by involving the field based on wrapper method. They applied combined TOPSIS and Shannon entropy on a unique simple set to verify the features chosen for malicious traffic classification in IoT networks. Kesswani et al. [14] present IDS-based SmartGuard that can catch malicious possibilities within the network as well as from outside the network. Hussain et al. [13] systematically studies security situations, current protection solutions, and attack vectors for IoT networks to analyze differences between IoT security requirements. They also examine the existing ML and DL solutions to guide various security concerns in IoT networks. Vishwakarma et al. [24] introduced a two-stage IDS model. In the first stage, they used a naive Bayes classification algorithm, and a k-means algorithm in the second stage. Similarly, a neural network and SVM-based hybrid classification model is proposed by Choudhary et al. [10].

We have discussed several existing IDS and feature selection methods designed using ML techniques. Some of them have good performance, but they have been evaluated as models developed on only one dataset, and most of them are challenging to understand due to a lack of visualization results. The primary problem that we have seen is that if we need to build a model, we first have to select the best features of the dataset. Each dataset has slightly different properties that make the model biased. Second, check the performance of the dataset to evaluate the selected feature. Our main goal is classification, which takes much effort to reach. Therefore, we have proposed a model to address these issues.

## 3 Proposed Work

### 3.1 Layered Architecture

The layered architecture of the interactive IDS dashboard is divided into three layers, as shown in Fig. 1. All types of intrusion data are stored in the first layer in CSV format. The second layer contains all the programming and scripting. This dashboard has been made user-interactive in collaboration with Python and HTML. The top layer focuses on application services by creating an interactive user interface, visualization, and model demonstration.



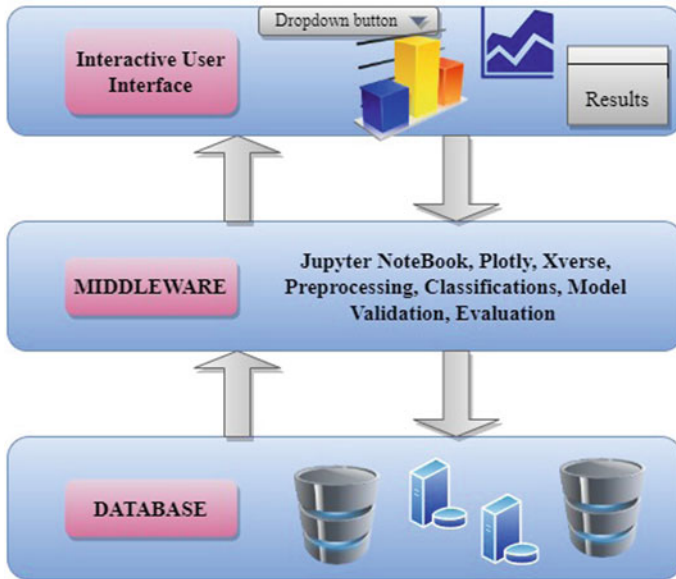


Fig. 1 Layered interactive IDS dashboard architecture

### 3.2 Preprocessing of Datasets

Preprocessing is a step that enables the data to become unrestricted for further data analysis. We generally focus on collecting the intrusion data in raw form. In networking data, some features such as IP addresses are not essential to the dataset when we need to analyze it because it makes the model biased. Likewise, some features have null values or irrelevant data. Therefore, we need to preprocess the data to make it informative.

We tried to build a model that preprocesses any networking data to make the dashboard more interactive. Our model removes all the features containing IP addresses, such as source IP and destination IP. Moreover, it replaces null nominal type features with the mode value of that feature and numeric features with the mean value of that feature.

### 3.3 Feature Selection Using Voting Technique

Features contain information about the data in which some features contribute significantly to the model's high performance, and irrelevant features may reduce the model's accuracy.

xverse [5], short for X uniVerse, is a Python module for feature engineering in machine learning. Presently, the xverse package operates only binary targets. The concept here is to apply a variety of techniques to select features. When an algorithm picks a feature, it assigns a vote for it. In the end, we calculate the total votes for each feature and then choose the best ones based on votes. This way, we select the best features with minimum effort in the variable selection procedure. In xverse, six algorithms involve, which give the best features. These algorithms are information value using weight of evidence, random forest, extra trees classifier, recursive feature elimination (REF), chi-square best variables, and Lasso regression.

The weight of evidence (WOE) [4] describes the predictive ability of an independent variable to the conditional variable. As it developed from credit scoring, it is commonly defined as differentiating good and bad customers. We determine anomalous events and regular events on the network in our approach. The formula of WOE calculation is described in Eq. (1).

$$WOE = \ln \left[ \frac{\text{Distribution of Anomalous Events}}{\text{Distribution of Normal Events}} \right]. \quad (1)$$

- **Anomalous event distribution:** The percentage of anomalous events in a given group
- **Normal event distribution:** The percentage of normal events in a given group

The information value (IV) method is a good way to choose the features that a predictive model needs. It supports ranking the factors in order of significance. Equation (2) is used to compute the IV. After that, selecting features based on the IV means that if the IV is more significant than the suspected predictive power of 0.5 and if the IV is between 0.3 and 0.5, it indicates reliable predictive power.

$$IV = \sum (\% \text{ of anomalous events} - \% \text{ of normal events}) \times WOE. \quad (2)$$

Random forest [3] and extra trees [2] classifiers from scikit-learn [21] package have a built-in property called feature\_importance. The importance of the features is calculated based on the Gini importance (or mean decrease impurity).

RFE is a feature selection approach that wraps around filter-based feature selection. Starting with all features in the dataset, the RFE searches for a subset of features and successfully extracts features until the expected number remains.

The best feature of chi-square is to select the most favorable characteristics based on the response. This problem has two parts: the observed count C and the expected total E. Chi-square calculates the difference between the expected and observed counts E and C. Equation (3) describes the formula for chi-square.

$$\chi^2 = \sum \frac{(C_i - E_i)^2}{E_i}. \quad (3)$$

The Lasso (least absolute shrinkage and selection operator) regression removes some of the features by reducing the coefficient of the less essential attribute to zero.

As a result, all the quality features receive votes from the algorithms that can help reduce the computation complexity to train only the selected features rather than all features.

### **3.4 Classification Algorithms**

The classification algorithm is a supervised learning approach that uses training data to determine the category of new observations. A model learns from a dataset and then classifies incoming observations into distinct classes or groupings in classification. Anomaly detection can be as simple as Yes or No and 0 or 1. All standard classification methods for good results have been incorporated into our dashboard.

**Decision Tree (DT)** DT is a learning method supervised classification and regression. Its goal is to create a model that can predict the value of a target class by deducing decision rules directly from data attributes. This process starts from the root node containing the entire dataset and selects the best feature using information gain and Gini index. After that, the dataset is split based on that feature and shifted to the next node. This process is repeated till it reaches the leaf node.

**Random Forest (RF)** As the name implies, a random forest is made up of multiple individual decision trees that work together. The random forest forecasts a class for each unique tree, and the class with the highest votes becomes the model's prediction results.

**Ada-Boost** It is an ensemble boosting classifier that is iterative. Because the weights are reassigned for each instance, this is known as adaptive boosting. It combines multiple classifiers to improve the classifier's accuracy. The AdaBoost classifier combines many under-performing classifiers to create a robust classifier with high precision. The fundamental goal of Adaboost is to establish the classifier's importance and train the data instance in each iteration to ensure accurate predictions of specific observations. Any machine learning method that can handle the weights on the training set can be used as a primary classifier.

**Gradient Boosting** This is one of the most powerful ML methods. It is used to lessen the model's tendency to misperception because it is a boosting method. Unlike the Ada-boosting approach, the gradient boosting algorithm does not use the base estimator. The gradient boost algorithm's basic estimator, the decision tree, is fixed.

**Extra Trees** This classifier is an ensemble method that combines the results of many de-correlated decision trees and organizes them according to their classification findings. In concept, it is comparable to an RF classifier and counters from it in creating DTs in the forest. Another difference is the way cut ends are chosen to separate nodes. The best partition is determined by random forest, while extra trees determine the best partition at random.

**K-Nearest Neighbor** The KNN algorithm is a supervised learning technique. It is also a flexible approach for resampling datasets and allocating missing values. As the name implies, it uses the K closest data points to forecast the class for a new data point. It compares the new observations to the existing data points and assigns them to the class that is the most similar to the existing classes.

We have included all these classification algorithms in our proposed dashboard that help identify which algorithm can give good results in which type of data.

## 4 Experimental Setup and Results

### 4.1 *Experimental Setup*

The experiments were performed on five different IDS datasets explained in Sect. 2.1. We have used the Python Jupyter notebook framework for model development of the complete code written in Python language. For the back end, we have used open access Python libraries such as Pandas, NumPy, and Sklearn. Moreover, we have used Plotly to create an interactive dashboard for the front end.

### 4.2 *Results*

In our interactive dashboard, the first dropdown list facilitates selecting any dataset to see the best features of datasets. After the selection, data is preprocessed and passed to the features selection step, where several algorithms execute, and each algorithm votes for the particular feature, as shown in Fig. 2, in which the bar graph indicates the data features with their corresponding votes, which suggests the importance of those features in the dataset. Moreover, Fig. 3 provides the interactive chosen facility of selecting any ML algorithm to perform the classification. After selecting the algorithm, data comes from the features selection area where we have chosen a minimum of three vote features, and that data is divided into training and testing in the 80% and 20% ratio, respectively. The second graph shows the validation score in comparison to the training score. The classification results are shown in the third graph, i.e., confusion matrix, and the computed results of the



**Table 1** Binary-Class classification

Classification Models	Measurement	Datasets			
		NSL-KDD	UNSW_NB-15	ToN-IoT	MQTT-IoT-IDS2020
Extra tree	Accuracy	99.44%	98.07%	99.98%	99.94%
	Precision	99.44%	98.08%	99.98%	99.94%
	Recall	99.44%	98.07%	99.98%	99.94%
	F1-score	99.44%	98.07%	99.98%	99.94%
Random forest	Accuracy	99.56%	98.53%	99.98%	99.94%
	Precision	99.56%	98.53%	99.98%	99.94%
	Recall	99.56%	98.53%	99.98%	99.94%
	F1-score	99.56%	98.53%	99.98%	99.94%
AdaBoost	Accuracy	97.13%	95.89%	99.94%	99.93%
	Precision	97.13%	95.89%	99.94%	99.93%
	Recall	97.13%	95.89%	99.94%	99.93%
	F1-score	97.13%	95.89%	99.94%	99.93%
DecisionTree	Accuracy	99.40%	98.49%	99.98%	99.93%
	Precision	99.40%	98.49%	99.98%	99.93%
	Recall	99.40%	98.49%	99.98%	99.93%
	F1-score	99.40%	98.49%	99.98%	99.93%
Gradient boosting	Accuracy	98.82%	96.69%	99.92%	99.94%
	Precision	98.82%	96.69%	99.92%	99.94%
	Recall	98.82%	96.69%	99.92%	99.94%
	F1-score	98.82%	96.69%	99.92%	99.94%
K-Nearest neighbor	Accuracy	99.14%	95.68%	99.85%	99.94%
	Precision	99.14%	95.71%	99.85%	99.94%
	Recall	99.14%	95.68%	99.85%	99.94%
	F1-score	99.14%	95.69%	99.85%	99.94%
Naive Bayes	Accuracy	51.43%	84.55%	64.94%	96.60%
	Precision	51.43%	84.76%	61.30%	96.75%
	Recall	51.43%	84.55%	64.94%	96.60%
	F1-score	51.43%	84.05%	51.20%	96.52%
Multi-layer Perceptron	Accuracy	92.67%	84.83%	37.88%	99.83%
	Precision	92.67%	86.67%	65.87%	99.83%
	Recall	92.67%	84.83%	37.88%	99.83%
	F1-score	92.66%	85.10%	25.16%	99.83%

## 5 Conclusion

Most IDSs are developed using machine learning techniques to efficiently process the most comprehensive data. Since the data is in unprocessed form, the final model is developed through several steps that the data has to pass through, such as cleaning up, irrelevant features, etc., which takes a lot of effort and time. In contrast, the main focus is on building powerful IDS to address this problem. One such interactive

dashboard has been developed. There are popular IDS datasets available that can be used to get the best features of that dataset. We have used various machine learning techniques such as votes for the best features, which will be more relevant to achieving the target class. Furthermore, we have an interactive feature from the most popular machine learning techniques, which examines the performance of selected features, using which the model's performance can be viewed.

We will create a dashboard that can preprocess any data and get the best features in the future. Apart from this, now we will also use deep learning techniques.

## References

1. Cisco Annual Internet Report - Cisco Annual Internet Report (2018–2023) White Paper - Cisco. <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>, (Accessed on 01/30/2022)
2. sklearn.ensemble.extratreesclassifier – scikit-learn 1.0.2 documentation. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html>, (Accessed on 01/21/2022)
3. sklearn.ensemble.randomforestclassifier – scikit-learn 1.0.2 documentation. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
4. Weight of evidence (WOE) and information value (IV) explained. <https://www.listendata.com/2015/03/weight-of-evidence-woe-and-information.html>, (Accessed on 01/18/2022)
5. Xverse · PyPI. <https://pypi.org/project/xverse/>, (Accessed on 01/12/2022)
6. Anthi, E., Williams, L., Słowińska, M., Theodorakopoulos, G., Burnap, P.: A supervised intrusion detection system for smart home IoT devices. *IEEE Internet of Things Journal* 6(5), 9042–9053 (2019)
7. Booiij, T.M., Chiscop, I., Meeuwissen, E., Moustafa, N., den Hartog, F.T.: ToN\_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion datasets. *IEEE Internet of Things Journal* (2021)
8. Choudhary, S., Dey, A., Kesswani, N.: CRIDS: Correlation and regression-based network intrusion detection system for IoT. *SN Computer Science* 2(3), 1–7 (2021)
9. Choudhary, S., Kesswani, N.: Cluster-based intrusion detection method for Internet of Things. In: 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA). pp. 1–8. IEEE (2019)
10. Choudhary, S., Kesswani, N., et al.: A hybrid classification approach for intrusion detection in IoT network. *Journal of Scientific and Industrial Research (JSIR)* 80(09), 809–816 (2021)
11. Eesa, A.S., Orman, Z., Brifcani, A.M.A.: A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems. *Expert Systems with Applications* 42(5), 2670–2679 (2015)
12. Hindy, H., Tachtatzis, C., Atkinson, R., Bayne, E., Bellekens, X.: MQTT-IoT-IDS2020: MQTT Internet of Things intrusion detection dataset. *IEEE Dataport* (2020)
13. Hussain, F., Hussain, R., Hassan, S.A., Hossain, E.: Machine learning in IoT security: Current solutions and future challenges. *IEEE Communications Surveys & Tutorials* 22(3), 1686–1721 (2020)
14. Kesswani, N., Agarwal, B.: Smartguard: an IoT-based intrusion detection system for smart homes. *International Journal of Intelligent Information and Database Systems* 13(1), 61–71 (2020)
15. Li, J., Zhao, Z., Li, R., Zhang, H.: AI-based two-stage intrusion detection for software defined IoT networks. *IEEE Internet of Things Journal* 6(2), 2093–2102 (2018)

16. Li, W., Meng, W., Au, M.H.: Enhancing collaborative intrusion detection via disagreement-based semi-supervised learning in IoT environments. *Journal of Network and Computer Applications* p. 102631 (2020)
17. Mohammad, R.M.A., Alsmadi, M.K.: Intrusion detection using highest wins feature selection algorithm. *Neural Computing and Applications* 33(16), 9805–9816 (2021)
18. Moustafa, N.: New generations of internet of things datasets for cybersecurity applications based machine learning: Ton\_iiot datasets. In: *Proceedings of the eResearch Australasia Conference, Brisbane, Australia*. pp. 21–25 (2019)
19. Moustafa, N., Slay, J.: UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: *2015 Military Communications and Information Systems Conference (MilCIS)*. pp. 1–6. IEEE (2015)
20. Pajouh, H.H., Javidan, R., Khayami, R., Ali, D., Choo, K.K.R.: A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks. *IEEE Transactions on Emerging Topics in Computing* (2016)
21. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12, 2825–2830 (2011)
22. Shafiq, M., Tian, Z., Bashir, A.K., Du, X., Guizani, M.: CorrAUC: a malicious BoT-IoT traffic detection method in IoT network using machine learning techniques. *IEEE Internet of Things Journal* (2020)
23. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the KDD CUP 99 data set. In: *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*. pp. 1–6. IEEE (2009)
24. Vishwakarma, M., Kesswani, N.: A two-stage intrusion detection system (TIDS) for internet of things. In: *Advances in Deep Learning, Artificial Intelligence and Robotics*, pp. 89–97. Springer (2022)



# An Analogous Review of the Challenges and Endeavor in Suspense Story Generation Technique



V. Kowsalya  and C. Divya 

## 1 Introduction

Suspense is a crucial element of fictional universes because it attracts readers and keeps them engaged in the narrative. In the same way, flipping one's sense of suspense to use the models to write more suspenseful tales might be beneficial. Based on the assumption that most tales immerse their viewers just for the sake of entertainment, [5] serious storytelling incorporates people as part of a narrative study that focuses on gaining knowledge and engaging in educationally purposeful activities. This technology develops a storytelling framework [6] even while building dramatic tale sequences around this by giving a piece of fundamental subject information. It does so by interjecting a storyline, suspenseful crisis, and effective execution, which is based on a proven idea of suspense and an artificial intelligence technique known as decision making. According to an experimental analysis of a suspenseful climax [1], the suspense produced might have been impacted only by the listener's fear of being hurt by the characters' imagined outcome [2]. The exact value of taking the difference of others and the input intended ultimate anxiety of the generated narrative yielded a fitness function [5]. Non-diegetic elements, or those that aren't part of the story's world, have always been gathering steam in serious storytelling contexts.

The method of modeling the reader's anticipation about the serious outcome is commonly used by automatic storytelling power generators that explicitly address suspense. The purpose of most suspense-based narrative generators is to generate state sequences that build to occurrences that readers can perceive as harmful or challenging for the major protagonists to resolve. Until now, practically all stories

---

V. Kowsalya (✉) · C. Divya

Centre for Information Technology and Engineering, Manonmaniam Sundaranar University, Tirunelveli, Tamil Nadu, India

were always handwritten. A system that produces tales automatically and otherwise structurally is still in its infancy, owing to the difficulties that making entertaining stories implies. The emotional effect of a scene element is recognized to have a critical role in suspense impression: a robber's razor and mask may signal a more horrific killing than gunshots; a malformed treat may urge viewers to assume a bigger risk. The production of realistic representations of personalities, their behaviors, and their intents is one of the challenges of a computational narrative generation.

This study looked at the steps involved in composing suspense stories, as well as the drawbacks, algorithms, and datasets used in the surveys. It will assist future researchers in producing the best results and in generating new ideas for making the tale more interesting.

## 2 Review of Previous Work

This chapter looks into Suspense story creation with computers. (Sect. 2.1), Videos and Image-Based Storytelling (Sect. 2.2), and Challenges and Attempts in Story-spinning (Sect. 2.3).

### 2.1 *Computerized and Suspense Story Generation*

A feeling of intense tension or uncertainty about what may happen is called suspense. The system is designed to give different emotional components to a reader, and the suspense is reckoned with a regression model, to give the best outcome using fitness function in an evolutionary algorithm. The prophecy of suspense is a facet such as an outcome, decorative element, or threat's appearance [1, 2]. Systems are labelled into interactive and non-interactive story generation systems [7]. Their goal was to discuss briefly the historically important works including the review method used for "goodness" of the system, algorithm, or technique [5]. Introducing the term serious storytelling – a prospective media genre – it defines serious storytelling with an afar amusement.

The IRIS story creation system is just a narrative planner that helps users to come up with exciting stories [6]. It will be used to create both collaborative text-based online games and intriguing non-interactive textual tales. The effectiveness of transformer-based probabilistic models, including Google's Transmitter or XLNet as well as OpenAI's GPT2 (trained and tuned), is applied to manual contextual writings. The substantial link between the reading ratings and word use in the narrative is revealed by examining the inter-metric connection with all metric grading. When sampling is raised, hyperparameter permutations demonstrate that the best-trained and perfectly tuned approaches produced data that respond satisfactorily to the prompts with just a slight rise in the frequency of unusual and challenging words

[3]. Here on 1024 Byte-Pair Encoder, using GPT-2-medium method is fine-tuned (BPE). In its execution, it demonstrates that human performance scores are the least deviated by GPT-2 language models.

A story generation system [1] was tested for its compatibility with described suspense and electromyography responses when the generated outcome was added to narrative passages. At various stages of the investigation, the volunteers were separated.

- **Outcome extraction (OE):** They randomly gathered the movies given to the  $N = 20$  participants to choose the best movies from initial collecting data. To identify tension situations and their accompanying results, a minimum of two participants per film is required, with a range of nine to ten movies per participant.
- **Suspense evaluation (SA):** The second stage contains the collection of the reported suspense for outcome of  $N = 39$  participants, 18 women (53.85%) and 21 men (46.15%). This takes place in a single room, and the words are shuffled randomly for each participant so that the effect of the sequence is minimized.
- **Affective evaluation (AE):** The feeling is measured using the Affective Norms for English Words (ANEW) model, which divides it into three aspects: valence, excitation, and domination. A paper-and-pencil test was designed for this stage. Approximately ten outcomes were presented to the subjects and they were expected to rate them. Participants read the instructions before rating the words.

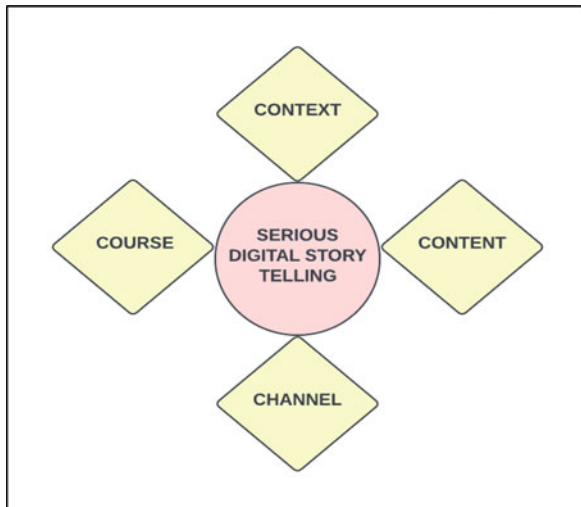
From the standpoint of conversation analysis [16], the role of narrative in the framework is formed in doctoral supervision (CA), and the dataset corpus consists of roughly 25 h of video-recorded exchanges between doctorate candidates and their supervisors at an Australian institution. Furthermore, the students' replies reveal their approach to storytelling as one of obtaining information rather than ensuring expressive attitude identification. For a story generation system for suspense [27], the proposal intends to create a system that generates a narrative that is tailored to elicit tension from its audience. For the suspenseful creation technique, they employ a planning algorithm throughout order to generate tale structure that could also impact the audience's narrative grasp at a given time in reading (Fig. 1).

**Context:** The scenario, location, venue, application context, and any specific situational modality where the storytelling is occurring are all considered to be inside the term "context" (for instance, television, commercials, games, residences, gatherings, celebrations, academics, urban settings, etc.).

**Content:** The narrative's real substance, or the humanly perceptual parts of the story, is represented using a variety of media with genre-specific features, like cinematic phrases, everyday language, poetry, or widespread social media. The information represents narrative elements within the context of serious narration.

**Course:** According to Bordwell's theory of storytelling, a course is made up of a storyline (expressed or non-diegetic actions), viewer interpretations, and assumed events that show how the contents develop in a cause-and-effect sequence within the internet context.

**Fig. 1** Serious digital storytelling follows the 4C paradigm



**Channel:** A channel could be considered the technique in use to distribute content, but it also contains storytelling components like viewer story sharing, viewer narrative interaction, and traditional narrative tools as a platform.

Analogy-based story generation (ASG) [11] The Riu system uses analogical retrieval and projection to explain its methodology. Using the story analogues through mapping (SAM) technique, the authors give an empirical survey of Riu's capacity to recall and construct short, non-interactive tales in their study. By passing data from a specific retrieved story, SAM performs analogical prediction to assume the effects of a certain viewer behavior in the storytelling world. The SAM method is used by the creative projection element to develop tales by analogically passing data from the source language to the target sector. The authors introduce and assess a unique computational analogy-based technique to tale production. Its technical approach consists of the following:

- For stories with a CUD and a HUD, dual representation formalism may be employed.
- A method for enhancing representations of stories, specifically plots, using FD.
- Analogies between stories are found using the SME algorithm by combining structure mapping theory and a mathematical model.

Pordelan [12]. The goal of this study is to see how digital storytelling affects the career counselling process using the life design paradigm. The usage of digital storytelling, as opposed to face-to-face storytelling, resulted in improved student self-efficacy in professional decision-making. This finding is a semi-technical design with a controlled group that consists of a pre-test, post-test, and follow-up. To reach this goal, 45 students in Tehran were picked at random to take an active role in the labor market entry program. Participants were divided into three groups: the normal control (15 pupils), the virtual narrative (15 pupils), and the facial expression

narration (15 pupils). An endeavor has been made to replace the oral personal stories of individuals that are performed as a component of the planning model protocol with digital storytelling. This paper has four limitations:

- Not possible to generalize the result.
- Specification of the training course's uniformity is required.
- This research is based on the life design paradigm, and its applicability to other paradigms requires more research.
- Other data gathering methods, such as observation and interview, were not employed in this study.

Wang et al. [24] examine various metrics for both the developmental narration issue and assess the created narrations using 42 human participants. They suggest building separate human surrogate representations from either the person assessment experiment and then fusing them into an assemblage to accommodate diverse human viewpoints. They show that this method works in terms of improving narrations, as judged by 31 individual participants.

## 2.2 *Motion Pictures and Imagery-Based Storytelling*

Deep learning-based short story generation [13] This study aimed to develop an encoder-decoder framework structure for the automatic generation of short story captions (Scup) from a standardized caption dataset and a manually collected database of stories. They describe how to make a tale out of an input picture in three steps. To generate tales in a range of genres, the suggested approach combines natural language processing (NLP) and an encoder-decoder architecture. They have created a vast database of romance and thriller literature. Moreover, there are various connections between the stories and the images. Deep learning's and AI's ultimate objective is to serve people in a variety of ways that do not necessitate direct supervision. Many academics have worked on utilizing the capabilities of deep learning to build a wiser version of AI to accomplish this target. AI-based algorithms have been created to compete with humans in the arts and popular music. Despite various attempts, systems that can write like humans still haven't been developed because computers lack the necessary innate sense of writing. Calliope [4], a novel visualization tool narrative production system, uses an automated procedure to build visual tales from either a given spreadsheet. Calliope absorbed a new logic-oriented Monte Carlo tree search algorithm that scouts an input spreadsheet to crecscively generate story pieces in a logical order.

“Story to journey to smart map” (SJSM) [8] This design is intended for visitors who are blind or partly sighted (BPS). By telling the history of museum collections, it resembles universal access. However, the main advantage of this technology is that it handles information like a voyage or even travel as a physical smart map, negating the need for new software for button-based smart maps and being highly cost-effective. It has long been known that telling stories may increase and

foster emotional involvement. But BPS visitors have unique difficulties while trying to follow a narrative at a scenic spot. The SJSM strategy has a lot of potential for increasing emotional connection with museum exhibits and ensuring universal health coverage. Yingjie song [15]. This research offers a new narrating genre that uses mixed reality technology to assist children and parents in learning how to construct crafts and then utilize them as impacts on the human tools to design, develop, tell, and share stories. In addition to sending the motion information to the client, the hands' location is translated into the holographic reference system. Crafts instruction and narrative are handled by the HoloLens user. To execute gesture interactions, it converts the incoming hand information to the show's hand marking. Just after the client uses the craft as instructed by the training animations, HoloLens detects its 6DOF pose (elevation and rotations) and maps it with a virtual entity in the surroundings to create tactile interaction. Research findings show that their system is user-friendly and may pique users' interest in crafts and storytelling, as well as efficiently increase parent-child contact.

Microsoft HoloLens is the basis for their system. They introduce an LM control system for hand gesture detection and hand tracking to solve the constraints of its gesture interaction. According to this survey, their system has a high level of availability and is well liked by users.

Video storytelling [9] selects clips to represent the underlying storyline. A video tale dataset is used to assess the approach. Their strategy beats state-of-the-art starting points in the form of numerical evaluations and user studies. The Res Bonn approach for situational multimodal insertion and the narrator models dynamically optimized for clips results based on the tale were presented to address the issues provided by the video story and its duration. The NLP-based solution is being investigated to improve the narrative and generate smoother transitions between phrases.

A good type of visualization for generative storytelling [14] may be found in comic books. This paper outlines a structure for developing characters that interact with each other to generate short comic storylines. The technique is used to generate artificial characters with severe personality traits that are based on well-known comic book characters. This article offers an interactive generative framework for developing short comic tales based on the interactions of characters [17]. The session and recording analysis was aimed to see which particular alterations the participants make to their tale manuscripts, to inform the process of developing a computational framework based on cognitive representations of the narrative production process. They found that there are a variety of frequent alterations that humans apply to a newly formed version after doing the trials, annotating the films, and analyzing results and that this knowledge may be used to design storytelling systems. Video-based interactive storytelling [10] offers a framework of artificial intelligence-based agents that execute the same functions as cinematic experts in this study. Future advancements of the approaches suggested in this research, they hope, will make a significant contribution to the search for new and much more realistic forms of participatory cinematography.

By investigating one lengthy tale sequence from the viewpoints of ethnomethodology and conversational analysis (EMCA) [20], this study analyzes how socializing took place at the micro-level of ordinary parent-child interactions. The episode is organized in chronological order based on around 12 h of sound-visual recordings of parent-child contact.

- How the child’s stated experience is interactively recreated and found thru the child’s reminiscences and the father’s inquisitiveness.
- The youngster concentrates on his emotional responses to what happened, whereas the father orients to accountability, normative conduct, and moral order.

Rühlemann, C., [18] Their study makes use of the XML approach, a “combinatorial methodology” in which multi-modal CA translation is converted into XML structure, and they handle gaze variance by:

- The addressee-status hypothesis posits that the storyteller alternately looks at the receivers when their addressee state is evenly spaced and when it is asymmetrical.
- The texturing theory predicts that as the tale progresses from one part to the next, gaze alternating will “texture” the narration.
- The acceleration theory suggests that gaze alternation increases as you get closer to climax and decelerates as you get closer to completion.

They suggest that the hypothesized relationships should be investigated further to see if they can be applied to other speakers and speech contexts.

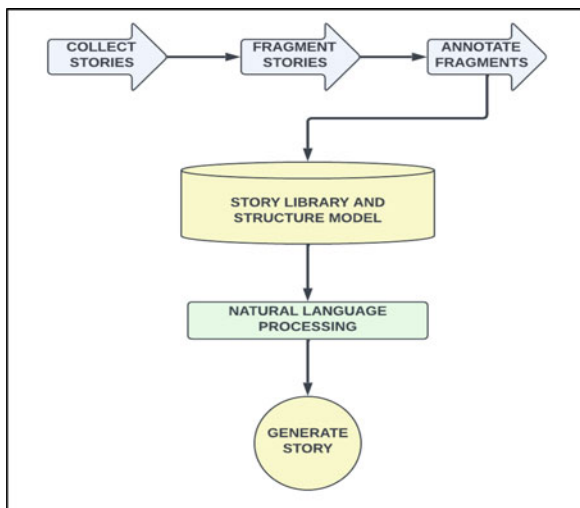
The coronavirus epidemic grabbed media at the same time as major news organizations began to expand existing digital storytelling platforms with more podcasting [19]. This study examines the extent to which dramatic storytelling strategies and other aspects, such as opinions, were included in 40 reality TV episodes linked to the coronavirus epidemic using content analysis. Reporters and hosts frequently appear in the broadcasts, while both personalities and specialists and narrative aspects help to emphasize storytelling above knowledge.

The emergence of podcasting [30] has prompted structural changes that have resulted in programming focused on maintaining audience attention, a trend that coincides with a strong increase in mobile users’ assessment and intervention with digital lengthy material. Three case studies were chosen for this study, each with different industrial and cultural conditions:

- True Murder (a forerunner of the American Drama producers’ series).
- S-Town (serial dynasty, which had a much larger audience than the previous).
- Ear Hustle (a Radiotopia experimental in subject-created material created by two convicts from San Quentin State Prison).

The purpose of this research is to investigate digital stories generated by the primary school teacher candidate (PSTC) [21] for preschool mathematics education, as well as their experiences with the DST production process. The PSTC’s DST for a mathematical learning result had both positive and bad aspects, according to the research. Whenever the positive and negative aspects of the stories are considered together, PSTC produces good texts, particularly in terms of “emotional challenge”

**Fig. 2** A framework for the creation of stories



and “uniqueness.” The story’s “mechanics” (spell, grammar, etc.), vocabulary and emotion, coherence, mathematical, visuals, and sound/music/rhythm aspects, on the other hand, are more dominated by negative traits.

Sarah Copeland [23]. In digital storytelling may become actively used in community outreach proposals to allow realistic ones to be transmitted to certain other stakeholder groups in societies, and also how related tales might provide civic groups momentum to clarify whatever people desire to improve. The main structure of tale generation is shown in (Fig. 2). First, it gathers the tales (Training). Next, break up the narrative lines and label the fragments. Finally, move the narrative libraries or dataset into the structured model and complete the convolutions and tokenization of NLP to generate the complete story.

The CDST technique is presented as a way to deal with difficulties of community of place. There are five phases to this method:

1. Preparing.
2. Storytelling.
3. Story automation.
4. Digital narrative sense-making.
5. Digital story hosting.

A storytelling for adopting CDST uses the loop of trust framework, which includes four trust factors:

- Legitimacy.
- Authenticity.
- Synergy.
- Commons.



The fundamental goal of this approach is to increase the social effect of daily stories, and it serves as a reference for long-form storytelling. The purpose of their study [25] was to see if people's attention was captivated creatively throughout tale encounters. We investigated whether hot spots, or motions that highlight possible bad outcomes, limit viewers' ability to focus more than cold spots, or motions that don't highlight negative outcomes. All through suspenseful hot spots, people who participated overlooked numerous probing and had longer response speed than it was during cool regions in three trials. These studies show that people's visual attention tuned shifts while they watch movies [29]. This writer's implementation seems to be a collection of graphic stories using contrasting aspects that encourage kids to think about their online behavior. For such a humorous method, the review showed that their major study goal can be met. More trials and comparisons with different methodologies are required to fully understand its possibilities and drawbacks.

To follow the narrative and retain the teller-ship, storytellers turn their eyes to the tale receiver rather than their co-teller just at query solution. Researchers introduce a newly built adversarial machine learning method, supervised learning framework, based on GAN and utilize the benchmarking MovieQA and TVQA datasets to propose a unique technique they call adversarial multimodal network (AMN) [22, 28].

This article introduces Monet [26], a storytelling system that uses a collection of preconfigured editing techniques to build fascinating stories from private photos. This approach has two key aspects:

*Photo summarizing* is a technique for representing a photo collection by identifying a set of such "best" photographs.

*Story remixing* is indeed a technique for turning chosen images into attractive music videos.

They invited the uploaders to designate the accuracy and completeness of their photographs depending on the performance of photographing summarizing algorithms. Monet receives a significantly higher grade for the attractiveness of the stunning visuals, indicating that their styling themes and video composition algorithm are both successful.

### ***2.3 Difficulties and Endeavors of Spinning the Story***

Some of the difficulties that an automated narrative developer tackles are handling story data, good judgment skills, trying to imply realistic characterization acts, and originality which seems to be just a few of the skills required (Table 1).

- Neural-based approaches face a tough task in managing tale formation.
- The creation by the interpolated method was created to address these issues; however, it is a superior ending-guided method than storyline-guided ones.

**Table 1** Comparability of survey methods, dataset, outcomes, and limitations

Author/year	Methods	Dataset	Outcomes	Drawbacks
Delatorre [2020]	Evolutionary algorithm Predictive algorithm Generative algorithm	Reviews of films and books	The outcomes show a strong link between the projected emotionality triggered by the outcome and the experienced suspense, as measured by calculated data and EMG responses	The criticality of the various factors contributing to the occurrence and the challenge of evaluating the variation of human reactions to suspense
Shi Danqing [2020]	Rule-based methods Template-based method The novel aggregation methods	Real-world COVID-19 dataset Car sales and startup failures	Calliope was rated higher than data shot in terms of customer satisfaction	There are certain research-related efficiency slowdowns in the development and execution
Wang xi [2021]	Blind and partially sighted (BPS), self-assessment manikin (SAM), QUB smart map controller	The story of Titanic's journey	According to the outcomes of a receiving survey, the strategy enabled BPS participants to have a high emotional connection with museum artifacts	It discusses the difficulties of transforming 2D photos into smart recreations as well as the advantages of employing a multimodal approach in museums
Kyungbok Min [2021]	Template-based method Retrieval-based method NLP methods	Book dataset Conceptual caption dataset	Here as the outcome, an encoder-decoder architecture model is proposed to construct a short story captioning (SSCap)	This resulting text was produced using predefined rules and personally selected characteristics and has certain limitations in the ways presented

- Considering the restrictions introduced towards the generating system to enhance storyline cohesion and clarity.
- Endeavors to resolve the issue of narrative stability by splitting the generating methods into two hierarchical levels: a hypothesis and a narrative.

- The limitation can be overcome by neural-based techniques. When symbolic networks are created in an unstructured way, the uniqueness of the systems suffers.
- The tale creation approach has various flaws, including duplication, off-topic behaviors, and logical contradictions.
- Evolutionary algorithms are usually developed to produce great recommendations for issues that are difficult to solve using standard approaches.
- Predictive algorithms are applied in customer engagement (CRM) activities such as advertising, selling, and customer service.
- Unsupervised learning employs generative architecture to characterize data processes, permitting computers to comprehend the actual world.
- Rule-based algorithms are a prevalent type of methodology in machine learning and artificial intelligence. They both wish to find patterns in data, which may be represented as IF-THEN logic.
- The template-based method works effectively since the design has no important qualities with just a picture even though patterns function directly upon image pixels.
- Techniques that employ self-assessment to examine participants' feelings emerge in traditional emotional stimulating methods.
- Natural language processing allows computers to speak in people's native languages even while automating many language-related tasks.

### **3 Conclusion and Discussion**

A measurable evaluation of the impacts of a suspenseful ending would permit the generated suspense to be balanced out according to the event's hazard. New outcomes will frequently need modifications in the structuring and conclusion of the scenario. Even while such revisions might have seemed small in results with equivalent real consequences, numerous plots will almost obviously require alterations to guarantee stability and accuracy. For many researchers, it would be beneficial for the story generation system if people understand the benefits, difficulties, and approaches. As in the future, more advanced approaches created in the domain of NLP will be required to generate relatively high textual narratives in data stories. It might contain a gripping storyline describing a variety of scenarios including various sorts of anxiousness as well as other troublesome circumstances. Professional testimony may be required to solve coming generations' ethical problems, and experts can communicate their thoughts as digital tales via the Delphi technique.

## References

1. Delatorre, P., León, C., Salguero, A. G., & Tapscott, A. (2020). Predicting the effects of suspenseful outcomes for automatic storytelling. *Knowledge-Based Systems*, 209, 106450.
2. Delatorre, P., Leon, C., & Hidalgo, A. S. (2021). Improving the Fitness Function of an Evolutionary Suspense Generator Through Sentiment Analysis. *IEEE Access*, 9, 39626–39635.
3. Das, A., & Verma, R. M. (2020). Can machines tell stories? A comparative study of deep neural language models and metrics. *IEEE Access*, 8, 181258–181292.
4. Shi, D., Xu, X., Sun, F., Shi, Y., & Cao, N. (2020). Calliope: Automatic visual data story generation from a spreadsheet. *IEEE Transactions on Visualization and Computer Graphics*, 27(2), 453–463.
5. Lugmayr, A., Sutinen, E., Suhonen, J., Sedano, C. I., Hlavacs, H., & Montero, C. S. (2017). Serious storytelling—a first definition and review. *Multimedia tools and applications*, 76(14), 15707–15733.
6. Fendt, M. W., & Young, R. M. (2016). Leveraging intention revision in narrative planning to create suspenseful stories. *IEEE Transactions on Computational Intelligence and AI in Games*, 9(4), 381–392.
7. Ansag, R. A., & Gonzalez, A. J. (2021). State-of-the-Art in Automated Story Generation Systems Research. *Journal of Experimental & Theoretical Artificial Intelligence*, 1–55.
8. Wang, X., Crookes, D., Harding, S. A., & Johnston, D. (2021). Stories, journeys and smart maps: an approach to universal access. *Universal Access in the Information Society*, 1–17.
9. Li, J., Wong, Y., Zhao, Q., & Kankanhalli, M. S. (2019). Video storytelling: Textual summaries for events. *IEEE Transactions on Multimedia*, 22(2), 554–565.
10. de Lima, E. S., Feijó, B., & Furtado, A. L. (2018). Video-based interactive storytelling using real-time video compositing techniques. *Multimedia Tools and Applications*, 77(2), 2333–2357.
11. Zhu, J., & Ontañón, S. (2013). Shall I compare thee to another story?—An empirical study of analogy-based story generation. *IEEE Transactions on Computational Intelligence and AI in Games*, 6(2), 216–227.
12. Pordelan, N., Hosseinian, S., & Baei Lashaki, A. (2021). Digital storytelling: A tool for life design career intervention. *Education and Information Technologies*, 26(3), 3445–3457.
13. Min, K., Dang, M., & Moon, H. (2021). Deep Learning-Based Short Story Generation for an Image Using the Encoder-Decoder Structure. *IEEE Access*, 9, 113550–113557.
14. Nairat, M., Nordahl, M., & Dahlstedt, P. (2020). Generative comics: a character evolution approach for creating fictional comics. *Digital Creativity*, 31(4), 284–301.
15. Song, Y., Yang, C., Gai, W., Bian, Y., & Liu, J. (2020). A new storytelling genre: Combining handicraft elements and storytelling via mixed reality technology. *The Visual Computer*, 36(10), 2079–2090.
16. Ta, B. T., & Filipi, A. (2020). Storytelling as a resource for pursuing understanding and agreement in doctoral research supervision meetings. *Journal of Pragmatics*, 165, 4–17.
17. Leon, C., Gervas, P., & Delatorre, P. (2019). Empirical Insights Into Short Story Draft Construction. *IEEE Access*, 7, 119192–119208.
18. Rühlemann, C., Gee, M., & Ptak, A. (2019). Alternating gaze in multi-party storytelling. *Journal of Pragmatics*, 149, 91–113.
19. Nee, R. C., & Santana, A. D. (2021). Podcasting the pandemic: exploring storytelling formats and shifting journalistic norms in news podcasts related to the Coronavirus. *Journalism Practice*, 1–19.
20. Kim, Y., & Crepaldi, Y. T. (2021). Co-constructed storytelling as a site for socialization in parent-child interaction: A case from a Malay-English bilingual family in Singapore. *Journal of Pragmatics*, 172, 167–180.
21. BÜYÜKKARCI, A., & MÜLDÜR, M. (2022). Digital storytelling for primary school Mathematics Teaching: Product and process evaluation. *Education and Information Technologies*, 1–32.

22. Dressel, D. (2020). Multimodal word searches in collaborative storytelling: On the local mobilization and negotiation of coparticipation. *Journal of Pragmatics*, 170, 37–54.
23. Copeland, S., & De Moor, A. (2018). Community digital storytelling for collective intelligence: Towards a storytelling cycle of trust. *Ai & Society*, 33(1), 101–111.
24. Wang, K., Bui, V., Petraki, E., & Abbass, H. A. (2018). Human-guided evolutionary story narration. *IEEE Access*, 6, 13783–13802.
25. Bezdek, M. A., & Gerrig, R. J. (2017). When narrative transportation narrows attention: Changes in attentional focus during suspenseful film viewing. *Media Psychology*, 20(1), 60–89.
26. Wu, Y., Shen, X., Mei, T., Tian, X., Yu, N., & Rui, Y. (2016). Monet: A system for reliving your memories by theme-based photo storytelling. *IEEE Transactions on Multimedia*, 18(11), 2206–2216.
27. Cheong, Y. G., & Young, R. M. (2014). Suspenser: A story generation system for suspense. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(1), 39–52.
28. Yuan, Z., Sun, S., Duan, L., Li, C., Wu, X., & Xu, C. (2020). Adversarial multimodal network for movie story question answering. *IEEE Transactions on Multimedia*, 23, 1744–1756.
29. Lazarinis, F., Alexandri, K., Panagiotakopoulos, C., & Verykios, V. S. (2020). Sensitizing young children on internet addiction and online safety risks through storytelling in a mobile application. *Education and Information Technologies*, 25(1), 163–174.
30. Dowling, D. O., & Miller, K. J. (2019). Immersive audio storytelling: Podcasting and serial documentary in the digital publishing industry. *Journal of Radio & Audio Media*, 26(1), 167–184.

# Friend Recommendation System Using Transfer Learning in the Autoencoder



Bhargav Rao and Aarti Karande

## 1 Introduction

This paper shows the use of autoencoders used for recommendation systems. Features of the users are mapped to recommend friends based on their mutual connections.

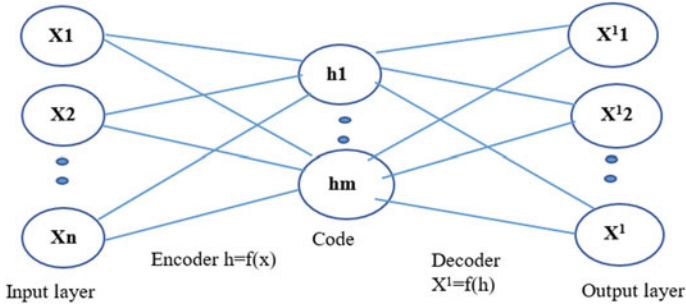
### 1.1 Autoencoder

Most real-world problems like classification, prediction, or analytics use a neural networks family. Convolutional neural networks (CNN) are built with special connections of neural networks. It takes images or values as inputs. CNN generates output as a probability distribution of the predicted classes. Autoencoder (AE) works basically like that of neural networks only. It generates the same output as that of input, enhancing the work of CNN. It aims to transform input to similar output with the least possible amount of noise [1]. Autoencoder means unsupervised learning. The mapping process from original data into an intermediate state is called a replication process. The replication process uses the input size reduced to its smaller representation. Then it again tries to restore the same. It tries to minimize

---

B. Rao  
Information Technology Department, S.P.I.T, Mumbai, India  
e-mail: [bhargav.rao@spit.ac.in](mailto:bhargav.rao@spit.ac.in)

A. Karande (✉)  
MCA Department, S.P.I.T, Mumbai, India  
e-mail: [aartimkarande@spit.ac.in](mailto:aartimkarande@spit.ac.in)



**Fig. 1** General structure of autoencoder

the loss between restoration [2]. It is easy to train input for specific types with appropriate training data. CNN is widely used in the working of recommendation systems.

### Architecture of the Autoencoder

The architecture of an autoencoder is based on code dimension along with the mapping between encoder and decoder. Autoencoder is a feedforward neural network based on the backpropagation concept. The autoencoder component is made of an encoder, code and decoder. It reconstructs the output with less error. As shown in Fig. 1, the encoder compresses the input data into the code value, lowering the dimension of the input. The decoder reconstructs the original input dimensions of the code dimension. Autoencoders are lossy in nature. The output is degraded compared to the original input. The middle layers hold the reduced representation of the input layer. The output is reconstructed from the middle layer. Coder works as per the complexity of the connection. A decoder is based on the complexity of the model distribution. Each node is now treated as a variable.

### Parametrization of Autoencoder

- **Code dimension:** The number of nodes present in the middle layer, or in the last layer of the encoder, is called code dimension. Code dimension classifies autoencoders as under-complete and over-complete. In under-complete, code dimension size is smaller than the input size. In over-complete, the code dimension is larger than or equal to the input.
- **Number of layers:** It specifies the number of node layers present as the hidden layer. It defines deep and shallow autoencoders based on the number of hidden layers. A shallow autoencoder has just one hidden layer extracting only important features. Deep autoencoder has more hidden layers to perform multiple selections of useful features.

- **Loss function:** It is the objective function. It calculates the effectiveness of training neural networks. The loss function is used to find the errors, i.e., the difference between output data and the input data.
- **Optimizer:** These are different algorithms used to find optimal values as output. It uses mathematical functions to calculate the output during the training of a neural network.
- **Regularization:** It is a strategy for improving performance by reducing errors in the calculation. It checks for gradient optimization algorithms for training.

### Working of Autoencoder Network

Autoencoder performs training on the input ( $x$ ) to generate the output ( $r$ ) with intermediate as the code latent representation ( $h$ ).

1. **Encoder:** This takes the input ( $x$ ) for the network. It reduces the dimension of the input into a latent-space representation ( $h$ ). Encoding function  $h = f(x)$  is having functionality as per the requirement of the problem statement. Encoder transforms the input which is high-dimensional into a code with a low-dimensional presentation. It represents crisp and short data. Encoder represents the complexity of the problem.
2. **Decoder:** This is the most important part of the network. It regenerates the original input as output from the intermediate representation. It uses a decoding function  $r = g(h)$ . This transforms the short dimension code into a high-dimensional output. It regenerates the output with minimum errors of transformation.
3. **Autoencoder functionality:** Function  $g(f(x)) = r$  represents  $r$  as an output which is almost equal to the original input  $x$ . Based on the size of the input ( $x$ ), latent ( $h$ ), and output ( $r$ ), there will be three cases that can be possible with this.
  1. If  $h$  is in smaller dimensions than the input  $x$ , it is called under-complete. It is used in learning the most important features. It reduces its dimensionality by simply capturing useful features from the data [4].
  2. If the  $h$  is the same as that of  $x$ . Here the same image is restored without extracting any useful information about the distribution of the data.
  3. If  $h$  is greater in dimension than the input  $x$ , it is called over-complete. It adds the regularization terms. It gives some constraints on the hidden representations. This can learn useful information about the data structure.

### Types of Autoencoder

1. **Basic autoencoder:** This is a simple autoencoder with the simple structure of an encoder and a decoder. For an encoder with  $X$  data input sample with  $m$  features, the output of encoder  $X$  represents the reduced representation of  $X1$ . The decoder is tuned to reconstruct the original dataset by minimizing the difference between  $X$  and  $X1$ . Specifically, the encoder is a function  $f(X) = sf$



$(W.X + b.X)$  that maps an input  $x$  to hidden representation  $Y$ .  $sf$  is a nonlinear activation function, a weight matrix  $W$ , and a bias vector  $b \in R^n$ . The decoder function  $g$  maps hidden representation  $Y$  back to a reconstruction  $X1$ , where  $X0 = g(Y) = sg(X1.Y + b.Y)$ . The decoder's parameters are a bias vector  $b$  and matrix  $W1$ .

2. Vanilla autoencoder: It is the simplest autoencoder. Its structure is the same as that of a simple autoencoder. Code dimension has fewer dimensions compared to the input layer [11]. It reconstructs the input using Adam optimizer. It uses mean squared error as a loss function. It is used in image morphing.

$$\begin{aligned} \text{mixed image} &= \text{decoder}(\text{code1}*(1 - \alpha) + \text{code2}*\alpha), \\ \text{where } 0 \leq \alpha \leq &= 1 \end{aligned} \tag{1}$$

where  $\alpha \in [0,1]$ ,  $\text{code1}$ , and  $\text{code2}$  are the output of the encoder for the first and second images.

3. Convolutional autoencoder (CAE): The CAE is an extension of convolutional neural networks (CNN). CNN is a feedforward neural network. It can be trained with a gradient descent algorithm or stochastic gradient descent algorithm. They can learn to remove noise from the picture. They reconstruct missing parts from the input data. It consists of multiple convolutional nodes. Inputs are two-dimensional feature maps. It includes learning parameters as the elements of filter matrixes [10]. It handles bias (which is a scalar) with input two-dimensional feature maps. It works with filter matrixes with activation functions.
4. Multilayer autoencoder: In this more hidden layers are added in the structure instead of just one as the extension. Any of the hidden layers can be picked as the feature representation. It makes the network symmetrical. It uses the middle-most layer for feature extraction. Depending on the complexity of the problem statements, layers for feature extraction can be added. These layers are also evaluated for building accurate models.
5. Regularized autoencoder(RA): It encourages the model to train with reconstructing the input through the intermediate layers. To reduce the errors, regularization algorithms are used.
6. Sparse autoencoder: Based on the training, it learns the features for another task. It generates unique statistical features from the dataset. Sparse representation of data can be done by adding regularization in the loss function. It involves a sparsity penalty. This penalty varies as per the dataset [17].
7. Denoising autoencoder: This works similarly to the sparse autoencoder. It considers the reconstruction error of the loss function [11]. It adds extra value as noise to the input image. Then the autoencoder learns to remove it while generating output. Encoders represent the data in a robust representational manner and will work on the most important features.

8. Stacked autoencoder: It extracts the features at different levels in the autoencoder. It represents new features extracted from the input. Stacked autoencoder can be trained by a greedy layer-wise feedforward approach. It adds hidden layers for getting more fine-tune features.
9. Variational autoencoder (VA): Autoencoder uses a variational learning approach for latent representation. It handles additional loss components as a stochastic gradient variational Bayes estimator. The probability distribution of the latent vector matches the training data. VAs are much more flexible and customizable. VA lacks the ability to tackle sequential data.
10. Sequential variational autoencoder (SVA): It constructs hidden space following a Gaussian distribution. It uses an LSTM neural network as a recurrent encoder and decoder. SVA is a well-designed generative model. It gives an intermediate nonsequential vector. It represents information captured from the recurrent encoder [12].
11. Extreme learning machine autoencoder (ELM-AE): ELM network handles the encoder and decoder function.  $H = g(W.X + b)$  where the  $g$  is the activation function and the  $W$  is the weight matrix of the input layer which is randomly assigned.  $X$  is the data matrix [10].
12. Stacked sparse denoising autoencoder (SSDA): It is inductive. It transforms learning-based techniques. During training, it learns to denoise from a large amount of data. It handles noisy samples as input accordingly outputs the clean samples [6].
13. Adversarial autoencoder (AA): It defines prior distribution,  $q(z|x)$  function as an encoding distribution. It defines an aggregated posterior distribution of  $q(z)$  on the hidden code vector. For the prior distribution,  $p(x|z)$  functions as the decoding distribution.
14. The linear autoencoder(LA): It has only one hidden layer. The encoder compresses the input data into a lower dimension. The decoder regenerates the original feature space. The latent representation is the only important feature space with a projection matrix [14].

## 2 Friend Recommendation

Different social media platforms encourage users to add images, contents, and tables as per their interest. This data may not be in the standard format. Social networks are the group of users based on their interests. Based on the properties of interest, a set of proprietary and closed social networks are formulated. Sparsity problems indicate insufficient transactional or feedback data for specific similarities. A normal recommender system reduces information overloading. It provides a personalized suggestion for assisting the users in the decision-making.

## 2.1 *General Recommendation System*

Addressing the sparsity problem of hybrid data, it is required to improve the recommendation's quantity and quality. Here it cares about the user's interest and transactional data to combine different social information. For example, before adding a friend, the user's personal data or his/her likings, activities, etc. need to be matched with other people's interests. Privacy of the user's data is important. Social networks need to take care of increasing cybercrimes. Hence, systems need to build the trust power of the user. General recommendation systems can be classified as content-based recommendations, collaborative recommendations, and hybrid recommendations. A content-based system is based on the user's previous work. It shows the user's social interests based on the content of users. Due to privacy issues, it is not easy to collect the user's information for the recommendation [6]. In the collaborative approach, the recommendation is based on items consumed by users as that of the referred user's item. Hybrid approaches combine the content-based and collaborative recommendations approach [16] [21].

## 2.2 *Different Models Used for Recommendation System*

- Topic modeling: It extracts the information from the messages. It works based on the concept of topics. It works in text mining only. This collection of information is unstructured and explains the semantic meaning of the text. The lifestyles are extracted using the topic modeling [23].
- Friend-book: It is a friend recommendation system. It works handling similar lifestyles of the users. The system discovers lifestyles from user-centric sensor data [24].
- Collaborative filtering: It works on the basis of other people's opinions for a recommendation. It learns dot product function from data based on the latent user [20]. This method generates sparse data problems [22]. User feedback is not enough for reflecting the latent interests of users [6].
- Demographic filtering (DF): It classifies users' data according to their demographic information. It takes the information based on the user's cumulative behavior or preferences of access. Then, it sorts and recommends the new services. Based on the search made by a new user, recommendations are for him. Consequently, the cumulative preferences of previous operations are applied to that category [25] [5]. Based on the stored data, new predictions are made.
- Content-based filtering (CBF): It works the same as that of information filtering research. The recommendations are as per associated features mapping. Based on the filtered features, the user has been rated. It learns the profile of the user's interest [25]. Privacy policy needs to be considered while collecting the user's data [6].

- Support-based pruning: This method checks for the frequency of the items present in the network. If an item set is infrequent, then all of its supersets must be infrequent too. It checks to trim the exponential search space based on the support measure.
- Social content features: It checks the probabilities of two users having similar interests. Similarities are calculated using text similarity algorithms. Based on content, social content features are selected [7].

### ***2.3 General Model for Recommendation System***

1. Rating system: General recommendation system works on very large entities. It has large-scale databases for accurate prediction. It takes a random whole network. Then it extracts the subnetwork of any random individual from the visualized graph. This subnetwork can be represented using different terminologies:
  1. Direct user rate: This rate will check common friends or common interests among the connection of the people. Direct connection gives explicit information showing the very common behavior of the user.
  2. Indirect user rate: This rate checks for indirect connection. It holds implicit information with useful hints. This builds the uncommon behavior of the user. This will help to find more potential friends.
2. Classifying activities: Based on the liking, working, and visiting behavior of the user, the network can be classified. The frequency of the execution of any activity defines its behavior as common behavior or uncommon behavior. The natural recommendation process works only using common behavior. Integrity can be enhanced by considering the user's uncommon behavior. Uncommon behavior can be identified as per a similar interest but with less frequency. To identify uncommon behavior, multilayer NN can be used as per the feature extraction.
3. Friend recommendation: The overall activity set of the user is added in the large circle representation. Circles define the different users with the performed activities in a particular timestamp. Activities are grouped as least frequently and most frequently. Based on the set of activities, common users can be recommended to be friends with each other [9]. Transfer learning aims to transfer knowledge from a source to a related target domain. Transfer learning faces the challenges for "what to transfer" and "when to transfer."

### ***2.4 Autoencoder for Recommendation System (Fig. 2)***

Autoencoder models can handle heterogeneous data like rating, audio, visual, video, etc. Autoencoders can understand user behavior, their demands, and item features.

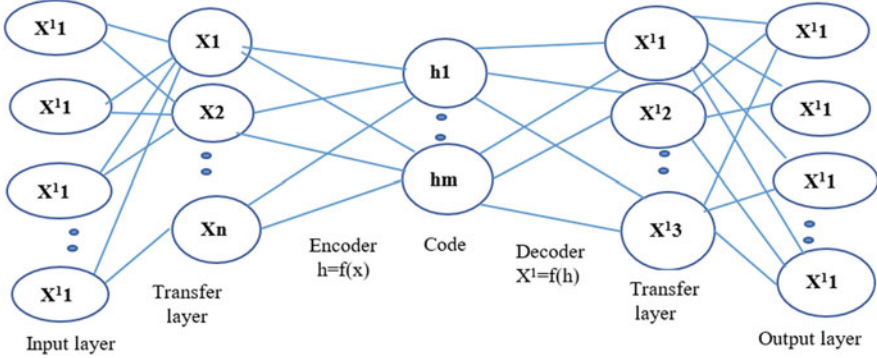


Fig. 2 Social representation using autoencoder

Recommendation models based on autoencoder are more adaptable in multimedia scenarios handling input noises. Sparse autoencoder learning algorithms retrieve features automatically based on unlabeled data. There are data-dependent nodes in a trained model. Sparse autoencoders inhibit the hidden nodes for regularization purposes. Results preserve the most relevant features. It reduces volume dimension [19]. Transfer learning can be used in many areas like data dimensional reduction, noise cleaning, feature extraction, data reconstruction, and so on [26]. As shown in Fig. 2 transfer learning can be used as a node in Autoencoder for transferring the representation of data using the encoding and decoding process.

## 2.5 Autoencoder Parameter for Recommendation System

The following points need to be considered for building a recommendation system using an autoencoder.

- To handle classification problems, the loss function needs to be the mean square error (MSE) [16]. The loss function can penalize activations within a layer.
- Optimizers work very fast on little memory. It is ideal for handling large amounts of data.
- During training, dropout is used for reducing noise. It is then automatically disabled during execution.
- Metrics measure the accuracy. It shows the metrics of statistical and decision support precision. Accuracy is measured by comparing the expected evaluation with the actual user calculations.
- Decision supports accuracy metrics. Metrics are precision and recall. Precision is obtained from the fraction of recommended items relevant to the user.

$$RMSE = \sqrt{\frac{1}{n} \sum_{u,i} (p_{u,i} - r_{u,i})^2}$$

where  $p(u, i)$  is the expected evaluation for the user ( $u$ ) in item ( $i$ ),  $r(u, i)$  is the actual evaluation of the user, and ( $n$ ) is the total number of evaluations in the set of items. The recall is the fraction of relevant items that are also part of the recommended item set [27].

- Accuracy of the recommendation mechanism is inversely proportional to the RMSE value. It regularizes the weights of a network. But it cannot regularize the activities.

## 2.6 Working of Autoencoder Model for Recommendation System

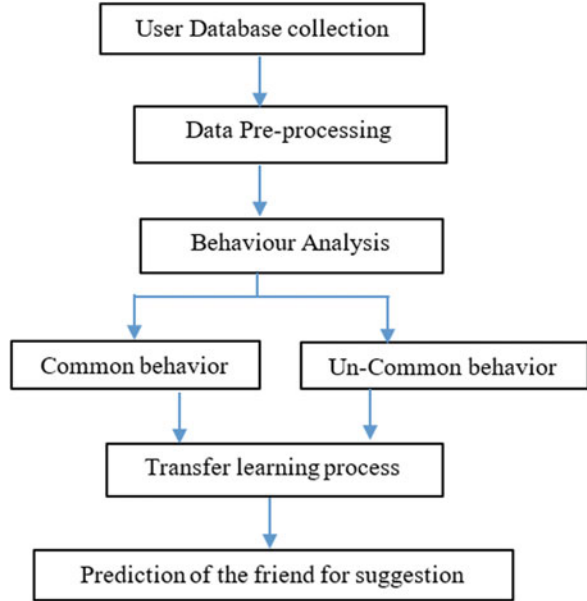
Transfer learning shares features from the source to the destination structure of the source, and the destination may be different but they need to be related to reason [15]. The similarity of the model can be measured using trained models from the source to the target domain. Use a simple transformation model [3, 18]. Model learns from a dataset present in the source. This learning is applied to new datasets but in similar problems [13]. It solves the problem more quickly and effectively. It extracts the particular latent or common features with the same marginal distribution [8].

Common behavior depends on the common activities of the users. It may not be fixed or pre-defined. One or more activities altogether can be a common behavior. Any activity of a user which is not on the list of the common behavior is called uncommon behavior. For different datasets, the uncommon behavior will be different. Similarly, one or more activities altogether can be uncommon behavior. The system requires common and uncommon behavior of the user. This data is collected from other peer users and from the field itself. A recommender system for social media is to predict users' interests and a list of items as per interest [6]. The following steps are used for the recommendation system. The general flow of the system is shown in Fig. 3.

**Step 1: Data acquisition:** The data used in this is derived from a dataset of graph nodes provided by the stanford.edu website which is called the social circles (Facebook), and then these nodes are then created into a dataset of user1, user2, mutual friends among them, and friends or not. This dataset now in csv format goes into data preprocessing.

**Step 2: Data preprocessing:** This stage converts the raw data into the final dataset using data preprocessing techniques. These tasks include data transformation, data cleaning, and data standardization. Filtering techniques can be used to avoid problems of data redundancy. All the duplicate data were identified and removed to avoid ambiguity. Now, we converted the data into a NumPy array. Then we converted the NumPy array into a matrix where every row shows if they (row number and column number) are friends or not. Then this data is converted to

**Fig. 3** Transfer learning-based recommender system



torch tensors so we can finally use this data inside our autoencoder model. These steps help to improve the results.

**Step 3: Autoencoder model:** It applies modeling techniques as per the processed data. It provides behavioral parameters to the learning dataset. The purpose of an autoencoder with the transfer learning concept is to retain a dimensional representation of data. User overlap rate is calculated in the encoder system of the autoencoder. Only important features are selected to reduce the size of the latent representation. Trained parameters from the encoder are utilized for the decoder. Highly prioritized parameters are used here to get the results. An error is measured between input ( $x$ ) and output  $u(x)$ .  $u(x) = \text{decode}(\text{encode}(x))$  needs to be minimized [25]. The decoder will utilize the values of the trained parameters from the latent representation.

**Step 4: Evaluation:** The recommendation system for the user's behavioral analysis performs classification based on the behavioral frequency of the execution. High-frequency common behavior and less frequency are uncommon behavior that is considered for analysis. The autoencoder model reduces the input data of behavior using latent representation ( $L$ ). Then the recommendation system will relate this measurement in the decoder system to predict the friend ( $O$ ). Learning parameters are the elements of filter matrixes [10].

$$L = f \left( \sum_{n=1}^d W_i * x_i + b * 1 \right)$$

Here, the  $b$  is bias, the  $x_i$  ( $i = 1, 2, \dots, d$ ), and  $d$  is feature maps.  $W_i$  ( $i = 1, 2, \dots, d$ ) are  $d$  transferred frequencies based on filter matrixes. The  $\Sigma$  is a sum

operator. The  $f$  is an activation function; the  $L$  is the output feature map for latent representation.

$$O = g(L) = \left( \sum_{n=1}^d Frqi * hi + bj * 1 \right)$$

Here, the  $b_i$  is bias, the  $h_i$  ( $i = 1, 2, \dots, d$ ) are the latent representation of frequency feature maps,  $Frq_i$  ( $i = 1, 2, d$ ) are  $d$  transferred frequencies on based filter matrixes to the decoder, the  $\Sigma$  is sum operator, the  $g$  is an activation function, and the  $O$  is the output feature map. RMSE is calculated for the calculated frequencies to the predicted frequencies. Error is also calculated for transferred feature maps from latent representation to the decoder layer.

Step 5: Result parameter: Accuracy, precision, and error calculation are involved in the result comparison. The recommenders are evaluated using classification accuracy metrics. The accuracy of the recommendation systems needs to be optimized. It helps in enhancing the technical performance and life cycle of the system. It handled the parameters like scalability and reliability. These important features are applied to the next turn of the evaluation. Here the characteristics of the trained model are transferred to the new set of data. A list of common behavior can be converted to uncommon behavior based on the frequencies of usage. And vice versa is also applicable.

### 3 Transfer Learning

Transfer learning is machine learning research in which a model when developed for a certain task is reused with its weights and connection with new other layers of the network. This enables models to be trained on things that don't have a lot of data for them. For example, there is a lot of data available for Twitter that can be used to train the model for LinkedIn (which works on similar terms) which does not have a lot of pre-defined data available for usage.

#### 3.1 Pseudocode for Reference

Following is the pseudo code for creating neural network architecture sae of six layers.

```

la from sae.children()
[:-3], Linear(20,40), Linear(40,80), Linear(80,1000),
Sigmoid activation function
    set criterion as MSELoss
    set optimizer as RMSProp
initializing nb_epoch to 100
for epoch ∈ (1, nb_epoch+1) do

```



```

initializing train_loss to 0
initializing s to 0
for id_user ∈ (0,1000)
input is initialized to Variable (training_set_torch
[id_user]).unsqueeze(0)
target is initialized to input
if torch.sum of target.data which is greater than
zero is greater than zero then
output is set as la(input)
target.required_grad is set to False so it trains only
on input's data
now output is set to output times target not
equal to zero
loss is set to criterion which takes parameters
output and target
mean_corrector is set to 1000/(torch.sum of target.data
which is greater than zero is greater
than zero plus 1e-10 to avoid 0/0)
backward method is called
set train_loss to train_loss + numpy squareroot of
(loss.data * mean_corrector)
s is incremented
optimizer is used by optimizer.step()
print train_loss/s

```

This code shows the method for the neural network working for 100 epochs. During each epoch, 100 user-id data is trained to find minimum loss. Each epoch is working with backpropagation to recheck the model. The mean square root is the error function.

## 4 Outcome of the Model

This model pre-synthesis the raw data to extract important features. The size of the input data is reduced to a small-size latent representation. As shown in Fig. 4 training loss with autoencoder shows graphs start decreasing as the important features trained the new model. This model can be reused to save time. Using Transferring learning, more layers of execution have used an autoencoder. Hence the more accurate result is obtained using transfer learning as shown in Fig. 5. Training loss in the graph shows behaviour of parameters used for transfer learning. Behavior analysis is used to relate common and uncommon behavior. A friend recommendation system can be used to predict the behaviour analysis based on transfer learning. Autoencoder with a number of layers recalculates the result for accuracy.

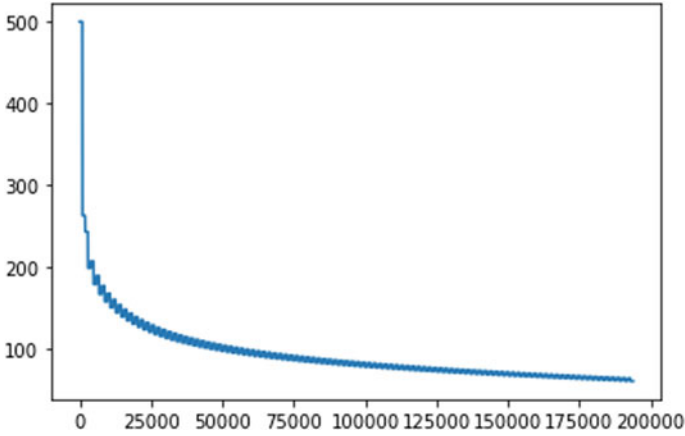


Fig. 4 Training loss with autoencoder

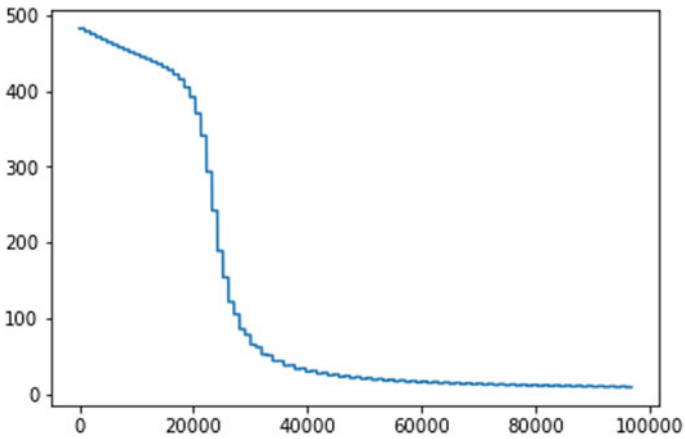


Fig. 5 Training loss with transfer learning on autoencoder

## 5 Conclusion and Future Work

Analyzing the working of the recommendation system focuses on the accuracy of the system. Accuracy is based on the parameter used for prediction. Using the tangible parameter, it will be useful to transfer measuring from the input to the output processing system. The transfer learning system will use the processed value instead of re-evaluating the same again. Autoencoder provides the provision of an encoding and decoding system, to generate the original data separating from noise. The recommendation system takes the user’s behavior handling, from the original data to the predicted data. Using an autoencoder, data preprocessing and parameter evaluation is done. Prediction of a friend is calculated based on the latent

to output representation. This chapter focuses on the different types of autoencoders. It also specifies different methods available for recommendation techniques. This chapter concludes with a method based on transfer learning to predict the friend recommendation system. The transfer learning model is trained and developed for one type of data and then reapplies the learning on another related data. It transfers or maps the knowledge from the source to the target domain. This is used when a small dataset is mapped with a large data project. As per the preliminary training, the reuse of learned parameters from the source was applied to the new dataset. The learning process on the data is done using random weight initialization. It tries to conquer the discrepancy of training samples for some categories by adapting classifiers trained for others.

### Acknowledgments

- (a) user dataset from <https://data.world/ahalps/social-influence-on-shopping>

### References

1. Pierre Baldi (2012) Workshop on Unsupervised and Transfer Learning Autoencoders, Unsupervised Learning, and Deep Architectures JMLR: Workshop Conference Proceedings 27:37–50
2. Qinxue Meng, Daniel Catchpoole, David Skillicorn, Paul J. Kennedy (9 Feb 2018) Relational Autoencoder for Feature Extraction arXiv:1802.03145v1 [cs.LG]
3. <https://www.tensorflow.org/> (accessed on 3rd July 2021).
4. Veronica Kazak (2018) Unsupervised feature extraction with Autoencoder for the representation of Parkinson's disease patients NOVA Information Management Model
5. Great learning Blogs: <https://www.mygreatlearning.com>
6. Yiteng Pan Fazhi He Haiping Yu (2020) Learning social representations with deep autoencoder for recommender system Springer Science+Business Media, LLC, Springer Nature 2020
7. Han Siyao China hsy (2014) Friend Recommendation of Microblog in Classification Framework: Using Multiple Social Behavior, International Conference on Behavioral, Economic, and Socio-Cultural Computing <https://doi.org/10.1109/BESC.2014.7059527>
8. Meidi Sun, Hui Wang, Ping Liu, Shoudao Huang, Peng Fan, (2019) A sparse stacked denoising autoencoder with optimized transfer learning applied to the fault diagnosis of rolling bearings, Measurement, Volume 146, Page 305-314, ISSN 0263-2241, <https://doi.org/10.1016/j.measurement.2019.06.029>.
9. Y. Dong et al., (2012) Link Prediction and Recommendation across Heterogeneous Social Networks, IEEE 12th International Conference on Data Mining, 2012, pp. 181–190, <https://doi.org/10.1109/ICDM.2012.140>.
10. J. Zhai, S. Zhang, J. Chen and Q. He, (2018) Autoencoder and Its Various Variants, 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 415–419, <https://doi.org/10.1109/SMC.2018.00080>.
11. Hieu Mac, Dung Truong, Lam Nguyen, Hoa Nguyen, Hai Anh Tran, and Duc Tran. (2018) Detecting Attacks on Web Applications using Autoencoder. In Proceedings of the Ninth International Symposium on Information and Communication Technology (SoICT 2018). Association for Computing Machinery, New York, NY, USA, 416–421. <https://doi.org/10.1145/3287921.3287946>

12. D. Huang et al., (2019) A Variational Autoencoder Based Generative Model of Urban Human Mobility, 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), pp. 425–430, <https://doi.org/10.1109/MIPR.2019.00086>.
13. Majumdar, (Jan. 2019) Blind Denoising Autoencoder in IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 1, pp. 312–317, <https://doi.org/10.1109/TNNLS.2018.2838679>.
14. Elyor Kodirov, Tao Xiang, Shaogang Gong (2017) Semantic Autoencoder for Zero-Shot Learning; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3174–3183
15. Aarti Karande, Prof. Prachi Dalvi (2020) Emotion Identification Using CNN-Based Transfer Learning Second International Conference on Advanced Computing Technologies and Applications 2020. <https://doi.org/10.1007/978-981-15-3242-9>
16. Diana Ferreira, Sofia Silva, António Abelha and José Machado (2020) Recommendation System Using Autoencoders Appl. Sci., 10, 5510; <https://doi.org/10.3390/app10165510>
17. L. Wen, L. Gao and X. Li, (Jan. 2019) A New Deep Transfer Learning Based on Sparse Auto-Encoder for Fault Diagnosis, in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 1, pp. 136–144, <https://doi.org/10.1109/TSMC.2017.2754287>.
18. Florian Strub, Romaric Gaudel, Jérémie Mary. (2016) Hybrid Recommender System based on Autoencoders the 1st Workshop on Deep Learning for Recommender Systems, Boston, United States. pp.11–16, <https://doi.org/10.1145/2988450.2988456>. final-01336912v2f
19. Lacic, E., Reiter-Haas, M., Kowald, D. et al. (2020) Using autoencoder for session-based job recommendation User Model Inter 30, 617–658 <https://doi.org/10.1007/s11257-020-09269-1>
20. Hidasi, B., Quadrana, M., Karatzoglou, A., Tikk, D. (2016) Parallel recurrent neural network architectures for the feature-rich session-based recommendation In: Proceedings of the 10th ACM Conference on Recommender Systems, pp. 241–248. ACM
21. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S. (2017) Neural collaborative filtering In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182. International World Wide Web Conferences Steering Committee
22. Liang, D., Krishnan, R.G., Hofman, M.D., Jebara, T. (2018) Variational autoencoders for collaborative filtering arXiv preprint. arXiv:1802.05814
23. Fathima Mol, Neetha B S (2015) Friend Recommendation System for Social Networks: A Semantic and Profile-based Approach, International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Published by, [www.ijert.org](http://www.ijert.org) NCICN-2015 Conference Proceedings
24. Z. Wang, C. E. Taylor, Q. Cao, H. Qi, and Z. Wang. (2014) Friend book: A semantic-based friend recommendation system for social networks IEEE Transactions on Mobile Computing, Page(s): 1
25. Iateilang Rynksai L. Chameikho (2014) Recommender Systems: Types of Filtering Techniques, International Journal of Engineering Research & Technology (IJERT) IJERT ISSN: 2278-0181 Vol. 3 Issue 11, Nov-2014
26. Hinton, G.E.; Zemel, R.S. Autoencoders, (2018) Minimum description length and Helmholtz free energy, In Advances in Neural Information Processing Systems; MIT Press: Cambridge, USA, 1994; pp.3–10. 46.
27. Santana, M. (2018) Thesis: Deep Learning para Sistemas De Recomendação(Parte 1)-Introdução. Acknowledgment user dataset from <https://data.world/ahalps/social-influence-on-shopping>

# Analysis on the Efficacy of ANN on Small Imbalanced Datasets



Gauri Naik , Deep Siroya , Manav Nisar , Bhavya Shah ,  
and Himani Deshpande 

## 1 Introduction

There are many methods available now to analyze tabular data. The neural networks are highly efficient and provide good accuracy. In this study, artificial neural network (ANN) is chosen for analysis. As we know artificial neural network models are efficient with big sized datasets, but they face problems when dealing with smaller datasets. The problem faced with some of the smaller datasets is imbalance as well. This study focuses on achieving the solution for the same. The below study with various sections gives us a chance to grasp the results of 13 standard datasets which are imbalanced in nature. The next section, Literature Survey, throws light on various ways in which artificial neural network models can solve the problem of imbalance in datasets. The following section, Methodology, talks about artificial neural network, TabNet, and two other customized neural networks. Section 4, Results, provides insights about the differences in the discussed models and is followed by the conclusion.

## 2 Literature Review

In [1], we come to know that the main problems faced when dealing with imbalanced data are as follows: (1) Normally, the cost of missing a minority class is significantly higher than the cost of missing a majority class. (2) The majority of learning systems are unprepared to cope successfully with substantial disparities

---

G. Naik · D. Siroya · M. Nisar · B. Shah (✉) · H. Deshpande  
Department of Information Technology, Thadomal Shahani Engineering College, Mumbai, India  
e-mail: [himani.deshpande@thadomal.org](mailto:himani.deshpande@thadomal.org)

in the number of instances in each class. (3) When information is unbalanced, the classification algorithm fails to fulfill expectations (most classifiers agree that class appropriation is equal). There are two types of imbalance datasets, binary imbalance and multi-class imbalance. There are two ways of solving the imbalance problem, one by resampling (under sampling and oversampling) and another by adjustment of cost on the original dataset. Artificial neural network is an algorithm used for classification of such data as it does not assume any class distribution among the class. Clustering is used to reduce the imbalance ratio among the classes.

In [2], feedforward artificial neural network is chosen. A backpropagation method is incorporated to train the data of ANN. The actual projected output of training data from this trained network is then trained using particle swarm optimization (PSO). G-mean is calculated to evaluate the performance of the classifier. The experimental effects that the ANN version put forward can gain higher overall performance to the ANN classifier without the usage of any of the sampling techniques. In the future, the proposed ANN version and appropriate preprocessing approach has been used to compare their performance to other standard statistics sets.

Another paper [3] focuses on application of artificial neural network in areas of research. The study found that neural network models, such as feedback and feedback propagation, performed better than artificial neural networks when applied to human problems.

A similar paper [4] focusing on small datasets in the medical sector with complexities in data of the human body (Eq. 1) explains how ANN also takes nonlinearity into account in small datasets.

After researching about the problems caused by imbalance datasets [5, 6], we started to research about artificial neural networks [7] and test various algorithms to analyze their respective efficacies. A paper [8] focuses on the fundamental nature of the imbalanced learning problem and state-of-the-art solutions applied to this problems. Another paper [9] throws light upon the best feature selection methods to be used, depending upon the size of the majority and minority classes of the dataset.

### 3 Methodology

Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subtype of artificial intelligence and are at the heart of deep learning algorithms. Their name and construction are propelled by a human's mind, imitating the technique that natural neurons converse with each other.

Artificial neural networks (ANNs) have node layers, which have input-output layers, with a minimum of one hidden layer. Every node connects with another. They have a weight and threshold. When the output of any node exceeds the threshold value, that node is triggered, sending data to the next layer of that network. Else, nothing is sent to the next layer.

The ANN approaches which we have used in this paper are as follows:

1. **MLP:** A multilayer perceptron (MLP) is a deep ANN made up of multiple perceptions. They contain an input layer to receive the signal and an output layer that predicts the information and a variable number of hidden layers between the input-output layers. The hidden layers are the real prediction mechanisms of the model. An MLP with just one hidden layer can be used for any consistent function. MLPs are usually used for supervised learning problems: they train on input-output pairs and try to correlate them. Training involves modifying parameters or weights or biases to reduce error. Backpropagation changes these weights and biases with respect to the error. The real error can be calculated by many ways, for example, by using root mean square error (RMSE).
2. **TabNet:** It uses some kind of soft function collection to find the elements that are useful for the current model. This is accomplished through a consecutive multistep decision technique. The input data is handled hierarchically at many levels. The composition reads: “The idea of sequential top-down attention is inspired by its applications in the processing of visual and language data, such as for the visual answer of questions (Hudson & Manning, 2018) [10] or in reinforcement learning (Mott et al., 2019) [11] when looking for a small relevant subset of information in high-dimensional input.”
3. **Three-Layer Sequential ANN [3L-ANN]:** This is a sequential model developed by us which includes three dense layers – an input, a middle, and an output layer. The input layer contains a batch size of 32 and uses the ReLU activation function. The middle layer contains a batch size of 64 and uses the ReLU activation function. The output layer gives a binary output by using the sigmoid activation function. The compile method uses the “binary cross-entropy” loss function and the “adam” optimizer. It judges the accuracy of the model based on the accuracy of the results.
4. **Five-Layer Dropout Networked Sequential ANN [5L-DOL-ANN]:** This is another sequential model developed by us which includes five dense layers. The first input layer contains a batch size of 32, using the ReLU activation function. This is followed by a dropout layer with a threshold of 0.2. The second layer has a batch size of 64 and also uses the ReLU activation function. This is followed by another dropout layer with a threshold of 0.2. The third layer has a batch size of 32 with a ReLU activation function. It is again filled by a dropout function with a threshold of 0.2. The fourth layer has a batch size of 16 with the ReLU activation function. The output layer gives a binary output by using the sigmoid activation function.

The compile method uses the “binary cross-entropy” loss function and the “adam” optimizer. It judges the accuracy of the model based on the accuracy of the results, similar to the previous network.

## 4 Dataset Analysis

Here we will be looking at imbalanced smaller non-image datasets of all different categories. The number of instances in the dataset ranges from 148 to 5472. These data are related to the fields of medical research, life sciences, etc. As can be seen from these tables, the experimental datasets are diverse in the number of attributes and imbalance ratio. All the datasets are suitable for binary classification. All datasets are taken from the Keel repository, to maintain standardization.

$$\text{Imbalance ratio} : \frac{\text{No. of positive outcomes}}{\text{No of negative outcomes}} \quad (1)$$

Maximum correlation for each dataset is also mentioned to help the reader perform further analysis (Table 1).

All the datasets, except three, have an imbalance ratio  $\geq 2$ .

## 5 Results

We have used four models for experimental purposes. These models, as defined in the methodology section in this paper, are as follows:

- Multilayer Perceptron (MLP)
- TabNet
- Three-Layer Sequential ANN [3L-ANN]

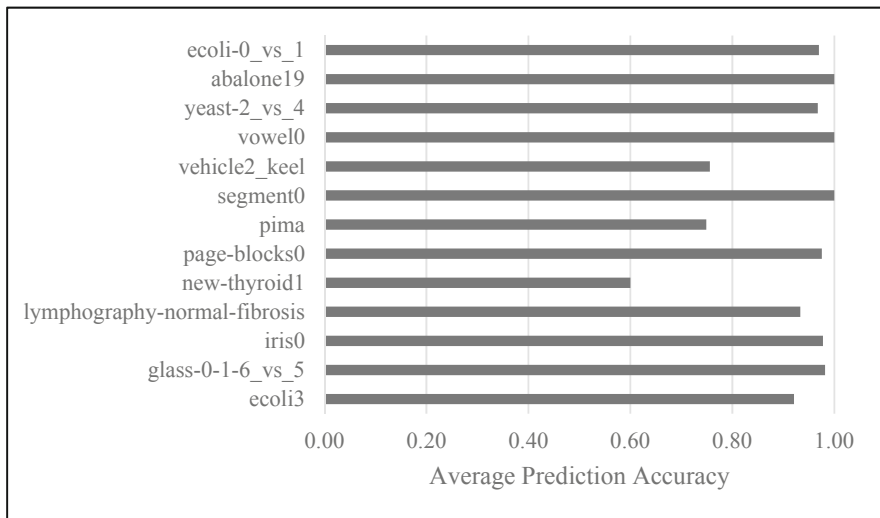
**Table 1** Dataset analysis

Dataset	No. of attributes	No. of instances	Imbalance ratio
Iris0 [16]	4	150	2
Abalone [12]	8	4174	<b>129.44</b>
Ecoli-0_vs_1 [13]	7	220	1.86
Ecoli3 [14]	7	336	8.6
Glass-0-1-6_vs_5 [15]	9	184	19.44
Lymphography-normal-fibrosis [17]	18	148	23.67
New-thyroid1 [18]	5	215	5.14
Page-blocks0 [19]	10	5472	8.77
Pima [20]	8	768	1.87
Segment0 [21]	19	2308	6.01
Vehicle2_keel [22]	18	846	2.88
Vowel0 [23]	13	988	9.98
Yeast-2_vs_4 [24]	8	514	9.08



**Table 2** Results based on average prediction accuracy

Dataset	MLP	TabNet	3L-ANN	5L-DOL-ANN
ecoli3	0.96	0.92	0.96	0.91
glass-0-1-6_vs_5	1.00	0.98	0.94	0.96
iris0	1.00	0.98	0.67	0.65
lymphography-normal-fibrosis	0.93	0.93	0.96	0.96
new-thyroid1	1.00	0.60	0.82	0.85
page-blocks0	0.98	0.98	0.90	0.90
Pima	0.75	0.75	0.64	0.65
segment0	1.00	1.00	0.85	0.86
vehicle2_keel	0.96	0.76	0.73	0.73
vowel0	1.00	1.00	0.91	0.92
yeast-2_vs_4	0.92	0.97	0.91	0.90
abalone19	1.00	1.00	0.01	0.91
ecoli-0_vs_1	0.98	0.97	0.36	0.36

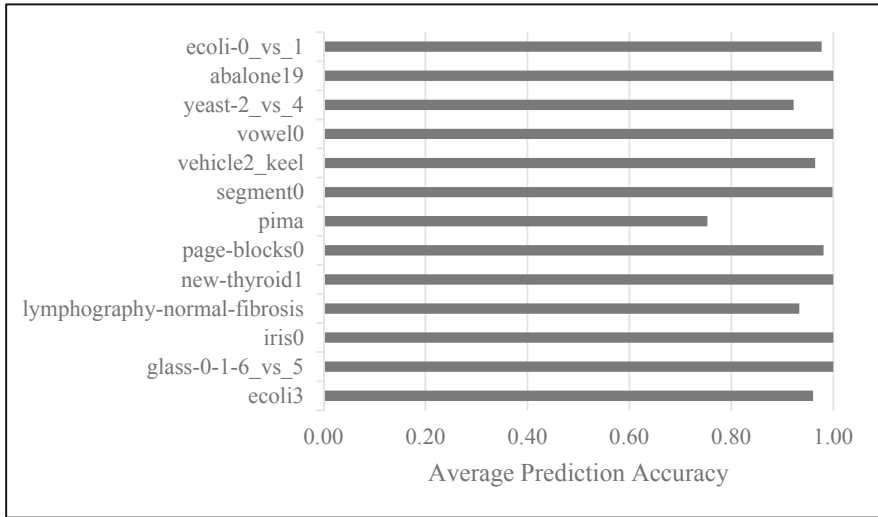


**Fig. 1** Average prediction accuracy for MLP

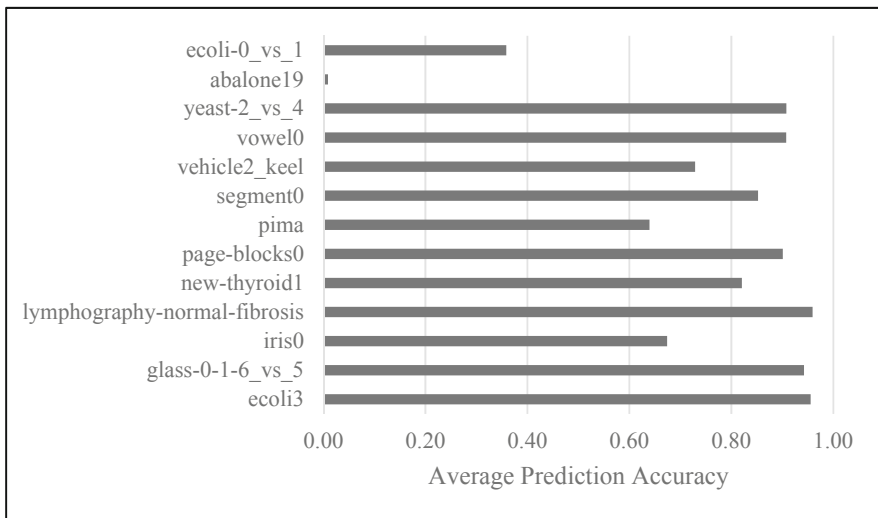
- Five-Layer Dropout Networked Sequential ANN [5L-DOL-ANN]

We have found out average prediction accuracy of each model over all the datasets mentioned in the dataset analysis section of this paper. Table 2 shows the results, and Figs. 1, 2, 3, and 4 show the information in a graphical format. Figure 5 shows the average accuracy of a particular over all the datasets, to brighten the knowledge about the performance of the models over all the datasets.

The following are some graphs for the analysis of performance of the models.



**Fig. 2** Average prediction accuracy for TabNet



**Fig. 3** Average prediction accuracy for 3L-ANN

Figure 1 shows that the classification results using MLP classifier on different datasets, and it has been found that MLP has come with best results on abalone19, vowel0, and segment0 datasets.

Figure 2 displays the classification results for TabNet, which has produced highly accurate results for most of the datasets.

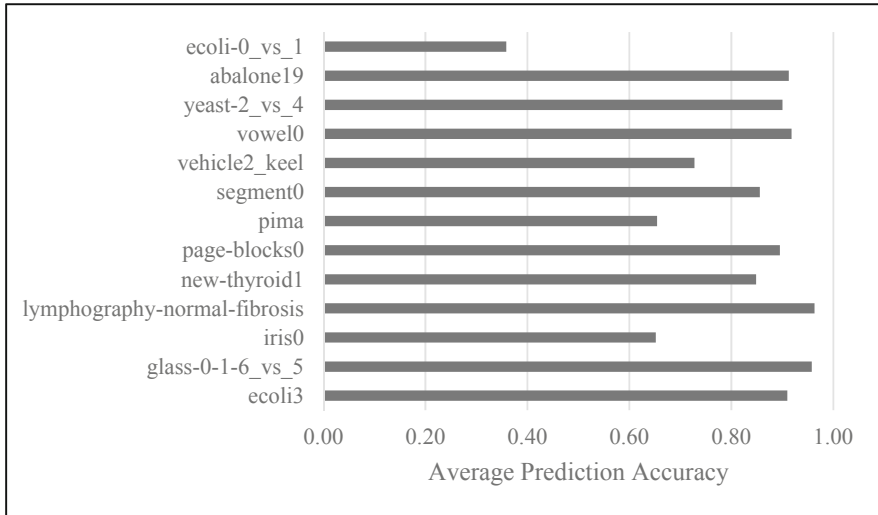


Fig. 4 Average prediction accuracy for 5L-DOL-ANN

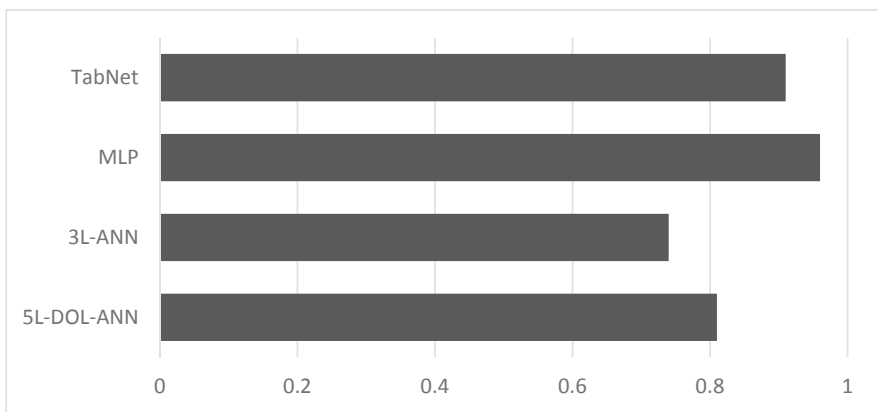


Fig. 5 Average of prediction accuracy for all models

Figure 3 shows the results achieved by Three-Layer Sequential ANN. Evidently, the algorithm has not shown a satisfactory result with ecoli-0\_vs\_1 and abalone19 datasets.

Figure 4 shows the classification results achieved by Five-Layer Dropout Networked Sequential ANN. The results shown by this model are better than the previous model. However, the result for ecoli-0\_vs\_1 dataset is unsatisfactory.

Figure 5 summarizes the results for all the algorithm applied on all the imbalanced datasets. The best results are achieved by MLP.

## 6 Conclusion

The main purpose of this study is to find which ANN model will give highest accuracy for small imbalance datasets. In this research we investigated four different models that are MLP, TabNet, and two customized models, one having three layers and other having five layers in ANN, and compared the performance of 13 different datasets. In most cases, the MLP model gives the best accuracy. It must be noted that in the datasets where the max correlation is low or negative, the prediction accuracy for our customized models is poor. TabNet and our customized models perform similarly in most cases, with one or two exceptions. TabNet is able to outperform all the other modes in one case, just like our customized models. In 11 out of 13 cases, MLP has given the best results.

MLP has given us an accuracy of greater than 90% in all, but one cases. In the Pima Indian diabetes dataset, where normal machine learning models provide an accuracy of about 70%, MLP was still able to outperform them by 5%.

## References

1. A. Sonak et al., "A new approach for handling imbalanced dataset using ANN and genetic algorithm," 2016 International Conference on Communication and Signal Processing (ICCSP), 2016, pp. 1987–1990, doi: <https://doi.org/10.1109/ICCSP.2016.7754521>.
2. A. Adam et al., "A Modified Artificial Neural Network Learning Algorithm for Imbalanced Data Set Problem," 2010 2nd International Conference on Computational Intelligence, Communication Systems and Networks, 2010, pp. 44–48, doi: <https://doi.org/10.1109/CICSyN.2010.9>.
3. O. Abiodun et al., (2018). "State-of-the-art in artificial neural network applications: A survey." *Heliyon*, 4(11), e00938.
4. A. Pasini, "Artificial neural networks for small dataset analysis." *Journal of thoracic disease* vol. 7,5 (2015): 953–60. doi:<https://doi.org/10.3978/j.issn.2072-1439.2015.04.61>
5. V. KrishnaVeni, T.Sobha Rani, "On the Classification of Imbalanced Datasets," *IJCST Vol 2. SP 1*, 2011
6. Vaishali Ganganwar, "An overview of classification algorithms for imbalanced," *International Journal of Emerging Technology and Advanced Engineering* ISSN 2250-2459. Volume 2. Issue 4, 2012.
7. Vidushi Sharma, Sachin Rai, Anurag Dev, "A Comprehensive Study of Artificial Neural Networks," *International Journal of Advanced Research in Computer Science and Software Engineering*. Volume 2. Issue 10 , 2012.
8. Haibo He, Eduardo A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions On Knowledge And Data Engineering*. VOL. 21, 2009.
9. Inaz Jamali, Mohammad Bazmara and Shahram Jafari, "Feature Selection in Imbalance data sets," *IJCSI International Journal of Computer Science Issues*. Vol 9. Issue 3.No 2., 2012.
10. D. Hudson, & C. Manning. (2018). "Compositional Attention Networks for Machine Reasoning."
11. A. Mott et al., (2019). *Towards Interpretable Reinforcement Learning Using Attention Augmented Agents*.
12. KEEL on "Abalone19 Data Set": <https://sci2s.ugr.es/keel/dataset.php?cod=115>
13. KEEL on "Ecoli-0\_vs\_1 Data Set": <https://sci2s.ugr.es/keel/dataset.php?cod=140>

14. KEEL on “Ecoli3 Data Set”: <https://sci2s.ugr.es/keel/dataset.php?cod=139>
15. KEEL on “Glass-0-1-6\_vs\_5 Data Set”: <https://sci2s.ugr.es/keel/dataset.php?cod=120>
16. KEEL on “Iris0 Data Set”: <https://sci2s.ugr.es/keel/dataset.php?cod=50>
17. KEEL on “Lymphography-normal-fibrosis Data Set”: <https://sci2s.ugr.es/keel/dataset.php?cod=1337>
18. KEEL on “New-thyroid1 Data Set”: <https://sci2s.ugr.es/keel/dataset.php?cod=145>
19. KEEL on “Page-blocks0 Data Set”: <https://sci2s.ugr.es/keel/dataset.php?cod=147>
20. KEEL on “Pima Data Set”: <https://sci2s.ugr.es/keel/dataset.php?cod=155>
21. KEEL on “Segment0 Data Set”: <https://sci2s.ugr.es/keel/dataset.php?cod=148>
22. KEEL on “Vehicle2\_keel Data Set”: <https://sci2s.ugr.es/keel/dataset.php?cod=151>
23. KEEL on “Vowel0 Data Set”: <https://sci2s.ugr.es/keel/dataset.php?cod=127>
24. KEEL on “Yeast-2\_vs\_4 Data Set”: <https://sci2s.ugr.es/keel/dataset.php?cod=131>

# Lightweight and Homomorphic Security Protocols for IoT



Ishaan Singh, Aakarshee Jain, Ikjot Singh Dhody, and B R Chandavarkar

## 1 Introduction

IoT devices [1] are computing devices that can exchange data wirelessly on the Internet of Things (IoT). It involves expanding the Internet network far beyond the commonly used devices such as mobile phones, laptops, and tablets to any hardware device requiring a physical connection, thereby facilitating communication between connected devices. These can later be monitored and managed remotely. Though the IoT devices differ a little in terms of how they work, they have numerous similarities. The device can be a sensor, an intelligent camera [2], or even something like an RFID tag. Generally, all IoT devices take note of the things going on in the outside, physical world. Some IoT devices are readily reachable over the Internet. However, many IoT devices operate privately.

Various challenges can tamper with the efficient and effective deployment of an IoT system [3] and its nearby devices. These challenges include scalability, availability, power capacity, interoperability, security, and confidentiality [4]. As a result, it is not surprising that proper management—to organize, control, and invigilate the same—is mandatory and necessary. Many communication and wireless protocols are involved in the data exchange. Each of these protocols has limitations in power consumption, bandwidth, and range. Hence, all protocols chosen for IoT environments need to be lightweight, so that on-board memory consumption and compute requirements are minimized. Apart from all this, security and privacy

---

I. Singh (✉) · A. Jain · I. S. Dhody · B. R. Chandavarkar

Department of Computer Science and Engineering, National Institute of Technology Karnataka, Surathkal, India

are significant issues to be addressed. It is not difficult to gain access or control over vehicles by manipulating a cellular network [5]. Not only this, protecting IoT devices against fraud can become more challenging due to the variety of threats and the difficulty of providing security to resource-limited devices.

Homomorphic encryption [6] is one such way of ensuring some privacy. It enables computations such as multiplication and addition, to be performed on encrypted data, without needing to decrypt the data-in-transit. Hence, privacy and confidentiality are not compromised at any cost [7]. An evident example of the same is the case of elections [8], where privacy is crucial and necessary. People lose faith in the government due to the malpractices in the elections. However, the reasonable question is, “What if it was viable to fetch the election data in its encrypted format and make the necessary computations without needing to decrypt it?”. This is precisely what a homomorphic algorithm does. We can now take all the votes in the encrypted ciphertext format and later aggregate them into a single encrypted tally using homomorphic encryption. Further computation can be made on these results, after which decryption of the final result will fetch the candidate-wise tallies of the same quickly. The great thing is all of this is done without the need of compromising privacy, security, and confidentiality. Thus, the chapter deals with the significant issues that need to be studied thoroughly. A similar study is made in this paper [9].

In the context of IoT devices, their networks, and its limitations, this chapter discusses the requirement for homomorphic and lightweight algorithms. Apart from this, it also discusses various security protocols that address this requirement. Further, comparing nine historically proposed lightweight homomorphic encryption schemes on various parameters, the chapter classifies them as partially homomorphic encryption (PHE) or fully homomorphic encryption (FHE). Finally, the chapter aims to unify the representations used by different authors to enable their comparison on common parameters.

The remainder of the chapter is organized as follows. Section 2 presents the standard networking architecture for IoT devices and discusses the limitations of IoT devices in the context of being part of this architecture. Section 3 presents the need to study lightweight and homomorphic cryptosystems to counter the limitations presented in Sect. 2. Section 4 presents various security protocols that find implementation in real-life IoT systems to the reader’s better understanding of the architecture at play. Section 5 then follows up on Sect. 4 by studying various related works on securing the IoT networks using lightweight and homomorphic cryptography. Section 6 presents the author’s recommendations on what factors should truly influence the industry’s choice for a lightweight and homomorphic security protocol for large-scale IoT networks and concludes our findings. Following this, in Sect. 7, the chapter presents scope for future work in this domain of research.

## 2 IoT Devices, Their Network, and Limitations

Internet of Things (IoT) has emerged as the meta of modern technology in the twenty-first century. A multitude of daily appliances, sensors, vehicles, smart cards, and other electronics are tagged as “IoT”. In fact, any piece of invention, which can collect some data and transmit it over a network, can be a part of an IoT network. IoT devices are not like traditional computers as they are limited in terms of compute capacity and on-board storage. They require all their computation to be lightweight for this reason.

The idea behind IoT networks is to introduce in-transit devices that reduce the load on the network’s servers, helping with the compute traffic and energy requirements by abstracting the computation and traffic control into these nodes, instead of performing the same on the servers. While the data generated at the IoT devices seldom undergoes computation on-board, the real ownership of the data-in-transit is with these devices. The network gateways perform all computation on the data and its routing, and this exposes a lot of vulnerabilities from the point of view of network security.

Because of this ideology, any piece of data generated by the IoT device, by means of invention or measurement, spends most of its time in the network, rather than on the device. All computation is outsourced, and naturally, security is a major concern in such networks.

IoT networks have certain bottlenecks as a consequence of the design, which can be explained as follows:

1. **IoT devices are “lazy”**—The only task the IoT devices perform is data encryption and decryption, apart from temporary storage. The reason for this behaviour is that IoT devices are normally sensors or measurement devices that have to provide live feed of some information to another device connected to the IoT network [10]. Those recipient devices use this information for further applications. Hence, the encrypted data is instantly relayed to the server nodes in the IoT Network for any further computation and transmission.

Moreover, this “laziness” is not by choice or design. These devices are incapable of heavier compute workloads, and the choice of the encryption algorithm should also be optimal. It should be **lightweight**, and the key requirements of the algorithm should be such that it is light on the **memory restrictions** of the device.

2. **Internet Gateways are a public channel**—Gateways on the Internet manage a lot of encrypted data and relay it. This data is coming in from various IoT devices that may be genuine or malicious in nature. Most of the data being measured by the genuine IoT devices is a result of measurement or is transaction-critical, meaning that this data has to be forwarded only in an encrypted manner, maintaining its **secrecy**. The malicious IoT nodes, on the other hand, are waiting for a chance to intercept any information from the network and use it for their objectives.

Hence, the number of decryptions needed while the data is in transit has to be reduced, to as much as possible, ideally zero. This is because even if a node in



the network is able to decrypt a piece of information and manipulate it, there are possibilities for an intruder to do the same as no system is fully and truly secure. Eliminating the need for decryption is the ideal approach. This is exactly what **homomorphic** cryptosystems have to offer.

The IoT network, itself, is the owner of the data-in-transit, and any computation needs to happen online. Data secrecy and security is a major factor, in addition to the whole system being fast and lightweight to increase traffic flow in the network.

In the following section, we justify the need for a lightweight and homomorphic encryption algorithm to meet these bottlenecks.

### 3 Need for a Lightweight Homomorphic Encryption

With the growth of IoT devices, there has been a tremendous increase in confidential data picked up or recorded by them. This is the reason why homomorphic encryption is so essential these days [11]. The data does not need to be decrypted before some operations are performed on it, maintaining the secrecy in the system.

#### 3.1 Need for a Lightweight Encryption

For IoT devices, which have limited computing and processing power [4], performing plaintext encryption, especially FHE [12], can put unnecessary load on the device. Lightweight cryptographic techniques are being used in various IoT devices, such as sensors, healthcare devices, contact-less smart cards, and RFID tags [13]. Two points of thought talk about the need for lightweight encryption for IoT devices:

1. **End-to-end communication efficiency** The application of lightweight encryption algorithms will allow low-resource devices to efficiently and effectively communicate in real time, with end-to-end security. In the specific case of IoT devices and homomorphic encryption, we can consider the different types of homomorphic encryptions, such as FHE, LHE, SHE, PHE, as was mentioned in [14]. For example, using a SHE can reduce the load on an IoT device by 1/1400 compared to generic FHE algorithms. Many such optimizations in selecting the cryptosystem to use can ensure lightweight, efficient end-to-end communication between two IoT nodes.
2. **Possibility of using lower-resource devices** With the shift towards more lightweight encryption algorithms, it will become increasingly simpler to incorporate lower-power devices [15] into the network. This allows for effective communication and widespread outreach with a simpler hardware implementation.

### 3.2 *Need for a Homomorphic Encryption*

Imagine an organization collecting data from IoT devices stores that data on a third-party cloud provider such as AWS or Microsoft Azure. Now, for some use cases in their application, they might need to perform some more computation on that encrypted data and store the computed data on the cloud database [16]. This task could be done in three ways by this organization:

1. **On cloud—using traditional cryptographic schemes** In this scenario, the organization could decrypt the data on the cloud, perform the computation, and then encrypt it back. This could potentially lead to some security vulnerabilities as the ciphertext is being decrypted outside the organization ecosystem, opening doors to attackers to exploit the decryption stage in some manner.
2. **Incorporate local machines** In this case, the organization could download said data on their systems, decrypt it, perform the computations, encrypt it, and then upload the data back to the cloud. This method is unnecessarily complex and will cost a lot while being time-consuming. The added latency of data transfers and computation costs related to encryption and decryption is an overhead that can be avoided using better mechanisms of online data modification.
3. **Utilize homomorphic encryption algorithms** Perhaps the most inexpensive and straightforward solution to this problem will be utilizing homomorphic encryption algorithms' services. Here, the fundamental need to decrypt the data is eliminated, allowing the organization to perform the appropriate computations on the encrypted data, which will reflect those computations when decrypted. Moreover, this computation happens online and in transit, eliminating the latency issues discussed in the point above.

Having understood the options that an organization has when trying to perform computation on their data while using a third-party provider, let us look at some points that establish why homomorphic encryption is an excellent solution to the same [17]:

1. **Maintaining security while benefiting from cloud features** Utilizing homomorphic cryptographic algorithms gives organizations the freedom to utilize cloud storage services without worrying about security problems. They also do not have to worry about efficiency and security since the HE algorithms ensure both.
2. **Collaboration and Innovation** With the availability of HE algorithms, organizations do not need to worry about data vulnerability while using third-party providers. This will allow quicker and more efficient collaboration and a broad horizon for new ideas.
3. **Security Mandates** Homomorphic encryption will allow organizations working with heavy security mandates due to sensitive data such as healthcare and finance to outsource services for logistics, analytics, research, and other requirements from third-party providers, without any risk of non-compliance.

While we have looked at the benefits and need for HE algorithms, a thing to keep in mind is that homomorphic encryption makes sense only for minimal computations such as addition/subtraction. In other higher-level computations, the cryptographic system becomes too complex [18]. Also, the HE algorithm’s encrypted data is substantially larger than traditional ciphertext, so some careful deliberations will be needed before committing to the HE system. In the following section, various security protocols for IoT networks are discussed.

### 4 Security Protocols for IoT Networks

To understand what factors affect one’s choice for a lightweight and homomorphic encryption cryptosystem in an IoT setup, it is essential to see some security protocols [19] that drive an IoT network. This section discusses these protocols in detail, following which the cryptosystems that secure these networks are elaborated upon (Fig. 1).

To date, IoT networks and protocols have been designed keeping in mind the computational handicap of the client IoT devices. What is interesting to understand is how IoT devices communicate with one another. What is even more critical to understand is how security is ensured and made available in IoT networks. In the

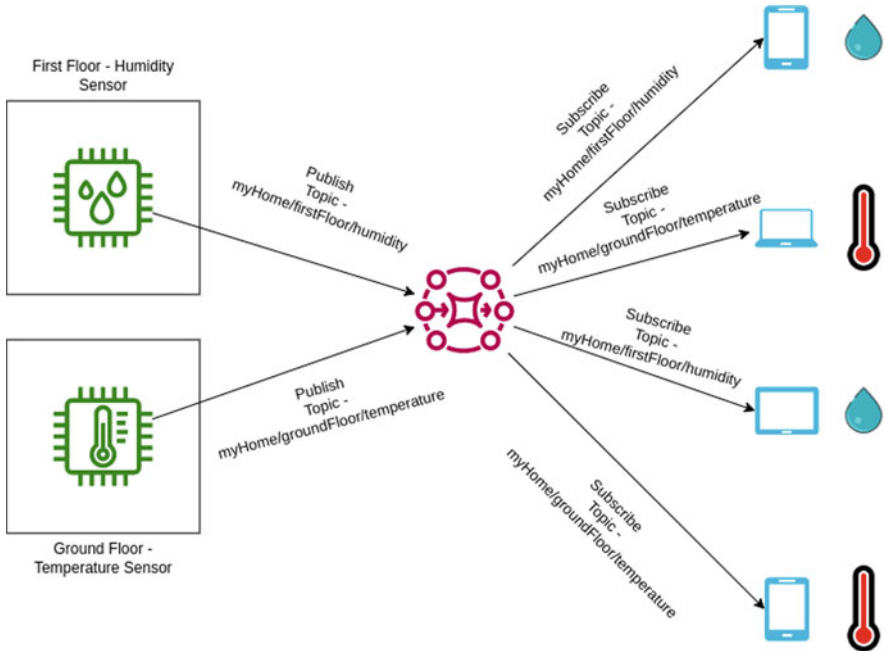


Fig. 1 The MQTT—Broker architecture

upcoming subsections, some of the most popular security protocols [MQTT, CoAP (DTLS), 6LowPAN, ZigBee] have been described.

## 4.1 Message Queuing Telemetry Transport (MQTT)

MQTT is a bi-directional, lightweight, bandwidth-efficient protocol that supports secure data transfer over an IoT network. The broker is at the heart of this protocol, and it is called the MQTT broker. It uses the publish–subscribe model to allow communication in the network.

Instead of directly sending and receiving messages to and from the devices with whom we want to communicate, the MQTT [20] architecture introduces the concept of the aforementioned introduced *broker*.

**Broker** A broker [21] is simply software running on a machine in the IoT network, which acts somewhat like a postal service. Clients that have sensors or are collecting data would want to send that data possibly to multiple devices in the network. A broker is introduced to ensure that the entire computation does not have to be handled by a single client device. Nevertheless, to understand how this information transfer happens via the broker, *topics* need to be defined.

**Topic** A topic [22] in the context of IoT networking is simply a multi-layered UTF-8 string that the broker uses to filter and send messages to different nodes in the network. Each layer brings in another level of specification towards the type of information being transmitted. As a simple example, one may want to think of a multi-purpose sensor in the living room of someone’s house. While transmitting the temperature information from the living room, the sensor will need to send the topic as part of the metadata to tell the receiving broker about the type of information that has been sent. The topic sent may look like “*Home1/GroundFloor/LivingRoom/Temperature*”. As we can see, each layer specifies more details about the type of information transmitted.

Clients can publish information about a particular topic, reaching the MQTT broker. This information is then relayed to all the devices that have subscribed to receive information that pertains to said topic. In this way, the requirement for the sender client to know the IP addresses of all devices that may need that information is eradicated, all the while easing the compute process.

In terms of security aspects [23], MQTT runs on top of SSL/TLS (Secure Socket Layer/Transport Layer Security) in the transport layer and takes full advantage of all the security features it provides. A VPN is used for a physically secure network at the network level. At the application layer, client identifiers, usernames, and passwords are utilized to authenticate and authorize devices using data packets.

## 4.2 *Constraint Application Protocol (CoAP)*

As the name suggests, CoAP or constrained application protocol is designed explicitly for those client devices that have a constraint on the amount of computing power or storage power they possess. It utilizes popular Internet protocols such as UDP to communicate with one another. They are also used in networks with low power and bandwidth to communicate. It can easily be integrated with the Hypertext Transfer Protocol (HTTP) to provide simple web support. This can also be visualized because it uses similar request types as HTTP, such as GET, POST, PUT, PATCH, and DELETE. It is different from MQTT [20] in the sense that it is not event-driven-like MQTT. It works on the request–response structure.

Talking about the security aspect, CoAP uses UDP as its transport layer protocol and uses datagram transport layer security (which will be described next) as the security algorithm on top of UDP. CoAP supports features that are multicast support (sending one packet to many hosts simultaneously), and it also has low overhead costs in terms of performance and computes requirements, making it perfect for usage in low-power secure IoT networks.

- **Datagram transport layer security (DTLS):** This is the underlying security algorithm in CoAP that secures the UDP in the transport layer. As it is a UDP protocol, it also suffers from packet reordering and loss of data. It also gains from the fact that UDP connections will have inherently lower delays and latency, ideal for IoT networks. DTLS is used in WebRTC protocols, gaming, IM, and live video feeds.

## 4.3 *IPv6 Over Low-Power Wireless Personal Area Networks (6LoWPAN)*

Similar to CoAP, 6LoWPAN is also suited to low-power networks. This can include sensor networks and IoT networks. It allows IPv6 packets to be transmitted over IEEE 802.15.4 networks. It plays an important role in multiple use cases, such as smart grids to collect data for load forecasting, automation, and motoring.

In terms of security [24], since it is exclusively for IEEE 802.15.4 networks, it benefits from the AES-128 link-layer security mechanisms it provides. Due to the higher overhead of DTLS, it is not considered for implementation in this protocol. An additional element called the 6LBR [25] (6LoWPAN Border Router) is present, which is responsible for the secure connection of the lossy network to the widely used orthodox Internet.

## 4.4 ZigBee

ZigBee is another widely used protocol in the IoT world. It is used mainly because of its battery-saving properties, making it a hit in IoT devices since they are usually low-power devices. It requires proximity, just like Bluetooth or Wi-Fi, so it is essentially a wireless ad hoc network. Like the 6LowPAN protocol, this protocol is also built on the IEEE 802.15.4 network standard. Although it is similar to WPANs such as Bluetooth, it is cheaper and more straightforward. This comes with a 10–100-metre line-of-sight limitation as to its major drawback. To overcome this issue, it uses intermediate devices to reach devices farther away. This protocol is not only battery-saving, but it is also a secure protocol since it uses 128-bit symmetric encryption keys. It is best suited for transferring data from input devices like sensors, such as home automation use cases.

In terms of the security aspect, ZigBee [26] has two security [27] possibilities:

1. **Centralized Network Security Model:** A higher security standard is set in the centralized model as it incorporates a new entity in the ZigBee network, called the Trust Centre. The trust centre acts as a security server for the ZigBee network and determines a network key. Every new device that wants to join the network is given a unique key called the TC Link Key (TCLK). Every new device is configured with this key, and it is used to encrypt the network data transfer.
2. **Distributed Network Security Model:** In this model, there is no central authority such as the trust centre. When a new device wants to join the network key, it gets the network key and nothing else to uniquely identify it, as in the previous case with the TCLK. This makes it less secure, but it is easier to design and implement.

After the above discussion, it is clear that the most critical factor in designing security protocols for IoT devices is considering the minimal compute resources available on an IoT client. This means that a heavyweight encryption algorithm will not suit security maintenance on such systems. Also, to maintain secrecy, homomorphic algorithms are more suitable to keep data confidential, while it is available to third-party providers.

## 5 A Study of Existing Lightweight and Homomorphic Cryptosystems

IoT and its developments are a reasonably new field of research, and much literature on this topic is outdated, dating back to when mathematical models for the same were being introduced. This section highlights nine of the latest, already proposed mechanisms in this field. These papers showcase the implementation of lightweight homomorphic encryption in IoT. The following subsections discuss the nine algorithms with respect to their key-generation process and encryption–decryption

process. The paper also proves the homomorphic nature of these algorithms using mathematical equations cited from the respective work in the literature and states reason for the security of these cryptosystems.

### **5.1 The ElGamal–Elliptic Curve Encryption Homomorphic Scheme**

In this section, the system discussed in [28] is described. This chapter described an IoT-friendly system based on fundamental dynamic change and proportional offloading of data to preserve energy by only selectively transferring data to the cloud according to the data topic. In terms of security, they have proposed an ElGamal–Elliptic curve cryptography method that lends a homomorphic factor to the scheme while preventing the infamous man-in-the-middle (MITM) attack. The proposed system has three steps: key generation, publish–subscribe, and encryption–decryption.

#### **Key Generation**

The following text describes the key-generation process used by the ElGamal–Elliptic curve encryption homomorphic scheme. First, a prime  $p$  is chosen, and an elliptic curve  $\mathcal{E}$  is defined on its prime field. After this, the base point  $\alpha$  on  $\mathcal{E}$  is defined. Then a secret key  $a$  is chosen, and  $\beta$  is calculated as follows:

$$\beta = \alpha.a. \quad (1)$$

To make it clear here,  $\alpha$  and  $\beta$  are both public and are now broadcasted, and  $a$  is the secret.

#### **Encryption–Decryption**

The following text describes the encryption and decryption process used by the ElGamal–Elliptic curve encryption homomorphic scheme. The data generated from the IoT devices are sent to an MQTT broker for storage and analysis. Clients are responsible for publishing and subscribing to the various topics on the MQTT broker, and all the nodes that receive data on a particular topic are subscribers of the same. In this scheme, the level of encryption applied depends on the battery level of the IoT node:

- Battery percentage of node  $> 50\%$   $\implies$  curve with high security key is used.
- Battery percentage of node  $> 25\%$   $\implies$  curve with medium security key is used.
- All other cases  $\implies$  curve with low security key is used.

Since the key type was decided based on the battery level of the IoT node, now the encryption key to be used can be generated after analysing that mapping. Once the key is generated, the data is encrypted as shown below:

- (i) A secret  $k \in Z$  is selected.
- (ii)  $y_1$  and  $y_2$  are calculated as follows:

$$y_1 = k.\alpha \quad (2)$$

$$y_2 = x + k.\beta \quad (3)$$

here,  $x$  is the plaintext that has to be encrypted.

- (iii) Now,  $y_1$  and  $y_2$  together constitute the ciphertext that will be decrypted as shown below:

$$x = y_2 - a.y_1, \quad (4)$$

where  $x$  is the same data that we had initially encrypted.

## Homomorphic Nature

Figure 2 illustrates the homomorphic nature of the ECEG algorithm. This algorithm is additive and is described as shown below:

- (i) Consider two ciphertext pairs  $C_1 = (c_1, c_2)$  and  $C_2 = (c'_1, c'_2)$ .
- (ii) According to the claim, if these ciphertexts are added, and the result is decrypted, then the obtained plaintext should also be the addition of the corresponding plaintext pair.
- (iii) For the proof, let us take  $M_1$  and  $M_2$  to be the plaintext pair that is going to be used in this cryptosystem. Then we get

$$C_1 = (\alpha k_1, M_1 + k_1\beta), \quad C_2 = (\alpha k_2, M_2 + k_2\beta). \quad (5)$$

- (iv) After adding  $C_1$  and  $C_2$ , we get

$$C_1 + C_2 = (\alpha k_1 + \alpha k_2, M_1 + k_1\beta + M_2 + k_2\beta) = ((k_1 + k_2)\alpha, (M_1 + M_2) + (k_1 + k_2)\beta). \quad (6)$$

- (v) As is seen above, the addition of these ciphertexts results in the encryption of the addition of the mentioned plaintexts.



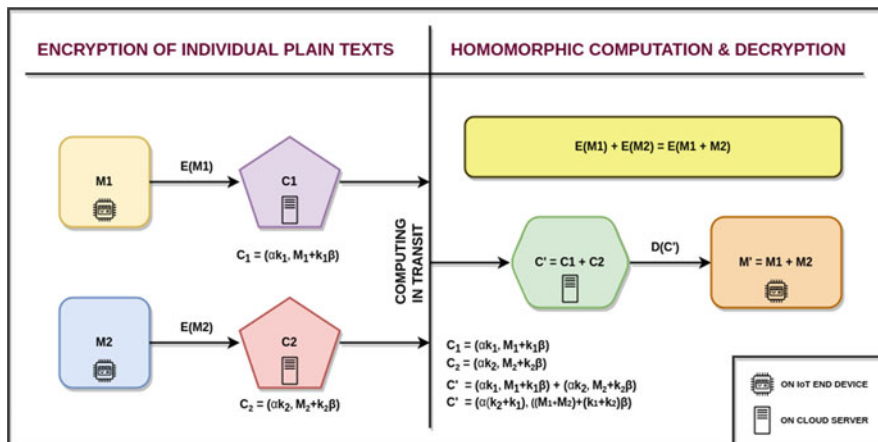


Fig. 2 Homomorphic nature of ElGamal–Elliptic curve cryptosystem (E—encryption, D—decryption)

### Security

It might seem that the more minor keys might not be able to provide enough security to prevent attacks, but this is a misconception, as the paper mentions that this system, including the shorter keys, is resistant to man-in-the-middle attacks, replay attacks, brute force attacks, and eavesdropping. It also consumed lesser energy, thanks to the dynamic key exchange and data offloading [29] based on battery levels of the node.

### 5.2 Pure ElGamal Homomorphic Cryptosystem

As seen in the above section, a combination of ECC and ElGamal algorithms can generate a homomorphic cryptosystem. Nevertheless, even a pure ElGamal cryptosystem can be used for homomorphic encryption [30]. In general, these homomorphic algorithms exploit the basic rules of exponentiation in order for them to work, but the pure ElGamal system is a little different. Here, the ciphertext pair, when multiplied, will generate another ciphertext that on decryption will give the product of the plaintext pair, and not the addition of the same, as would be expected. This is illustrated in the following subsections.

#### Key Generation

The following text describes the key-generation process used by the pure ElGamal homomorphic cryptosystem. In this phase, first, the global public elements are set.

A prime number is chosen, along with a primitive root. Let us call the prime number and the primitive root  $q$  and  $g (< q)$ , respectively. After this, a private key ( $K_{pr}$ ) is generated following the criteria shown below:

$$K_{pr} < q - 1. \quad (7)$$

After this, a part of the public key ( $K_{pu3}$ ) is generated as shown below:

$$K_{pu3} = g^{K_{pr}} \text{ mod } q. \quad (8)$$

Now that both the private and public keys are ready, i.e., Public Key =  $\{q, \alpha, K_{pu3}\}$ , Private Key =  $K_{pr}$ , it is time to analyse the encryption and decryption steps.

### Encryption–Decryption

The following text describes the encryption and decryption process used by the pure ElGamal homomorphic cryptosystem:

- (i) Let the plaintext be called  $M$ , where  $M < q$ . A random integer  $r$  is first selected.
- (ii) Then, a parameter  $h$  is calculated as shown below:

$$h = (K_{pu3})^k \text{ mod } q. \quad (9)$$

- (iii) The ciphertext in this cryptosystem is an integer pair  $C = (C_1, C_2)$  that is calculated as shown below:

$$C = (C_1, C_2) = (g^k \text{ mod } q, h * M \text{ mod } q). \quad (10)$$

- (iv) Now that the ciphertext is ready, decryption can be performed by the receiver using his/her private key as shown below:

- (a) Calculate the value of  $h$  as shown below:

$$h = (C_1)^{K_{Pr}} \text{ mod } q. \quad (11)$$

- (b) Once  $h$  has been calculated, then the plaintext  $M$  is retrieved by using the modular inverse of  $h$ .

$$M = (C_2 * K^{-1}) \text{ mod } q. \quad (12)$$

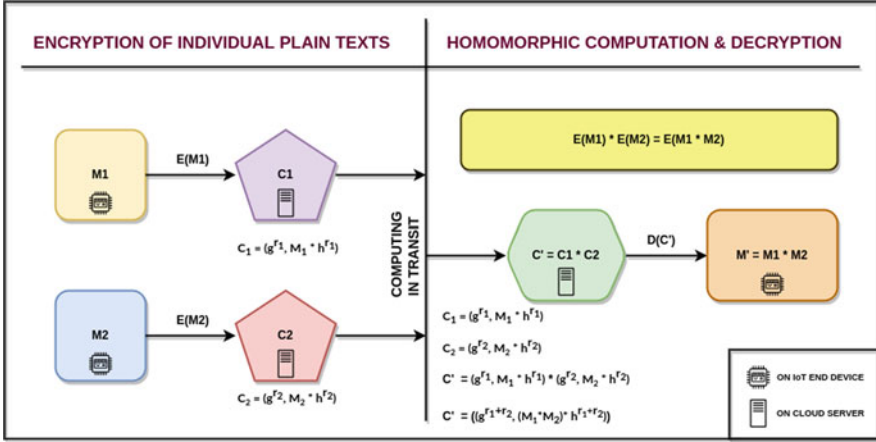


Fig. 3 Homomorphic nature of the ElGamal cryptosystem (E—encryption, D—decryption)

**Homomorphic Nature**

Figure 3 illustrates the homomorphic nature of the ElGamal cryptosystem. This cryptosystem was claimed to have been homomorphic [31], and the proof of this statement will be discussed in this section:

- (i) Consider two ciphertext pairs  $C_1 = (c_1, c_2)$  and  $C_2 = (c'_1, c'_2)$ .
- (ii) According to the claim, if these ciphertexts are multiplied, and the result is decrypted, then the obtained plaintext should also be the multiplication of the corresponding plaintext pair.
- (iii) For the proof, let us take  $M_1$  and  $M_2$  to be the plaintext pair that is going to be used in this cryptosystem. Then we get

$$C_1 = (g^{r_1}, M_1 * h^{r_1}), C_2 = (g^{r_2}, M_2 * h^{r_2}). \tag{13}$$

- (iv) After multiplying  $C_1$  and  $C_2$ , we get

$$C_1 * C_2 = (g^{r_1+r_2}, (M_1 * M_2) * (h^{r_1+r_2})). \tag{14}$$

- (v) Hence, we can see that the above result clearly resembles the ciphertext that would be obtained if  $M_1 * M_2$  was encrypted. Thus, if this result would be decrypted, we will obtain  $M_1 * M_2$ . This proves the homomorphic nature of the ElGamal cryptosystem.

## Security

The ElGamal scheme works and is widely used in industrial applications only because of the security standards [32] it maintains. Cracking the ElGamal algorithm is synonymous with cracking the discrete logarithm problem. This basically means that if an attacker wants to gather the user's private key ( $K_{pr}$ ), then he/she will have to compute  $K_{pr} = d\log_{g,q}(K_{pub})$ . If the attacker instead tries to crack the one-time key ( $h$ ), then he/she will have to know the randomly selected number  $k$ , which would once again require the cracking of the discrete logarithm  $k = d\log_{g,q}(c_1)$ . Due to the relatively large size of the selected parameters in the cryptosystem and the difficulty of handling such numbers in the discrete logarithm problem, ElGamal is a secure algorithm [33].

### 5.3 Integer-Based LHE for Mobile Cloud Networks

Much research has already been done in the field of lightweight homomorphic encryption. One such field is the field of "Cloud Computing" in devices such as mobile phones. Considering the challenges they face lately, in terms of "security" and "mobility", many suggestions and improvements have been proposed. Further, FHE techniques are expensive, in terms of both computation and power. This prevents mobile from performing the calculation efficiently and effectively. Thus, to address these problems, several methodologies have been proposed. One such methodology [34] is based on "Gentry's scheme", where the choice of plaintext was taken to be an integer and not bits, thus resulting in easy, smooth, and faster encryption. The overall efficiency seems to accelerate as the overall computation in the case of "key generation" and "Encryption" is comparatively low. It also supported both multiplication and addition as far as homomorphic encryption is concerned. All this has been done keeping security in mind, such that the data does not get leaked or exposed anywhere.

#### Key Generation

The following text describes the encryption and decryption process used by the integer-based LHE algorithm.

The proposed lightweight homomorphic encryption (LHE) scheme employs a private key for data encryption by each contributor  $\mu_i$ . The private key is shared between its associated DC and  $\mu_i$  and used for symmetric data encryption. The  $n$  private keys need to meet the following conditions:

- (i) For the summation,

$$2^l n < r \quad \text{and} \quad rf(L_i, s_i) < p. \quad (15)$$

(ii) For the product,

$$(2^l)^n < r \quad \text{and} \quad rh(L_i, e_i, r, s_i) < p. \quad (16)$$

The second part of both conditions means that the first part of  $\mu_i$ 's private key is multiplied by a summation or a combination of summation and multiplication of some values is below  $p$ . This condition also allows the summation or product result to be recovered quickly if a data recovery needs to be made for some reason.

### Encryption–Decryption

The following text describes the encryption and decryption process used by the integer-based LHE algorithm.

The data that is encrypted goes through a two-staged process. Sufficient randomness are added. The equations for the same as cited in [35] are

$$e_i = 2^{l_d + \lceil \log_2 n \rceil} . b_i + d_i \quad (17)$$

$$a_i = (e_i + L_i k_i) \text{ mod } \phi. \quad (18)$$

Here,  $a$  is the encrypted form of  $e$ , on which homomorphic computation can be performed.

$\mu$  is responsible for passing  $\alpha$  to cloud server once the computations have been performed. This is initiated for storing for future use or for performing any kind of computation that is needed. On receiving  $\alpha$ , cloud server begins to perform computations on the encrypted ciphertext to produce the required output that has been requested by the data client. Clearly, the homomorphic nature is properly visible.

The client and the DC share the key pair  $(p, r)$ . The plaintext can be obtained from this ciphertext as follows:

$$e_i = (\alpha_i \text{ mod } p) \text{ mod } r \quad (19)$$

$$d_i = \alpha_i \text{ mod } 2^{(l_d + \lceil \log_2 n \rceil)}. \quad (20)$$

Data client deciphers alpha I with the use of master keys named p and r in order to retrieve or recover the value of  $e_i$  by the usage of the equation:

$$e_i = (\alpha_i \text{ mod } p) \text{ mod } r. \quad (21)$$

Following the decryption, the data client is able to recover the plaintext that was encrypted using (Fig. 5):

$$d_i = e_i \text{ mod } 2^{l_d + \lceil \log_2 n \rceil}. \quad (22)$$

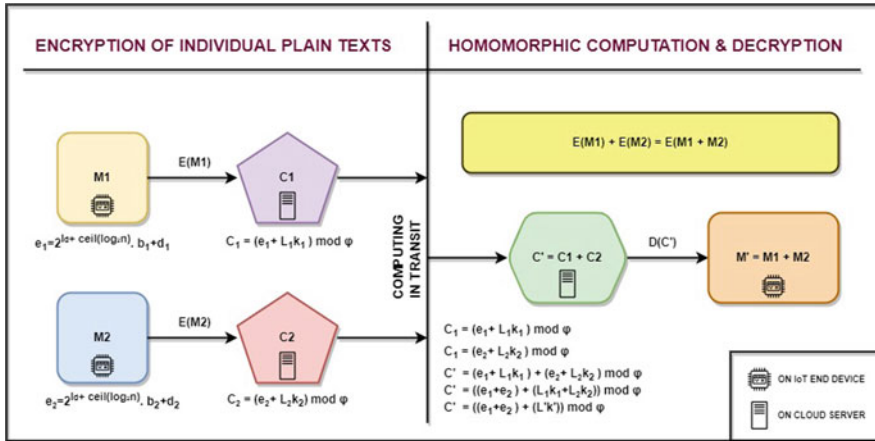


Fig. 4 Additive homomorphic nature of the integer-based LHE (E—encryption, D—decryption)

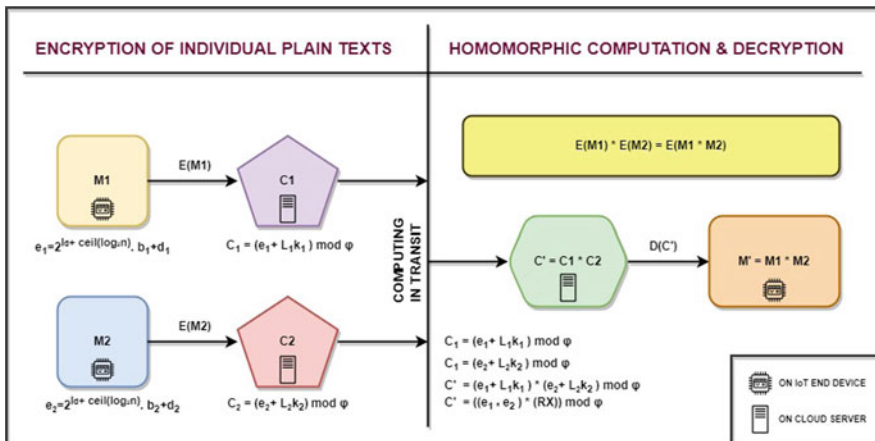


Fig. 5 Multiplicative homomorphic nature of the integer-based LHE (E—encryption, D—decryption)

### Homomorphic Nature

Figure 4 illustrates the additive homomorphic nature of the LHE cryptosystem. The additive homomorphism can be shown by using the data recovery technique as follows.

Compute the intermediate data representation  $e'$ .

$$e' = f(\alpha_1 + \alpha_2) \text{ mod } p \text{ mod } r \tag{23}$$

$$e' = ((e_1 + L_1 k_1) + (e_2 + L_2 k_2)) \bmod \phi \bmod p \bmod r \quad (24)$$

$$e' = ((e_1 + e_2) + (\sum L_i k_i)) \bmod \phi \bmod p \bmod r \quad (25)$$

$$e' = ((e_1 + e_2) + (\sum L_i (rs_i + pq_i))) \bmod \phi \bmod p \bmod r \quad (26)$$

$$e' = ((e_1 + e_2) + (\sum L_i (rs_i + pw'))) \bmod p \bmod r \quad (27)$$

$$e' = (e_1 + e_2). \quad (28)$$

Here,  $\sum (L_i pq_i) \bmod pw = pw'$  and  $f(\alpha) = (e + Lk) \bmod \phi$  as seen earlier.

Use  $e'$  to find  $d'$  and verify the result.

$$d' = (e_1 + e_2) \bmod 2^{l_d + \lceil \log_2 n \rceil} \quad (29)$$

$$d' = \sum (2^{l_d + \lceil \log_2 n \rceil} . b_i + d_i) \bmod = (e_1 + e_2) \bmod 2^{l_d + \lceil \log_2 n \rceil} \quad (30)$$

$$d' = \sum d_i. \quad (31)$$

Similarly, one can verify multiplicative homomorphism as well as seen in [35].

## Security

The security of the integer-based lightweight homomorphic encryption lies in the fact that the length of  $rs_1/rs_2$  is supposed to be large enough but at the same time smaller than chosen safe prime number  $p$ , hence ensuring proper decryption of the plaintext or the message encrypted. Thus, this becomes helpful in eradicating the chances of Brute force attack.

## 5.4 Goldwasser–Micali Encryption Scheme

The Goldwasser–Micali (GM) encryption scheme [36] is a public key algorithm. Traditionally, it holds the significance of being the first probabilistic asymmetric key algorithm that is provably secure, given standard cryptographic assumptions using semantic security concepts. This is because it is challenging for an adversary to solve the quadratic residuosity problem modulo a composite  $N = p.q$ , where  $p, q$  are large primes. The GM cryptosystem generates a value of sizes close to  $|N|$  for every single bit of the plaintext. Hence, a massive expansion takes place

during the generation of the ciphertext. Due to this reason, this technique is not considered efficient as the ciphertext can be several hundred times larger than the input plaintext. It was based on this scheme that the ElGamal cryptosystem was crafted. Classically, this scheme serves as a proof of concept, analogous to what the Feistel structure meant for DES.

## Key Generation

The following text describes the encryption and decryption process used by the Goldwasser–Micali cryptosystem.

In this phase, the receiver generates two primes ( $p$  and  $q$ ) and computes two other values—( $a$  and  $N$ ). These pairs respectively serve as the private and public keys for the receiver:

- (i) The receiver generates two random prime numbers  $p$  and  $q$  such that

$$p = q = 3 \text{ mod } 4. \quad (32)$$

- (ii) The receiver then generates a residue  $a$  using the technique shown below:

- (a) Compute arbitrary  $a_p$  and  $a_q$  as follows:

$$a_p = a \text{ mod } p \quad (33)$$

$$a_q = x \text{ mod } q. \quad (34)$$

- (b) If the following conditions hold, then  $a$  is a valid quadratic residue mod  $N = p.q$

$$a_p^{(p-1)/2} = 1 \text{ mod } p \quad (35)$$

$$a_q^{(q-1)/2} = 1 \text{ mod } q. \quad (36)$$

- (iii)  $N$  is computed as

$$N = p.q. \quad (37)$$

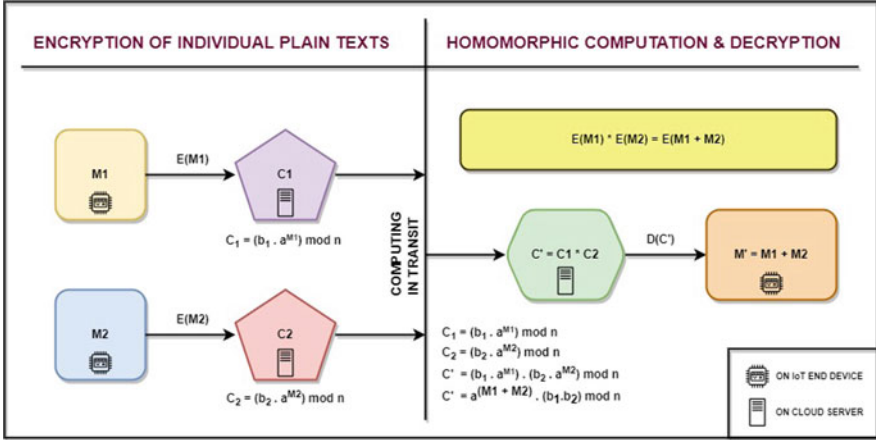
- (iv) Now, the receiver has  $(a, N)$  as the public key and  $(p, q)$  as the private key.

## Encryption–Decryption

The following text describes the encryption and decryption process used by the Goldwasser–Micali cryptosystem.

The sender now has the pair  $(a, N)$  as the receiver's public key:





**Fig. 6** Homomorphic nature of the Goldwasser–Micali cryptosystem (**E**—encryption, **D**—decryption)

- (i) The sender encodes his message  $m$  as a stream of bits  $(m_1, m_2, m_3, \dots, m_n)$ .
- (ii) For every bit in  $m$ , the sender generates a random value  $b_i$  such that

$$gcd(b_i, N) = 1. \tag{38}$$

- (iii) The generated ciphertext can be given by

$$c_i = b_i^2 \cdot a^{m_i} \pmod{N}. \tag{39}$$

- (iv) This ciphertext is sent over to the receiver.

The receiver uses the pair  $(p, q)$  to decrypt the ciphertext received:

- (i) For each  $i$ , using the prime factorization  $(p, q)$ , the receiver determines whether the value  $c_i$  is a quadratic residue as per the equations in key-generation step above.
- (ii) If so,  $m_i = 0$ , otherwise  $m_i = 1$ .
- (iii) The receiver now has the message  $m = (m_1, m_2, \dots, m_n)$ .

**Homomorphic Nature**

Figure 6 illustrates the homomorphic nature of the Goldwasser–Micali cryptosystem. This technique is additively homomorphic.

If  $c_1 = b_1^2 \cdot a^{m_1} \pmod{N}$ ,  $c_2 = b_2^2 \cdot a^{m_2} \pmod{N}$ , then  $c_1 \cdot c_2$  actually gives us the encryption for  $m_1 \oplus m_2$ .

$$c_1 \cdot c_2 = (b_1^2 \cdot a^{m_1}) \cdot (b_2^2 \cdot a^{m_2}) \bmod N \quad (40)$$

$$c_1 \cdot c_2 = (b_1 \cdot b_2)^2 \cdot a^{(m_1+m_2)} \bmod N. \quad (41)$$

It is to be noted that this addition holds homomorphically because all operations are on bits of the message. Hence, XOR translates to bit-wise addition mod 2.

## Security

The scheme relies on deciding whether a given value  $x$  is a square mod  $N$ , given  $N$ 's factorization  $(p, q)$ .

Semantic security involves hiding the plaintext and ciphertext relationship so that even partial information about the plaintext is complicated to procure. This involves adding an element of randomness in the key selection algorithm instead of using a deterministic algorithm. Even deterministic algorithms can be made randomized by the use of padding as done in padding RSA [37]. As described earlier in the key-generation step, solving the quadratic residuosity problem is a complex challenge that secures the GM cipher due to the arbitrary choice of  $p$  and  $q$  as seen in the key-generation step below.

## 5.5 Benaloh Cryptosystem

The Benaloh Cryptosystem [38] is another popular homomorphic encryption scheme used in IoT devices and applications. It is an extension to the Goldwasser–Micali [36] encryption scheme discussed above. The main difference between the Benaloh and GM cryptosystem is the fact that Benaloh encrypts a block of data at once, whereas GM only encrypts one bit at a time. One fact to keep in mind is that this scheme works in the group  $(\mathbb{Z}/n\mathbb{Z})^*$ .

### Key Generation

The following text describes the encryption and decryption process used by the Benaloh algorithm.

The key-generation phase in this scheme is very similar to other public key cryptosystems such as RSA, GM, etc. The following steps are followed to generate the public and private keys:

- (i) A block size  $r$  is chosen.
- (ii) Now, two large primes  $p$  and  $q$  are selected so that they follow the following conditions:

- (a)  $r \bmod (p - 1) \equiv 0$ .
  - (b)  $\gcd(r, \frac{p-1}{r}) = 1$
  - (c)  $\gcd(q - 1, r) = 1$ .
- (iii) Next, a parameter  $y \in (Z/nZ)^*$  is selected such that  $y^{\frac{(p-1)*(q-1)}{r}} \not\equiv 1 \pmod{n}$ , where  $n = p * q$ .
- (iv) The public key ( $K_{pu}$ ) is set as  $K_{pu} = \{y, n\}$ , and the private key ( $K_{pr}$ ) is set as  $K_{pr} = \{p, q\}$ .

### Encryption–Decryption

The following text describes the encryption and decryption process used by the Benaloh algorithm:

- (i) Let the plaintext be called  $M$ . A random integer  $u \in (Z/nZ)^*$  is first selected.
- (ii) The ciphertext  $C$  is constructed using the selected/generated parameters as shown below:

$$C = y^m u^r \bmod n. \quad (42)$$

- (iii) To decrypt the ciphertext  $C$ , the following steps are followed:

- (a) Parameter  $t$  is calculated first, as shown below:

$$t = C^{\phi/r} \bmod n, \quad (43)$$

where  $\phi = (p - 1) * (q - 1)$ .

- (b) Using the calculated  $t$ ,  $M$  is calculated as shown:

$$M = \log_x(t). \quad (44)$$

For this, the baby-step, giant-step algorithm can be used.

- (c) This works only because, for any  $m, u$ :

$$t = C^{\phi/r} \equiv (y^m u^r)^{\phi/r} \equiv (y^M)^{\phi/r} (u^r)^{\phi/r} \equiv (y^{\phi/r})^M (u)^\phi \equiv (x)^M (u)^0 \equiv (x)^M \bmod n. \quad (45)$$

### Homomorphic Nature

Figure 7 illustrates the homomorphic nature of the Benaloh cryptosystem. The same can be demonstrated as follows:

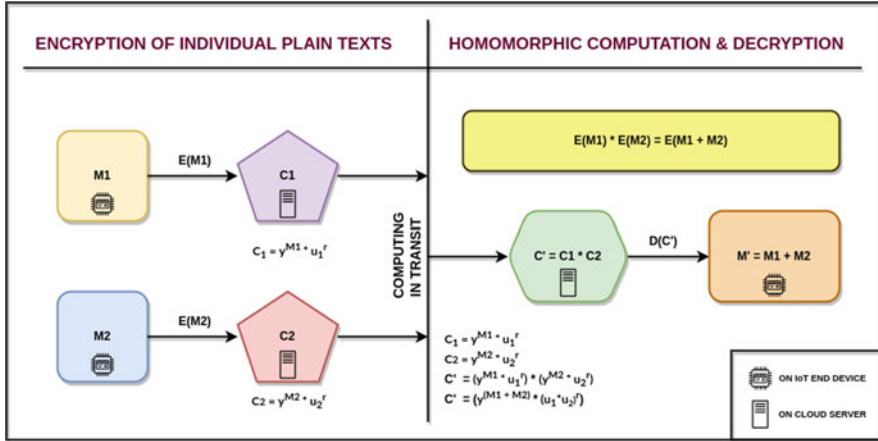


Fig. 7 Homomorphic nature of the Benaloh cryptosystem (E—encryption, D—decryption)

- (i) Consider two ciphertext messages  $C_1, C_2$ .
- (ii) In this cryptosystem, multiplying the two ciphertext messages will result in a ciphertext that on decryption will give the addition of their respective plaintexts.
- (iii) For the proof, let us take  $M_1$  and  $M_2$  to be the plaintext messages that are going to be used in this cryptosystem. Then we get

$$C_1 = (y^{M_1} * u_1^r), C_2 = (y^{M_2} * u_2^r). \tag{46}$$

- (iv) After multiplying  $C_1$  and  $C_2$ , we get

$$C_1 * C_2 = (y^{(M_1+M_2)} * (u_1 u_2)^r). \tag{47}$$

- (v) The above expression is the exact encryption of  $M_1 + M_2$  using the Benaloh cryptosystem. This proves the homomorphic property [39] of the cryptosystem.

### Security

The Benaloh cryptosystem works primarily because it is difficult to factorize large prime numbers, like other algorithms such as RSA. Nevertheless, it is not exactly the integer factorization problem that needs to be solved to crack the Benaloh system. A similar problem called the higher residuosity problem would need to be solved to crack Benaloh. This problem essentially states that given  $C, r, n$ , it is computationally infeasible to check if there exists a  $h$ , such that  $C = h^r \text{ mod } n$ .

## 5.6 Okamoto–Uchiyama Cryptosystem

This cryptosystem is similar to the Benaloh Cryptosystem, discussed above. This system works on the group  $(Z/nZ)^*$ , where  $n$  is of the form  $p^2q$  for very large  $p$  and  $q$  primes. This encryption scheme is additively homomorphic.

### Key Generation

The following text describes the key-generation process used by the Okamoto–Uchiyama Cryptosystem.

In this phase, the receiver generates public and private keys for the sender. The process is as follows:

- (i) The receiver chooses two large primes  $(p, q)$  arbitrarily.
- (ii)  $N$  is then calculated as

$$n = p^2q. \quad (48)$$

- (iii) Choose a  $g$  from the group such that

$$g^p \not\equiv 1 \pmod{p^2}. \quad (49)$$

- (iv) Compute

$$h = g^n \pmod{n}. \quad (50)$$

- (v) The public key is now  $(n, g, h)$ , and the private key is  $(p, q)$ .

### Encryption–Decryption

The following text describes the encryption and decryption process used by the Okamoto–Uchiyama Cryptosystem.

The sender now has the tuple  $(n, g, h)$  as the receiver's public key:

- (i) The sender selects a random number  $r$  belonging to the group  $(Z/nZ)^*$ .
- (ii) The ciphertext can be computed as

$$C = g^m h^r \pmod{n}. \quad (51)$$

The receiver uses the pair  $(p, q)$  to decrypt the ciphertext received. The plaintext can be computed as:

- (i) Compute  $a$  as

$$a = \frac{(C^{p-1} \bmod p^2) - 1}{p} \tag{52}$$

(ii) Compute  $b$  as

$$a = \frac{(g^{p-1} \bmod p^2) - 1}{p} \tag{53}$$

(iii) Find  $b^{-1}$  using  $b' = b^{-1} \bmod p$ .

(iv) Finally, the plaintext can be obtained as

$$P = ab' \bmod p. \tag{54}$$

### Homomorphic Nature

Figure 8 illustrates the homomorphic nature of the Okamoto–Uchiyama cryptosystem. The homomorphic property can be seen from the following equations:

$$C_1 = g^{M_1} . h^{r_1} \bmod n \tag{55}$$

$$C_2 = g^{M_2} . h^{r_2} \bmod n \tag{56}$$

$$C_3 = C_1 * C_2 = (g^{M_1} . h^{r_1}) . (g^{M_2} . h^{r_2}) \bmod n = (g^{M_1+M_2} . h^{(r_1+r_2)}) \bmod n. \tag{57}$$

Thus, this technique is visibly “additively homomorphic”.

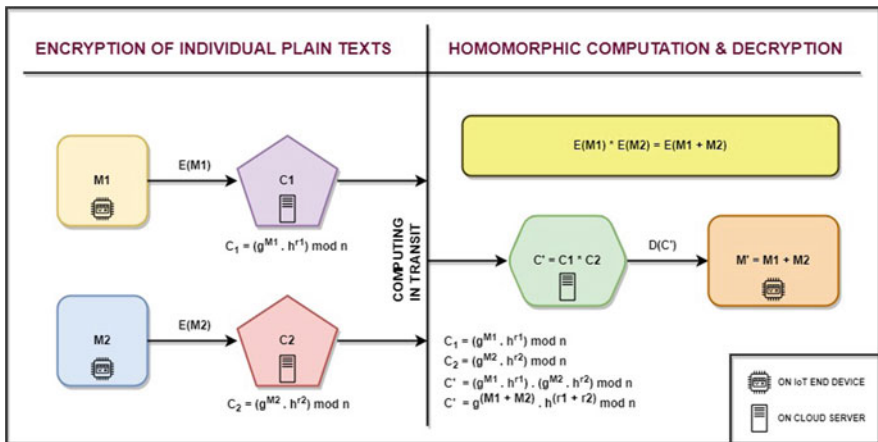


Fig. 8 Homomorphic nature of the Okamoto–Uchiyama cryptosystem (E—encryption, D—decryption)

## Security

The Okamoto–Uchiyama cryptosystem works primarily because it is difficult to factorize large numbers into prime factors, like other algorithms such as RSA. Just like Benaloh, given  $C$ ,  $g$ ,  $h$ ,  $n$ , it is computationally infeasible to compute a  $p$  or  $q$ .

## 5.7 Unpadded RSA

RSA is one of the oldest yet most popularly used algorithms supporting secure data transmission. Unpadded RSA [40] is not as secure as usual/padded RSA, but adding padding to introduce randomness in RSA strips it off of its homomorphic property. Hence, if top-notch security is unnecessary, unpadded RSA is just fine for IoT networks.

### Key Generation

The following text describes the key-generation process used by the unpadded RSA algorithm:

- (i) In the phase of key generation, two prime numbers  $p$  and  $q$  are selected such that  $p \neq q$ . Further, calculation of the value of “ $n$ ”,

$$n = p \cdot q \quad (58)$$

is performed.

- (ii) This is followed by the computation of the totient function,

$$n = (p - 1) \cdot (q - 1). \quad (59)$$

- (iii) After this, an integer “ $e$ ” is chosen such that

$$\gcd(\phi(n), e) = 1, 1 < e < \phi(n). \quad (60)$$

- (iv) Next, the value of “ $d$ ” can be easily computed using the property,

$$d \equiv e^{-1} \pmod{\phi(n)}. \quad (61)$$

- (v) Thus, this gives us the private and public keys of the form,

$$\text{Public Key} = (e, n) \quad (62)$$

$$\text{Private Key} = (d, n). \quad (63)$$

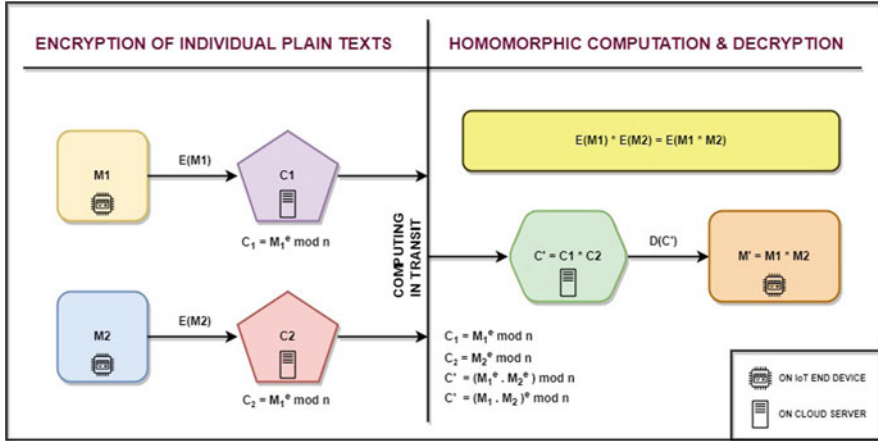


Fig. 9 Homomorphic nature of the RSA cryptosystem (E—encryption, D—decryption)

### Encryption–Decryption

The following text describes the encryption and decryption process used by the unpadded RSA algorithm.

The sender uses public key  $(e, n)$  to encrypt the plaintext  $M$ , where  $M < n$

$$C = M^e \bmod n. \tag{64}$$

This ciphertext is then sent over to the receiver, and the decryption is performed using the private key  $(d, n)$

$$M = C^d \bmod n. \tag{65}$$

### Homomorphic Nature

Figure 9 illustrates the homomorphic nature of the unpadded RSA cryptosystem. The homomorphic property can be better understood by the means of the following equations:

$$C_1 = M_1^e \bmod N \tag{66}$$

$$C_2 = M_2^e \bmod N \tag{67}$$

$$C_3 = C_1 * C_2 = (M_1^e * M_2^e) \bmod N = (M_1 * M_2)^e \bmod N. \tag{68}$$



As seen here, homomorphic multiplication is supported by this cryptosystem, but not addition. Hence, this is a reasonably lightweight partially homomorphic algorithm.

## Security

RSA is a public key encryption strategy that takes advantage of the fact that it is a challenging task to factorize a large composite number  $N$  into its component primes  $p$  and  $q$  where  $N = p \cdot q$  is the only valid prime factorisation of  $N$ . The server keeps the numbers  $p$  and  $q$  private and sends out a public key. It is assumed that no adversary can figure out the values for  $p$  and  $q$  in a reasonable time from the public information available to them. This probabilistic and semantic security secures RSA (Table 1).

Thus, this table compares different algorithms extensively based on different aspects of their execution.

## 5.8 Paillier Cryptosystem

This is a reasonably old algorithm that supports secure data transmission. It [42] is also a public key cryptosystem, such as RSA, and its security lies in the fact that finding the  $n$ -th class residue of a large number is difficult. The decisional composite residuosity assumption [26] is something that is shared by many cryptosystems and used as their proof of work and security. This system allows addition as the only homomorphic operation performed on the ciphertext, preserving its secrecy and privacy. There is no way to perform multiplication on the ciphertexts without the knowledge of the private key.

### Key Generation

The following text describes the key-generation process used by the Paillier cryptosystem.

In this phase, the receiver generates two primes ( $p$  and  $q$ ) and computes four other values— $(\lambda, \mu, n, g)$ . This key-generation step is also the reason for the algorithm's security:

- (i) The receiver generates two random prime numbers  $p$  and  $q$  such that

$$\gcd(pq, (p-1)(q-1)) = 1. \quad (69)$$

- (ii) The value of  $N$  is computed as

**Table 1** Summary—Study of Homomorphic Cryptosystems

Sl.No.	Algorithm	HE operations			Encryption type		Security	Drawback
		ADD	MUL	ASYMM	SYMM	ASYMM		
1	EIGamal–ECC	YES	NO	NO	YES	ECC keys are more difficult to crack, and ElGamal can only be cracked by solving the discrete logarithm problem	Difficult to integrate into applications without intricate internal knowledge of scheme	
2	Pure ElGamal	NO	YES	NO	YES	Solving the discrete logarithm problem is infeasible for larger numbers	Requires randomness and encryption is time-consuming	
3	Integer LHE	YES	YES	YES	NO	The security of the integer-based LHE lies in the fact that the length of $r_{s1}/r_{s2}$ is supposed to be large enough, but at the same time smaller than $p$ .	The security of the algorithm is limited to the size of chosen safe prime number $p$	
4	Goldwasser–Micali	YES	NO	NO	YES	The scheme relies on deciding whether a given value $x$ is a square mod $N$ , given the factorization $(p, q)$ of $N$	A major disadvantage of the Goldwasser–Micali scheme is the message expansion by a factor of $\lg n$ bits	
5	Benaloh	NO	YES	NO	YES	Higher residuosity problem is difficult to crack for large numbers	Solving the higher residuosity problem is less secure than the integer factorization problem	
6	Okamoto–Uchiyama	YES	NO	NO	YES	Integer factorization of a large composite number is considered infeasible	While decryption is faster than RSA, encryption is much slower. This will increase some load on the end IoT devices.	

(continued)

**Table 1** (continued)

Sl.No.	Algorithm	HE operations		Encryption type		Security	Drawback
		ADD	MUL	SYMM	ASYMM		
7	Paillier	NO	YES	NO	YES	Deciding composite residuosity is a difficult problem to solve, and guessing the random number $r$ is synonymous to solving the discrete logarithm problem.	Decryption is compute-heavy due to the large prime numbers involved
8	Boneh-Goh-Nissim	YES	NO	NO	YES	Integer factorization of a large composite number is considered infeasible, and computation of discrete logarithm is also difficult.	Difficult to integrate into applications without intricate internal knowledge of scheme as it involves ECC-like computation
9	Unpadded RSA	NO	YES	NO	YES	Integer factorization of a large composite number is considered infeasible	RSA can be semantically secure or homomorphic, not both. Unpadded RSA can easily be broken as checking a plaintext guess is trivial.

- ADD—additive homomorphism
- MUL—multiplicative homomorphism
- SYMM—symmetric cryptosystem
- ASYMM—asymmetric cryptosystem [41]

$$N = p.q. \quad (70)$$

(iii) The value of  $\lambda$  is computed as

$$\lambda = lcm((p - 1).(q - 1)). \quad (71)$$

(iv) The receiver selects a random number  $g$  such that the following modular multiplicative inverse exists.

$$\mu = (L(g^\lambda \bmod n^2))^{-1} \bmod n, \quad (72)$$

where  $L(a) = (a - 1)/n$ .

(v) Now, the receiver has  $(n, g)$  as the public key and  $(\lambda, \mu)$  as the private key.

### Encryption–Decryption

The following text describes the encryption and decryption process used by the Paillier cryptosystem.

The sender now has the pair  $(n, g)$  as the receiver's public key:

- (i) The sender encodes his message  $m$  where  $0 \leq m < n$ .
- (ii) The sender selects a random number  $r$  where  $0 < r < n$ .
- (iii) The ciphertext can now be computed as

$$C = g^m . r^n \bmod n^2. \quad (73)$$

(iv) This ciphertext is sent over to the receiver.

The receiver uses the pair  $(\lambda, \mu)$  to decrypt the ciphertext received. The plaintext can be computed as

$$P = L(C^\lambda \bmod n^2) . \mu \bmod n. \quad (74)$$

### Homomorphic Nature

Figure 10 illustrates the homomorphic nature of the Paillier cryptosystem. The homomorphic property of Paillier can be seen from the following equations:

$$C_1 = g^{M_1} . r_1^n \bmod n^2 \quad (75)$$

$$C_2 = g^{M_2} . r_2^n \bmod n^2 \quad (76)$$

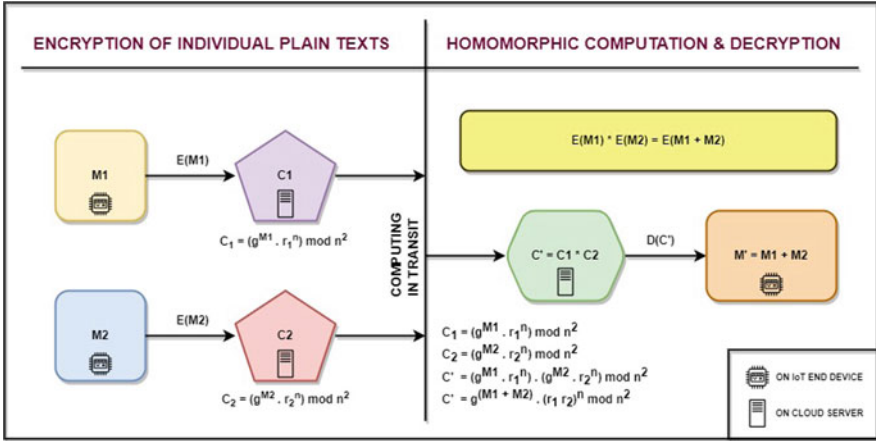


Fig. 10 Homomorphic nature of the Paillier cryptosystem (E—encryption, D—decryption)

$$C_3 = C_1 * C_2 = (g^{M_1} . r_1^n) . (g^{M_2} . r_2^n) \text{ mod } n^2 = (g^{M_1+M_2} . (r_1 . r_2)^n) \text{ mod } n^2. \tag{77}$$

Thus, this technique is visibly “additively homomorphic”.

**Security**

The scheme relies on the fact that deciding composite residuosity is a difficult problem to solve [5]. In other words, it is hard to decide whether a number  $z$  is an  $n$ -residue modulo  $n^2$  if  $z = y^n \text{ mod } n^2$ , given  $z, n$ . Here, every ecosystem user knows  $g$  and  $n$  but determining the values for  $m$  and guessing the random number  $r$  are synonymous with solving the discrete logarithm problem for  $C = g^m . r^n \text{ mod } n^2$ , which is considered hard to solve for the large numbers that this cryptosystem uses.

**5.9 Boneh–Goh–Nissim Encryption Scheme**

Boneh–Goh–Nissim encryption [43] technique adopts a similar approach as Paillier and Okamoto–Uchiyama. The BGN scheme was the first technique to bring operations such as addition and multiplication to light. It uses the encrypted ciphertext with a fixed size. Elliptic curves make the operation of multiplication easy and possible. This happens because one can easily illustrate the pairings for the same.

## Key Generation

The following text describes the encryption and decryption process used by the Boneh–Goh–Nissim cryptosystem:

- (i) The sender chooses two unique large prime numbers  $q_1, q_2$  and a cyclic group  $G$  of the order  $N = q_1 \cdot q_2$ .
- (ii) A map  $e$  is chosen that defines the additive pairing: defined as,  $G \times G \rightarrow G$ .
- (iii) A tuple is then generated, defined as:  $(q_1; q_2; G; G_1; e)$ .
- (iv) The sender selects two random generators  $g, u$  from  $G$  and computes  $h = u_2^q$ .
- (v) The public key would then be defined as:  $\{N, G, G_1, e, g, h\}$ .
- (vi) The private key would then be defined as:  $q_1$ .

## Encryption–Decryption

The following text describes the encryption and decryption process used by the Boneh–Goh–Nissim cryptosystem.

Consider the message space to be a set of integers in the range, 0 to  $T$ . When  $T = 1$ , encryption occurs. A random  $r$  is chosen from the range  $1 - N$ . The ciphertext is generated as

$$C = g^m h^r. \quad (78)$$

To retrieve the plaintext back from the ciphertext, the private key  $q_1$  is used.

$$P = C^{q_1} = (g^m h^r)^{q_1} = (g_1^q)^m. \quad (79)$$

To recover the message  $m$ , it suffices to compute the discrete logarithm of  $C^{q_1}$  to the base  $g^{q_1}$ , since  $0 \leq m < T$ , which takes time in the order of  $O(\sqrt{T})$ .

## Homomorphic Nature

Figure 11 illustrates the homomorphic nature of the Boneh–Goh–Nissim cryptosystem. This property can be demonstrated as shown below.

Given two encrypted ciphertexts  $C_1, C_2$ , a successful generation of an encryption  $m_1 + m_2 \pmod{N}$  is possible. It is not surprising that the computation is uniformly distributed. The computation can be done as follows:

$$C = C_1 C_2 H^r \text{ for a value of } r \text{ that is picked up randomly from the range } 1 - N.$$

This is true because:  $C_1 C_2 h^r = (g_1^m h_1^r)(g_2^m h_2^r) h^r = g^{(m_1 + m_2)} h^{(r_1 + r_2 + r)}$  is clearly an encryption of the summation  $m_1$  and  $m_2$  ( $m_1 + m_2$ ). Thus, the BGN technique is visibly “additively homomorphic”.

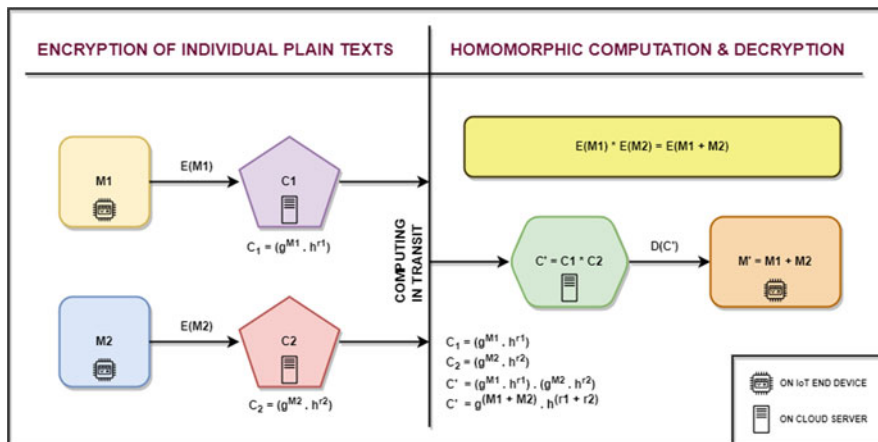


Fig. 11 Homomorphic nature of the Boneh-Goh-Nissim cryptosystem (E—encryption, D—decryption)

### Security

The security of BGN is based on the subgroup decision problem, which states that given a group  $G$  of composite order  $n$  that factorizes into two distinct random prime numbers only, distinguishing whether an element  $X$  is an element of a subgroup  $G_p$  or is an element outside it, in the group  $G$ , is a complex problem to solve. This provides BGN with the probabilistic security it needs.

## 6 Conclusion

This chapter studies, in great detail, various aspects of lightweight and homomorphic cryptosystems for IoT devices and their networks. Comparisons have been drawn between them based on their usage, working, security, drawbacks, computation involved, and extent of homomorphism supported by them. From our comparison and study results, one can see the needs and trade-offs of a homomorphic encryption system. If one had to choose which cryptosystem suits their IoT setup, the significant factors to consider are the compute capacity of the IoT devices and the kind of application that the cryptosystem will be exposed to and whether an additive or multiplicative homomorphism is desired. In future, as the IoT devices get more powerful and real time, manufacturers will open the doors for the usage of FHE algorithms on IoT devices. Nevertheless, this future is an ideal-case dream as of now.

## 7 Future Work

Apart from the theoretical comparison shown in this chapter, additional efforts can be put into standard implementations of these algorithms to compare them technically. There is still a need to compare these implementations on metrics such as time of execution, security, and ciphertext length, and many more metrics that these algorithms can be studied upon. Apart from this, a lot more powerful algorithms can be looked at for better research and examination in IoT.

## References

1. A. Yadav, L. B. Prasad, IOT Devices for Control Applications: A Review), <https://doi.org/10.1109/ICECA.2019.8821895> (2019).
2. S. Ullah, B. Rinner, L. Marcenaro, Smart cameras with onboard signcryption for securing IoT applications), <https://doi.org/10.1109/GIOTS.2017.8016279> (2017).
3. D. Sehrawat, N. S. Gill, Deployment of IoT based smart environment: Key issues and challenges, <https://doi.org/10.14419/ijet.v7i2.9504> (2018).
4. S. I. A. Sharekh, K. H. A. A. Shqeerat, Security Challenges and Limitations in IoT Environments, [http://paper.ijcsns.org/07\\_book/201902/20190224.pdf](http://paper.ijcsns.org/07_book/201902/20190224.pdf) (2019).
5. M. Pham, K. Xiong, A survey on security attacks and defense techniques for connected and autonomous vehicles, <https://doi.org/10.1016/j.cose.2021.102269> (2021).
6. B. Vankudoth, D. Vasumathi, Homomorphic Encryption Techniques for securing Data in Cloud Computing: A Survey, <https://doi.org/10.5120/ijca2017913063> (2017).
7. X. Yi, R. Paulet, E. Bertino, Homomorphic Encryption and Applications, <https://link.springer.com-nitks.knimbus.com/book/10.1007/978-3-319-12229-8> (2014).
8. I. Jabbar, S. N. Alsaad, Design and Implementation of Secure Remote e-Voting System Using Homomorphic Encryption, [https://doi.org/10.6633/IJNS.201709.19\(5\).06](https://doi.org/10.6633/IJNS.201709.19(5).06) (2017).
9. H. Li, T. Jing, A Lightweight Fine-Grained Searchable Encryption Scheme in Fog-Based Healthcare IoT Networks, <https://www.hindawi.com/journals/wcmc/2019/1019767/> (2019).
10. E. Bertino, K.-K. R. Choo, D. Georgakopolous, S. Nepal, Internet of Things (IoT): Smart and Secure Service Delivery, <https://doi.org/10.1145/3013520> (2016).
11. G. Peralta, R. G. Cid-Fuentes, J. Bilbao, P. M. Crespo, Homomorphic Encryption and Network Coding in IoT, <https://www.mdpi.com/2079-9292/8/8/827/pdf-vor> (2019).
12. M. A. Will, R. K. Ko, Chapter 5 - A guide to homomorphic encryption, <https://doi.org/10.1016/B978-0-12-801595-7.00005-7> (2015).
13. D. Maimut, K. Ouafi, Lightweight Cryptography for RFID Tags, <https://doi.org/10.1109/MSP.2012.43> (2012).
14. M. Matsumoto, M. Oguchi, Speeding Up Encryption on IoT Devices Using Homomorphic Encryption, <https://ieeexplore.ieee.org/document/9556230> (2021).
15. J. Henkel, S. Pagani, H. Amrouch, L. Bauer, F. Samie, Ultra-Low Power and Dependability for IoT Devices (special session paper), [https://www.researchgate.net/publication/312214220\\_Ultra-Low\\_Power\\_and\\_Dependability\\_for\\_IoT\\_Devices\\_Special\\_session\\_paper](https://www.researchgate.net/publication/312214220_Ultra-Low_Power_and_Dependability_for_IoT_Devices_Special_session_paper) (2017).
16. A. Siddiq, A. Karim, A. Gani, Big data storage technologies: a survey, <https://link.springer.com/article/10.1631/FITEE.1500441> (2017).
17. J. Zouari, M. Hamdi, T.-H. Kim, A privacy-preserving homomorphic encryption scheme for the Internet of Things, <https://doi.org/10.1109/IWCMC.2017.7986580> (2017).
18. G. Peralta, R. G. Cid-Fuentes, J. Bilbao, P. M. Crespo, Homomorphic Encryption and Network Coding in IoT Architectures: Advantages and Future Challenges), <https://www.mdpi.com/2079-9292/8/8/827/pdf-vor> (2019).



19. J. Cynthia, H. P. Sultana, M. N. Saroja, J. Senthil, Security Protocols for IoT, [https://doi.org/10.1007/978-3-030-01566-4\\_1](https://doi.org/10.1007/978-3-030-01566-4_1) (2018).
20. U. Hunkeler, H. L. Truong, A. Stanford-Clark, MQTT-S – A publish/subscribe protocol for wireless Sensor networks, <https://ieeexplore.ieee.org/abstract/document/4554519> (2008).
21. S. Gruener, H. Koziolok, J. Rückert, Towards Resilient IoT Messaging: An Experience Report Analyzing MQTT Brokers, <https://doi.org/10.1109/ICSA51549.2021.00015> (2021).
22. N. Tantitharanukul, K. Osathanunkul, K. Hantrakul, P. Pramokchon, P. Khoenkaw, MQTT-Topics Management System for sharing of Open Data, <https://doi.org/10.1109/ICDAMT.2017.7904935> (2017).
23. M. Singh, M. Rajan, V. Shivraj, P. Balamuralidhar, Secure MQTT for Internet of Things (IoT), <https://doi.org/10.1109/CSNT.2015.16> (2015).
24. S. Chakraborty, A. Majumder, 6LoWPAN Security: Classification, Analysis and Open Research Issues, <https://ssrn.com/abstract=3354367> (2019).
25. P. K. Kamma, C. R. Palla, U. R. Nelakuditi, R. S. Yarrabothu, Design and implementation of 6LoWPAN border router, <https://ieeexplore.ieee.org/document/7759025> (2016).
26. T. Jager, The Generic Composite Residuousity Problem, [https://link.springer.com/chapter/10.1007/978-3-8348-1990-1\\_5](https://link.springer.com/chapter/10.1007/978-3-8348-1990-1_5) (2012).
27. X. Fan, F. Susan, W. Long, S. Li, Security Analysis of Zigbee, <https://courses.csail.mit.edu/6.857/2017/project/17.pdf> (2017).
28. S. Gupta, R. Garg, N. Gupta, W. S. Alnumay, U. Ghosh, P. K. Sharma, Energy-efficient dynamic homomorphic security scheme for fog computing in IoT networks, <https://doi.org/10.1016/j.jisa.2021.102768> (2021).
29. G. Mitsis, P. A. Apostolopoulos, E. E. Tsiropoulou, S. Papavassiliou, Intelligent Dynamic Data Offloading in a Competitive Mobile Edge Computing Market, <https://www.mdpi.com/1999-5903/11/5/118> (2019).
30. A. Daeri, A. R. Zerek, M. A. Abuinjam, ElGamal public-key encryption, [http://ipco-co.com/PET\\_Journal/Papers%20CEIT%2014/025.pdf](http://ipco-co.com/PET_Journal/Papers%20CEIT%2014/025.pdf) (2014).
31. K. E. Makkaoui, A. Beni-Hssane, A. Ezzati, Cloud-ElGamal: An efficient homomorphic encryption scheme, <https://doi.org/10.1109/WINCOM.2016.7777192> (2016).
32. M. Yung, Y. Tsiounis, On the Security of ElGamal Based Encryption, <https://doi.org/10.1007/BFb0054019> (1998).
33. J. O. Blech, Proving the security of ElGamal encryption via indistinguishability logic, <https://doi.org/10.1145/1982185.1982527> (2011).
34. M. R. Baharon, Q. Shi, D. Llewellyn-Jones, A New Lightweight Homomorphic Encryption Scheme for Mobile Cloud Computing, <https://doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.88> (2015).
35. M. R. Baharon, Q. Shi, D. Llewellyn-Jones, A New Lightweight Homomorphic Encryption Scheme for Mobile Cloud Computing, <https://ieeexplore.ieee.org/document/7363129> (2015).
36. Çetin Kaya Koç, F. Özdemir, Z. Ödemiş Özger, Goldwasser-Micali Algorithm, [https://link.springer.com/chapter/10.1007/978-3-030-87629-6\\_4](https://link.springer.com/chapter/10.1007/978-3-030-87629-6_4) (2021).
37. D. E. Denning, Digital signatures with RSA and other public-key cryptosystems, <https://doi.org/10.1145/358027.358052> (1984).
38. L. Fousse, P. Lafourcade, M. Alnuaimi, Benaloh's Dense Probabilistic Encryption Revisited, <https://arxiv.org/abs/1008.2991> (2010).
39. A. ACAR, H. AKSU, A. S. ULUAGAC, M. CONT, A Survey on Homomorphic Encryption Schemes: Theory and Implementation, <https://doi.org/10.1145/3214303> (2018).
40. P. Ora, P. R. Pal, Data security and integrity in cloud computing based on RSA partial homomorphic and MD5 cryptography, <https://doi.org/10.1109/10.1109/IC4.2015.7375655> (2016).
41. G. J. Simmons, Symmetric and Asymmetric Encryption, <https://doi.org/10.1145/356789.356793> (1979).
42. I. Damgård, M. Jurik, A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System, <https://doi.org/10.5555/648118.746742> (2001).
43. O. Benamara, F. Merazka, A new distribution version of Boneh-Goh-Nissim cryptosystem: Security and performance analysis, <https://www.tandfonline.com>. <https://doi.org/10.1080/09720529.2020.1782570> (2020).

# Tool-Based Approach on Digital Vulnerability Management Hub (VMH) by Using TheHive Platform



V. Ceronmani Sharmila, M. Arvinth Sithartha, Samita Ramesh Babu, and M. Shruthi

## 1 Introduction

The rate of information security breaches has been drastically increasing throughout history. Record-breaking National Vulnerability Database (NVD) stated that “2021 has officially been a record-breaking year for vulnerabilities,” as 2021 became the year known for remote work. The amount of ransomware attacks had increased to peak, and an average of 50 Common Vulnerabilities and Exposures (CVEs) [1–3] had been logged each day, and 90% of these vulnerabilities discovered are so far can be exploited [4] with limited technical skill by any attackers. A solid percentage of 61 had discovered that there was no requirement such as clicking a link, downloading a file, or sharing any biometric or credentials needed for accessing/exploiting a user. Fifty-four percent of these vulnerabilities logged are classified as having a high chance for being easily accessible/exploitable by attackers in any means of time. As the era of work from home has normalized, security of a user system has become a prime focus of development.

Thousands of advanced vulnerabilities are being discovered causing industries/organizations/companies to manually reconfigure their security settings and are in need to constantly update the new patch in order to defend against newly discovered vulnerabilities throughout their network. These vulnerabilities are often identified as a part or one among top ten OWASP [5] identified vulnerabilities. These top ten are considered to be a security weakness verified by business organizations and security professionals and are ranked every year by the severity of the security risk these attacks pose to a platform.

---

V. C. Sharmila · M. A. Sithartha · S. R. Babu (✉) · M. Shruthi  
Department of Information Technology, Hindustan Institute of Technology and Science, Chennai, India  
e-mail: [csharmila@hindustanuniv.ac.in](mailto:csharmila@hindustanuniv.ac.in)

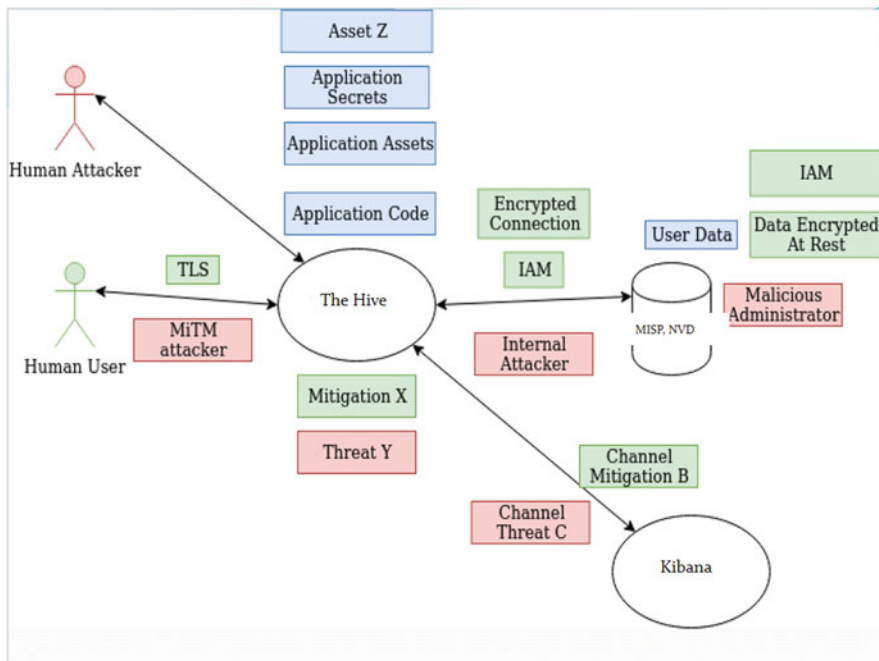


Fig. 1 UML diagram vulnerability management hub

Vulnerability management is generally said to be the process of identifying, grouping, prioritizing, and resolving in any platform as shown in Fig. 1. It has been a burden to perform security vulnerability [4] analysis and software updates on a daily basis for an organization. Every manual work consumes production time of the environment which causes a downtime. It has been a common method to manually update a patch provided by the package manager throughout the network system of an organization and not choosing an automatic method for the reason of time management of processes. In this research paper, we are bringing out a tool-based approach for analyzing and managing vulnerability based on their threat level [6], with the best mitigation [7] recommended by the tool by taking into consideration top OWASP-recognized attacks and the recent National Vulnerability Database (NVD) providing logs integrating along other open-source platforms such as the Hive which provides firewall, real-time vulnerability scanning, classification, and the best recommended mitigation. This collaboration enables us to communicate on, plan, execute, and manage workflow seamlessly with cost efficiency as in Fig. 2.

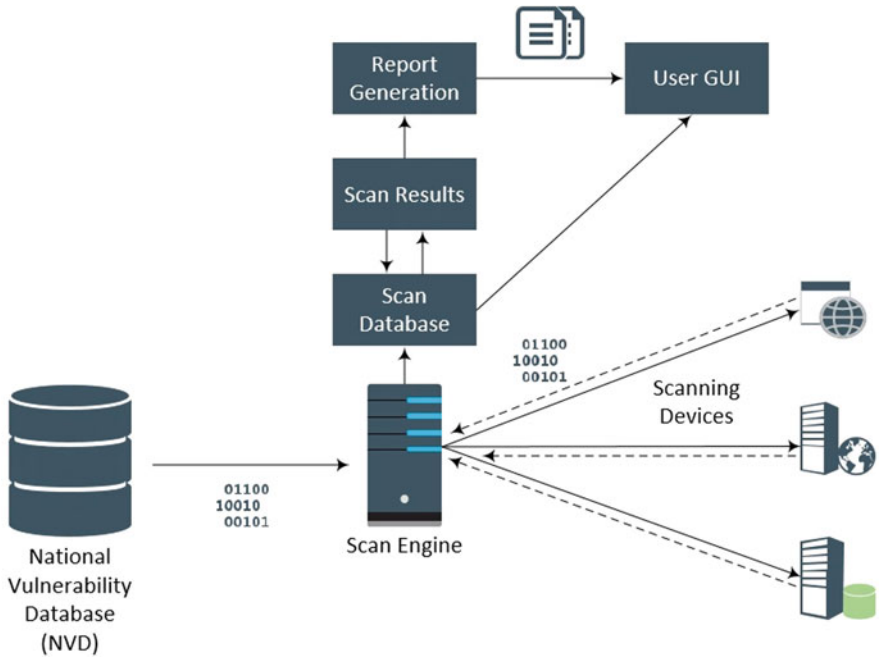


Fig. 2 System overview

## 2 Related Work

Traditional vulnerability assessment was done using the Common Vulnerability Scoring System (CVSS) [2]. It was explained how CVSS was used to calculate the vulnerability using the metrics. It is based on base score, temporal score, and environmental score known for its three main metrics. Base score is evaluated based on exploitability under which attack vector and complexity and User Interaction (UI) metrics are calculated as well. Scope and impact metrics are done under base score. Temporal score is for evaluating impact on CIA triad, i.e., confidentiality, integrity, and availability. Environmental score evaluates the modified base metric and CIA requirements.

A survey was conducted in [8] how CVSS gives an analytical approach based on 18 evaluation criteria. They have also tested the results suggesting that security modeling with CVSS data solely cannot accurately reflect system downtime. Nevertheless, the results show that metrics using additional CVSS datasets are strongly associated with downtime. As a result, models that use only the weak link to record metrics are less promising than those that consider all risks. Romalli Syed [5] proposed Cybersecurity Vulnerability Ontology (CVO) which provides knowledge of vulnerability domain, and Cyber Intelligence Alert (CIA) basically gives alert if any vulnerability occurs and suggests mitigations to mitigate them. The user has

to also measure the accuracy and performance of the CIA system. He integrated CVSS framework for analyzing the severity of vulnerabilities. This concept had few limitations such as manual ontology was not precise due to lack of knowledge, cognitive overload, and was biased.

Heterogeneous information network is a ranking method proposed by Wang et al. [2]. This method is used to assess the risk of exposure to a particular network. It takes into account the exploit of the vulnerability, the network components which have been impacted, and the value of the components at risk. Method used to calculate risk of each vulnerability and to use the measurement process, another model was also proposed, a model expansion approach, where it also considers additional factors of vulnerability and access. These methods are used to compare CVSS and graphical attack methods. In the future, they hope to develop an automated ontology. Roland et al [7] propose an assessment based on Strength and Vulnerability Intervention which is related to virtual threats. This method supplies multifactor analysis, identifies and constructs a digitalized threat scenario, and generates a report developing an intervention scenario. It is based on structural framework and decision-based analysis. In [3] we introduce and recognize Memory-Related Vulnerability Detection Approach using Vulnerability Features (MRVDAVF). It helps to analyze memory-related vulnerabilities like memory leak, double-free, and use-after-free and also two improved control flow graphs to detect some of the vulnerabilities. This paper has also proposed two algorithms, namely, vulnerability judging algorithm based on vulnerability feature and feature judging algorithm to detect memory-related vulnerabilities. However this method was limited to memory-related vulnerabilities

They have provided results based on four detection tools over three test cases displaying comparison studies of how MRVDAVF is feasible and accurate. Zhang et al. [3] have proposed a framework to calculate a risk assessment, automatically give analysis, and also collect information of real-time vulnerabilities from reports with deeper evaluation. George and Thampi [8] used Industrial Internet of Things IIoT-based framework for detecting and analyzing the smart connecting devices. This framework needs to improve the cost models to improve the efficiency in the network. A survey on vulnerabilities in cloud-based applications was conducted in [9]. They have analyzed various prospects collecting different databases including OWASP and analyzing the risk exposure and using vulnerability scanning tools to do comparative study and in [11] focusing on SQL injection detection using CNN framework which extracts the vulnerabilities in SQL automatically identifying attack traffic bypass and further study on multi-classification models. Qin et al.[10] proposed VulArcher which is a heuristic method for finding vulnerabilities in the android apps. Their main objective is to prevent the exploit use of APIs and provide an effective analysis method. Only drawback was that APIs were not updated. Vulnus was implemented for visual vulnerability analysis.

### 3 Approach on Framework

The design and operation of vulnerability management hub (VMH) software has been implemented within this framework with four modules: knowledge collector, asset collector, vulnerability collector, and processing module. The flow of the records has been indicated in Fig. 3. The primary three modules are for collecting information on various components of the community, and the fourth, for calculations, is liable for bringing results as close to actual time as possible.

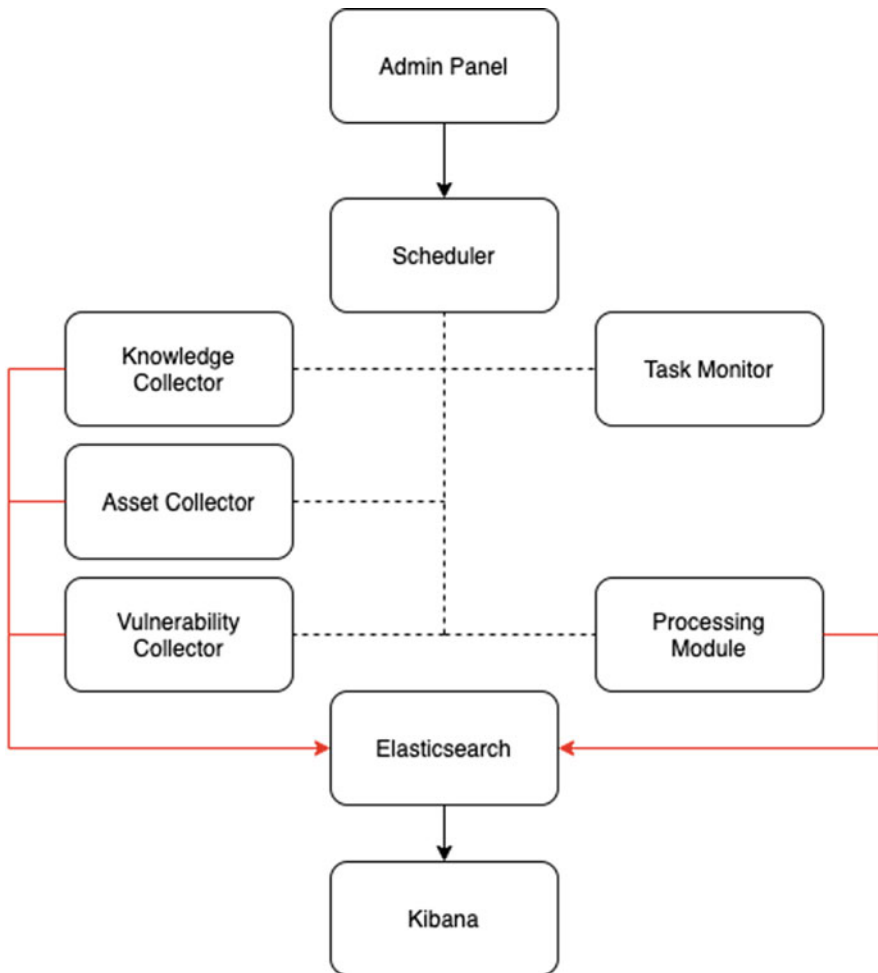


Fig. 3 Workflow framework

The above framework lets the modules work freely of one another, imparting asynchronously. Because of this, contingent upon how much information got by the module, it tends to be scaled upward, autonomously of different parts.

The scheduler module controls the request and timing of information download from every individual source, while the undertaking screen permits you to see the current framework status. The whole arrangement has been arranged for tasks in the cloud-based environment and depends on the Docker container technology. The information is saved in Elasticsearch, which authorizes its handling in a full-text mode, while the Kibana software is utilized for the outcomes.

Flexibility has been taken into consideration and solved by using Python language to implement the components, and the chance of efficient information handling on the server side. The whole project incorporates support for occupants and takes into account full information division between specific occupants. Vulnerability management hub (VMH) works in functional and authentic modes. The working mode permits the client to see all changes logged by the framework in real time, while the functional mode authorizes review and performs computational procedures on authentic information, in this way working on the quality and extent of the examination of events in the framework.

To work on the guideline of activity and the vulnerability management hub architecture, the information handling process has been isolated into individual stages following each other.

### ***3.1 Knowledge Collector***

The Knowledge collector module is responsible for collecting information, statistics, updating and downloading information that are publicly available databases such as National Vulnerability Database (NVD), Exploit Database and MISP Database. It analyzes the structure of the document or URL. It tries to collect vectors to assess the speed, scalability, and exploitation. It gathers all the information required for vulnerability assessment. The information gathered is based on a software known as Common Platform Enumeration (CPE). After passing the familiarization process, the information that is collected can be displayed in the presentation layer.

### ***3.2 Asset Collector***

The asset collector module is responsible for coping with information on detected assets and belongings defined for the monitored network. It has information assets. The primary supply is the Ralph device, a system that meets the necessities of corpo-

rate networks making an allowance for the management of the life cycle of a selected element of the environment. The second source of statistics is a vulnerability scanner which, in addition to scanning, has the functionality of detecting IT infrastructure components. In this way, VMH is in a position to tell the operator approximately the discrepancy between the information contained in the asset database and the statistics provided with the experiment consequences. The list underneath suggests how the asset is represented within the database. The confidentiality requirement, integrity requirement, and availability requirement fields permit you to calculate the environmental danger assessment for one's property. The enterprise owner and technical proprietor organize organizational devices accountable for precise assets, both from a technical and commercial enterprise angle. If any of the above-noted values is missing, an alarm will be generated within the system, informing the operator that the data is inconsistent.

### ***3.3 Vulnerability Collector***

The vulnerability collector module is responsible for downloading, updating, and normalizing records from Nessus and OpenVas vulnerability scanners. So that you can optimize the results, throughout statistics processing, vulnerabilities categorized as informational are neglected. The list under indicates the structure of the file of the detected vulnerability, after being processed through the module. This structure includes facts which include the port number, protocol, carrier name, vulnerability description, and statistics on how the vulnerability has to be constant. This fact is advisable for machine owners and other people implementing the corrective mechanisms.

### ***3.4 Processing Module***

The processing module is liable for making calculations and updating the evaluation of detected vulnerabilities, taking into consideration statistics obtained from previous modules. For every new or updated vulnerability, the acquired calculation consequences are entered in the fields environmental score v2 for CVSS 2.0 and environmental score v3 for CVSS 3.0. Inside the fields, environmental rating vector v2 and environmental rating vector v3 there is a vector notation of the vulnerability evaluation, way to which it is far feasible to without problems examine every of the person values affecting the very last environmental [12] assessment.



## 4 Methodology

### 4.1 Integration with the Hive

The Hive can be defined as an open security incident response platform (SIRP). In collaboration with the Hive software, the Cortex module is conveyed, which permits you to automate a portion of the work of analysts. Vulnerability management hub (VMH) utilizing the ElastAlert module enables you to make extra security events. Then again, an extra responder permits you to enhance the data of different occasions with data contained in the vulnerability management hub (VMH) and the coordinated asset information base.

#### Webhook Configuration of the Hive

The Hive can inform the outside system of occasions happening in the framework, for example, making an alert or an errand. Vulnerability management hub (VMH) utilizes the Hive webhooks to deal with alerts of the vulnerability and to consequently make tasks and cases from them to work with, built by the admin, and to follow up with the vulnerability management process. Also, activating the Hive webhook's usefulness permits you to recover data from the Hive webhook logs and imprint them in the label fields of weakness reports. This permits, in addition to other things, to deal with pointers mentioned in [13], proposed by Jinchang Hu and Jinfu Chen, and integrated to the vulnerability mitigation. This module uses a method to retrieve all the information about the vulnerability. It also retrieves the ones that are deleted and not processed.

To allow the integration with vulnerability management hub, a configuration is added in the form of input as in Fig. 4.

#### Configuration

##### Directory

/etc/thehive/application.conf:

##### Input

```
webhooks
{
  myLocalWebHook
  {
    url = "http://my_HTTP_endpoint/webhook"
  }
}
```

##### Command

```
read -p 'Enter URL of TheHive: '
```

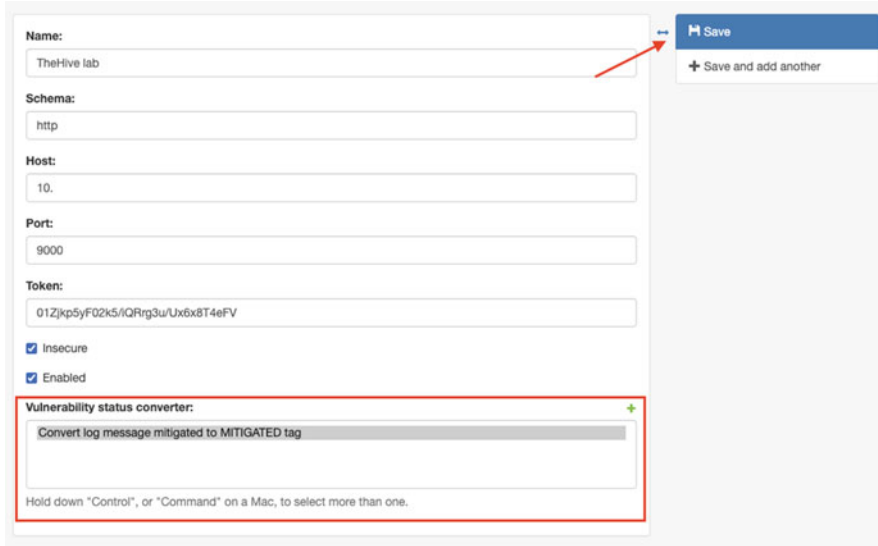


Fig. 4 Configuration of webhook

```

thehive_url
read -p 'Enter login ID: ' thehive_user
read -s -p 'Enter password: ' thehive_password

curl -XPUT -u$thehive_user:$thehive_password -H 'Content-type: application/json' $thehive_url/api/config/organisation/notification -d '
{
  "value": [
    {
      "delegate": false,
      "trigger": { "name": "AnyEvent"},
      "notifier": { "name": "webhook", "endpoint": "vmh" }
    }
  ]
}'

```

### ElastAlert Configuration

ElastAlert is an alert framework used for detecting anomalies or various kinds of patterns in data. This works by combining Elasticsearch with two types of components: rule types and alerts. Elasticsearch is queried on a regular basis and the

data is passed to a rule type that determines when a match is found. When a match occurs as like vulnerability metrics in actual attacks mentioned in [14] proposed by Hannes Holm, it is propagated to one or more alerts and the action is taken based on the match. ElastAlert saves its status in Elasticsearch, and when it starts, it resumes where it left off. If Elasticsearch does not respond, ElastAlert waits for recovery before continuing alerts that trigger errors that can automatically repeat for a specified period of time. The rule type is in charge of processing the data returned by Elasticsearch.

It is initialized with the rule configuration and passes the data returned by the query from Elasticsearch with the rule filter and returns a match based on that data. It creates alerts if a match is found. Matches are usually dictionaries that contain values from Elasticsearch documentation, but can contain any data added by the rule type.

ElastAlert is an irregularity cautioning system that chips away at the premise of information designs contained in Elasticsearch. The bundle is installed with the following command:

### **Configuration**

#### **Installation**

```
pip3 install elastalert
```

To run the ElastAlert as a service in the Linux background, the required configuration is created.

#### **Directory**

```
/usr/lib/systemd/system/elastalert.service
```

#### **Input**

```
[Unit]
```

```
Description=Elastalert
```

```
After=elasticsearch.service
```

```
[Service]
```

```
Type=simple
```

```
Restart=on-failure
```

```
ExecStart=/usr/local/bin/elastalert --config /home/elastalert/config.yaml
```

```
[Install]
```

```
WantedBy=multi-user.target
```

#### **Command**

```
systemctl enable elastalert
```

```
systemctl start elastalert
```

## 5 Simulations and Analysis

VMH is integrated using OpenVas and Nessus vulnerability scanners. Adding a scanner is done in the Scanners/Configs table by pressing the ADD CONFIG button in the top-right corner as shown in Fig. 3.

The connection to the Nessus scanner is possible using the available REST API, while the connection to the OpenVas scanner is possible using the OMP and GVM agreements, which require additional settings on the controller side. Go to Scanners tab, click Config tab, and here you will be able to enable or disable or even add a scanner configuration.

While using ADD CONFIG you will need to put necessary details such as name, host, port number, filter, scanner, username, and password as in Fig. 5. After filling in all the details, save the configuration and you will get a message saying config was added successfully as shown in Fig. 6.

VMH software also includes an IT resource management website called Ralph. Add-ons are done in a separate menu after pressing the ASSET MANAGEMENT button -> Configs. In the Configs panel, click the + button in the top-right corner

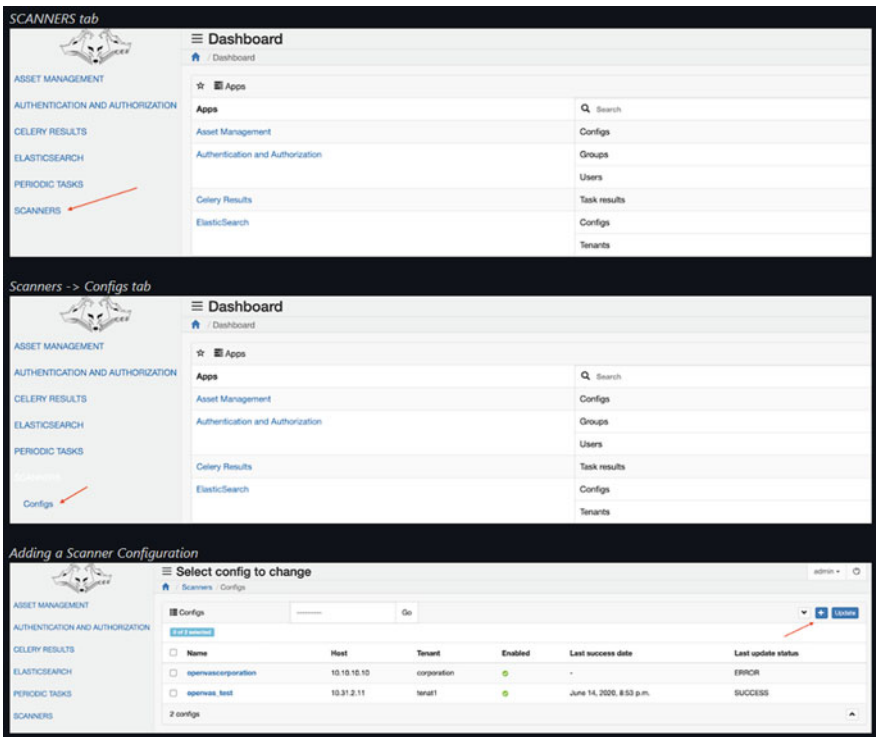


Fig. 5 Adding scanners

<input type="checkbox"/>	Name	Host	Tenant	Enabled
<input type="checkbox"/>	openvascorporation	10.10.10.10	newtenant1	<span style="color: red;">●</span>
<input type="checkbox"/>	openvas_test_supernew	10.31.2.11	tenant1openvas	<span style="color: green;">●</span>

**Fig. 6** Enabling scanners



**Fig. 7** Ralph configuration message

and enter the correct configuration of the resource base. After completing all the fields, save. To stop click the Save button on the left side of the form. If all data is correct, a message will appear as shown in Fig. 7.

Configuration was added successfully. To load data on the well-added resources website, go to ASSET ADMINISTRATION -> Configs tab, select the checkbox next to the previously added site to which we want to import data, and then select Import selected settings from drop-down menu and click the Go button. Once imported it will send the message “start import.” If the system is already importing data or performing risk statistics, and it will not be possible to start a new import, the message will appear in the same location and information.

VMH has a task scheduler in which the system administrator can define a specific time or duration of tasks such as updating data from vulnerability scanners or an asset database.

Setting the time interval or specific date and time of the specified task is performed in the Periodic Tasks table. For example, adding a time interval to the Periodic Tasks table -> Periods.

After setting the correct time and defining the task time, you can begin to define a specific schedule in the Periodic Tasks table. After pressing the ADD PERIODIC TASK button in the top-right corner, it is necessary to complete the required data.

The results of the planned activities described in the Periodic Tasks table are presented in the table of results for a vegetarian crop/activity results. Figure 8 contains the activity ID, name, action date, and status.

The Hive creates a task and tracks the management process providing vulnerability alert in case of identification. Logs and information are retrieved by this method which ultimately helps to mitigate the vulnerability using these factors. After mitigating it produces a mitigation report as shown in Fig. 9, which can be later used for evaluation or future records.

Select task result to change

Home / Celery Results / Task results

Search "Task results" [ ] [Go]

0 of 190 selected

📅 2020 / June 28 / June 29

<input type="checkbox"/> Task id	Task name
<input type="checkbox"/> ba5a4b74-63b1-4a68-b655-48f7963dcddf	vmc.ralph.tasks._update_assets
<input type="checkbox"/> bc80cdc5-5187-46f3-9640-d85584134a55	vmc.common.tasks.__release_lock
<input type="checkbox"/> eabeb8e5-fcf9-4e8c-8ea8-632e83b34735	vmc.scanners.tasks._update_scans
<input type="checkbox"/> 073b580a-8a81-49d4-bd20-dffcee88d9ee	vmc.common.tasks.__release_lock
<input type="checkbox"/> 5b8f5760-064d-4068-b729-a53ee8b183f6	vmc.scanners.tasks._update_scans
<input type="checkbox"/> 3ad82e31-44dc-44fa-bf66-6d8395edcb24	vmc.common.tasks.__release_lock
<input type="checkbox"/> ec4ebf2b-bd94-4ccc-b3a1-348b68936722	vmc.scanners.tasks._update_scans
<input type="checkbox"/> f1ec05f1-58e4-43d8-b53f-514db77767db	vmc.scanners.tasks._update_scans
<input type="checkbox"/> 0ec7070d-b2e8-42c8-9601-263a4e8006f2	vmc.common.tasks.__release_lock

Fig. 8 Periodic tasks

New Save Open Share Inspect

tags: "MITIGATED"

+ Add filter

\*.hook70.vulnerability 1 hit

Search field names

Filter by type 0

Selected fields: \_source

Available fields: \_id, \_index

```

_source
> tags: MITIGATED id: 3e3e842b-6dae-31af-ab53-8de4a17ee4b1 port: 80 protocol: tcp in
(Windows) description: Installed version: 7.1.29 Fixed version: 7.1.32 Installation
7.1.32, 7.3.9 or later. cve.id: CVE-2019-13224 cve.base_score_v2: 7.5 cve.base_scor
onig_new_deluxe() in regex.c in Oniguruma 6.9.2 allows attackers to potentially caus
possibly code execution by providing a crafted regular expression. The attacker provi

```

Fig. 9 Mitigation report

## 6 Conclusion and Future Works

As organizations lack confidence in their information security and asset security, this proposed approach accepts that security is one of the major parts of a sound business today and tends to set a path to solve this issue without giving up on productivity of the organization. We know very well that a significant number of the information security solutions that are available are extravagant and only one out of

every odd organization can bear the cost of them, which is the reason this approach is an attempt to provide clients of an organization with the best conceivable worth in network and information security at the most minimal conceivable expense. Hence the reason why this paper came out with the tool-based management system – VMH using the Hive platform. This method is accurate up to more than 75%, while the traditional scanners resulted with accuracy of 62%. This approach focuses on risk management process which provides the best information security using the Hive as base framework and integrating with open-source datasets and security tools to witness a virtual presentation of threats while categorizing their threat levels and trying to provide the best mitigation to overcome such threat to the respective organization.

As a long-term goal for this approach with the view of a better supply chain, the next big enhancement would be providing an accurate automated prioritization with better remediating and reporting solutions so that organizations can look forward to a bright and secure future with being one step ahead of the hackers while managing information security risk factors without giving up on the organization's productivity or performance.

## References

1. Tianlei Zang; Shibin Gao; Tao Huang; Xiaoguang Wei; Tao Wang, 2019, "Complex Network-Based Transmission Network Vulnerability Assessment Using Adjacent Graphs", IEEE Systems Journal, vol. 14, no. 1, pp. 572–581 <https://ieeexplore.ieee.org/document/8809927>
2. Peter Mell; Karen Scarfone; Sasha Romanosky, 2006, "Common Vulnerability Scoring System", IEEE Security & Privacy, vol. 4, no. 6, pp. 85–89. <https://ieeexplore.ieee.org/document/4042667>
3. Xiong Zhang; Haoran Xie; Hao Yang; Hongkai Shao; Minghao Zhu, 2020, "A General Framework to Understand Vulnerabilities in Information Systems", IEEE Access, vol. 8, pp. 121858–121873. <https://ieeexplore.ieee.org/document/9130665>
4. Anatoliy Gorbenko; Alexander Romanovsky; Olga Tarasyuk; Oleksandr Biloborodov, 2020, "From Analyzing Operating System Vulnerabilities to Designing Multiversion Intrusion – Tolerant Architectures", IEEE Transactions on Reliability, vol. 69, no. 1, pp. 22–39. <https://ieeexplore.ieee.org/document/8662611>
5. Xin Xie; Chunhui Ren; Yusheng Fu; Jie Xu; Jinhong Guo, 2019, "SQL Injection Detection for Web Applications Based on Elastic-Pooling CNN", IEEE Access, vol. 7, pp. 151475–151481. <https://ieeexplore.ieee.org/document/8877739>
6. Grzegorz Siewruk; Wojciech Mazurczyk, 2021, "Context-Aware Software Vulnerability Classification Using Machine Learning", IEEE Access, vol. 9, pp 88852–88867. <https://ieeexplore.ieee.org/document/9411853>
7. Segundo Moisés Toapanta; Omar Alexander Escalante Quimis; Luis Enrique Mafla Gallegos; Ma Roció Maciel Arellano, 2020, "Analysis for the Evaluation and Security Management of a Database in a Public Organization to Mitigate Cyber Attacks", IEEE Access, vol. 8, pp. 169367–169384. <https://ieeexplore.ieee.org/document/9193956>
8. Gemini George; Sabu M. Thampi, 2018, "A Graph-Based Security Framework for Securing Industrial IoT Networks from Vulnerability Exploitations", IEEE Access, vol. 6, pp. 43586–43601. <https://ieeexplore.ieee.org/document/8430731/>

9. Kyriakos Kritikos; Kostas Magoutis; Mano Papoutsakis; Sotiris Ioannidis, 2019, “A survey on vulnerability assessment tools and databases for cloud-based web applications”, *ScienceDirect, Array*, vol. 3–4, 100011. <https://doi.org/10.1016/j.array.2019.100011>
10. Jiawei Qin; Hua Zhang; Jing Guo; Senmiao Wang; Qiaoyan Wen; Yijie Shi, 2020, “Vulnerability Detection on Android Apps–Inspired by Case Study on Vulnerability Related With Web Functions”, *IEEE Access*, vol. 8, pp. 106437 – 106451. <https://ieeexplore.ieee.org/document/9102313>
11. Romilla Syed, 2020, “Cybersecurity vulnerability management: A conceptual ontology and cyber intelligence alert system”, *Science Direct Information & Management*, vol. 57, no. 6, 103334. <https://doi.org/10.1016/j.im.2020.103334>
12. Wenrui Wang; Fan Shi; Min Zhang; Chengxi Xu; Jinghua Zheng, 2020, “A Vulnerability Risk Assessment Method Based on Heterogeneous Information Network”, *IEEE Access*, vol. 8, pp. 148315–148330. <https://ieeexplore.ieee.org/abstract/document/9163374>
13. Jinchang Hu, Jinfu Chen, Lin Zhang, Yisong Liu, Qihao Bao, Hilary Ackah-Arthur, Chi Zhang, 2020, “A memory-related vulnerability detection approach based on vulnerability features”, *IEEE TUP Tsinghua Science and Technology*, vol. 25, no. 5, pp. 604–613. <https://ieeexplore.ieee.org/document/9036137>
14. Hannes Holm; Mathias Ekstedt; Dennis Andersson; 2012, “Empirical Analysis of System-Level Vulnerability Metrics through Actual Attacks”, *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 6, pp. 825–837. <https://ieeexplore.ieee.org/document/6259801>



# Performance Analyzer for Blue Chip Companies



Ishita Badole, Sakshi Chheda, Ojasa Chitre, and Dhananjay Kalbande

## 1 Introduction

A blue chip company is one considered to be leading in its sector and produces dominant goods or services. These companies are known for their ability to withstand economic turbulence due to their large market capitalization, consistent revenues, and sustained growth over time. Some examples are HDFC Bank, Infosys, ITC Limited. Individuals avoid investing in the stock market as such investments are considered risky. Past and current events relating to the company, stock price trends, financial results, and culture highly influence the reputation and thus the market performance of company stocks. In-depth analysis of its reputation and stock performance can help investors feel confident about trading blue chip stocks. In the proposed system, news articles of selected blue chip companies are scraped, and then sentiment analysis is performed on this textual data to produce sentiment scores. Historical stock data and generated sentiment scores are used for future price prediction. Pattern detection is performed on combined predicted stock price, sentiment scores, and historical stock data to produce advisory reports as the final output. Section 1 gives an introduction to the study, Sect. 2 discusses the motivation, Sect. 3 is the literature survey, Sect. 4 explains the proposed methodology, Sect. 5 discusses the results, and Sect. 6 provides the conclusion.

---

I. Badole (✉) · S. Chheda · O. Chitre · D. Kalbande  
Sardar Patel Institute of Technology, Mumbai, Maharashtra, India  
e-mail: [ishita.badole@spit.ac.in](mailto:ishita.badole@spit.ac.in); [sakshi.chheda@spit.ac.in](mailto:sakshi.chheda@spit.ac.in); [ojasa.chitre@spit.ac.in](mailto:ojasa.chitre@spit.ac.in);  
[drkalbande@spit.ac.in](mailto:drkalbande@spit.ac.in)

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023  
R. Misra et al. (eds.), *Advances in Data Science and Artificial Intelligence*,  
Springer Proceedings in Mathematics & Statistics 403,  
[https://doi.org/10.1007/978-3-031-16178-0\\_14](https://doi.org/10.1007/978-3-031-16178-0_14)

## 2 Motivation

Investors need to analyze various companies' stock performance and stay updated on the latest news events before making an investment decision. This can be laborious and even intimidating for new investors. The motivation of this study is to mitigate the time and effort required to make investment decisions.

## 3 Literature Survey

Saloni et al. [1] predicted the stock prices for S&P 500 companies using time series models such as ARIMA, Facebook Prophet, neural networks like Recurrent Neural Network (RNN) Long Short-Term Memory (LSTM) architecture, and combination of neural networks along with financial news articles. Their study shows RNN gives better results, and there exists a correlation between stock price movement and textual information. Naive Bayes Classifier was used by Sneh and Jay [2] to categorize the news text having negative or positive sentiment. They used four machine learning techniques: K-Nearest Neighbors, Support Vector Machines, Naive Bayes, and neural network for the stock price prediction. News data of only a few banks were considered. K-Nearest Neighbors performed best for their bank dataset.

Elmasry and Mohamed Abbas [3] propose an implementation that comprises three phases: the first phase includes detection and filtering of false news, second includes sentiment analysis by finding polarity scores, and third phase is predicting the trends of stocks using decision forest, boosted decision tree, and linear regression model. They considered data from Amazon, Google, and ExxonMobil. Linear regression achieved the best results for the considered dataset. Shah et al. [4] created a dictionary containing words and phrases pertaining to the pharmaceutical sector along with corresponding sentiment polarity strength. For each news article, score was calculated by comparing to the dictionary and cross-validated against stock prices. Decision to buy, hold, or sell was recommended by comparing the score to a predefined threshold. The authors achieved 70.59% directional accuracy, but analysis was done for only 6 months of news and the dictionary contained only 100 words.

Esichaikul and Phumdontree [5] used the United CNN-Bidirectional GRU model on data from Thai financial news websites and @ThaiValueInvest Twitter account. A web application was developed with functions to view daily news, export news, and provide sentiment indicators. The output was sentiment summary grouped by news category and by news source. Sentiment was not analyzed for individual companies. Samad et al. [6] performed two types of analysis – textual analysis and numerical analysis. Textual data was news articles, and numerical data was historic stock price data. Support Vector Machine (SVM) algorithm was applied on textual data to classify articles as positive or negative. Random Forest algorithm was used for price

prediction using the numerical data. Price trend analysis was carried out using actual and predicted prices. A dashboard was developed that showed article sentiment and prediction price. Random Forest algorithm and SVM algorithm gave 99% and 68% accuracy, respectively. Although the accuracy achieved was high, the method might not generalize as only five Malaysian companies were chosen for the study.

Agarwal et al. [7] generate investing insight by applying sentiment analysis using VADER (Valence Aware Dictionary and Sentiment Reasoner) tool. A compound score is generated based on the normalized aggregate of the sentiment. They do not use livestreamed data. Their sentiment analysis method needs removal of misleading sentences because it does not consider the context of the word. Kim et al. [8] have used a Long Short-Term Memory (LSTM) to create a sentiment dictionary. The dictionary is used to get a positive index of news articles. The dataset used is Korean. They have only given predictions for the next day and not the trend for a stock. They were able to get a 0.3034 correlation. Jariwala et al. [9] have observed that the K-means model performs poorer than the Naive Bayes and SVM. They have only used news headlines which has limited their accuracy. They have calculated the accuracy for headlines of Mahindra and Mahindra – 64.73% – and accuracy of 57.97% for Kotak Mahindra Bank Ltd.

## 4 Proposed Methodology

The proposed system is a technical analysis tool for traders, investors, educators, and market enthusiasts that provide visualizations and advisory reports on complex data that were once only comprehensible to professional traders. Figure 1 represents the proposed methodology of performance analyzer for blue chip companies. News articles are extracted using web scraping and filtered to retain articles on blue chip companies. Actual article text is considered instead of news headlines as headlines can be sensationalized. The sentiment analysis model and price prediction model are chosen based on performance comparison of models. Sentiment analysis is performed on the extracted text to generate the sentiment score for the article. Future price prediction is performed on integrated data that contains historic prices and sentiment scores. Pattern detection of candlestick charts is performed. The visualizations are provided as interactive charts which include candlestick charts with detected patterns, OHLC values, corresponding news headlines, and various technical indicators. These visualizations will be included in the advisory reports as the output of the system.

### 4.1 Data Collection and Filtering

For historical data, yfinance Python package is used to fetch daily market data of open, high, low, close, as well as daily traded volume from Yahoo Finance API

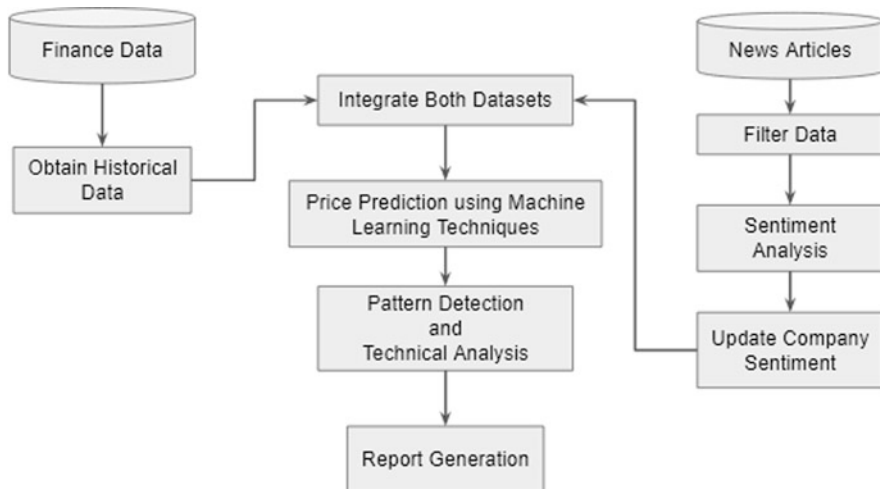


Fig. 1 Proposed methodology

Table 1 Selected blue chip companies

Sector	Companies
Financial services	Axis Bank, HDFC Bank, ICICI Bank, IndusInd Bank
Automobile	Bajaj Auto, Maruti Suzuki India
IT	HCL Technologies, Infosys, Tata Consultancy Services, Wipro
Oil and gas	Oil and Natural Gas Corporation, Bharat Petroleum Corporation, Indian Oil Corporation
Consumer goods	Asian Paints Ltd., Hindustan Unilever Ltd.
Telecom	Bharti Airtel Ltd.

(May 2019–March 2022). We have used BeautifulSoup for scraping news articles from trusted websites (up to 2015). The proposed methodology uses news articles from trusted sources to prevent fake news present on social media. Table 1 shows a list of companies we have considered. We have created a custom function to filter the articles that have been scrapped in order to only include the ones that are pertaining to a particular company. We have considered all possible ways a company may be mentioned in articles, e.g., DMart can also be referred to as Avenue Supermarts.

## 4.2 Sentiment Analysis

**Choosing the Model** We experimented with two pre-trained NLP models – VADER and FinBert – for sentiment analysis of financial text. Since our data is

unlabeled, publicly available labeled datasets in the financial domain were used to compare the performance of the models.

VADER is a lexicon and rule-based sentiment analysis tool that is specifically tailored for social media text [10]. It gives positive, negative, neutral, and compound scores. The compound score is a weighted score which is normalized between the range of  $-1$  and  $+1$ . FinBert is a pre-trained model built by further training the BERT language model for financial NLP tasks. It gives positive, negative, and neutral sentiment scores. The compound score is calculated as negative score subtracted from positive score. Its range is from  $-1$  to  $+1$ .

The datasets selected were Financial PhraseBank by Malo et al. [11] and gold commodity dataset by Sinha et al. [12]. Financial PhraseBank was also used on FinBert by its authors. The dataset has been manually annotated by 16 people with adequate background knowledge on financial markets. It consists of sentences from financial news categorized as positive, negative, or neutral based on the perspective of an investor. The categories are given as integer labels. We experimented with the subset of 100% annotator agreement. It consists of 2264 sentences.

Gold commodity dataset consists of 11,412 news headlines from 2000 to 2021 on the gold commodity. It was manually annotated by 3 subject experts. The headlines are categorized by sentiments as positive, negative, neutral, and none. The commodity news uses financial terms which are used in the stock market news. The headlines having sentiment “none” were removed. The given categories were converted to corresponding integer labels. Negative, neutral, and positive sentiments correspond to integer labels 0, 1, and 2 respectively. The models were applied on the datasets. Figure 2 shows a table consisting of sentences in the gold commodity dataset, and their actual sentiment is given in the price sentiment column. The compound scores generated were converted to integer label using the formula:

$$pred\_label = round(compound\_score + 1) \tag{1}$$

sentence	Price Sentiment	label	Finbert POS	Finbert NEG	Finbert NEU	Finbert Compound	Finbert Label
Gold futures end higher after 6 consecutive se...	positive	2	0.830826	0.103826	0.065347	0.727000	2
Gold prices mostly steady in Asia as investors...	neutral	1	0.302100	0.466713	0.231188	-0.164613	1
merrill sees gold trade in \$370-390 range in n...	neutral	1	0.254425	0.028172	0.717403	0.226254	1
gold rate today: gold, silver edge higher in m...	positive	2	0.936478	0.032290	0.031232	0.904188	2
gold, silver slip in morning trade	negative	0	0.026613	0.930240	0.043147	-0.903628	0

Fig. 2 FinBert applied on gold dataset

**Table 2** Accuracy result for models on Financial PhraseBank and gold commodity dataset

Model	Financial PhraseBank	Gold commodity dataset
VADER	0.63	0.15
FinBert	0.97	0.71

The label column contains the integer label corresponding to the price sentiment for that sentence. The FinBert sentiment scores and the predicted labels calculated from the compound scores using the Eq. (1) are also given. Table 2 shows the comparison of the two models on the datasets. The results of the comparison are discussed in detail in Sect. 5. FinBert performed better than VADER and was selected for the proposed system.

**Applying FinBert on Filtered Data** The FinBert model was downloaded from the Hugging Face model hub using the Transformers python library. For each article, the article text was tokenized such that it was truncated and padded to suitable length for FinBert using tokenizer. Sentiment scores were generated for positive, negative, and neutral labels. The FinBert compound score was calculated as negative score subtracted from positive score. The positive, negative, neutral, and compound sentiment scores are used in the price prediction.

### 4.3 Combining Results of Models

After filtering the data we noticed that all the dates did not have corresponding articles. In order to maintain continuity of data, we imputed the sentiment. In case of more than one article, we have aggregated using a mean. In case we have an absence of values, we have imputed with the previous value. When we have a new value of sentiment from articles, we need to also consider the fact that the previous sentiment also affects the overall sentiment. We have then merged this with the OHLC data extracted using yfinance. The formula we have used is:

$$UpdatedSentiScore = (0.4 * OldSentiScore) + (0.6 * NewSentiScore) \quad (2)$$

### 4.4 Stock Price Prediction

**Choosing the Model** We experimented with two prediction models – Vector Auto Regression (VAR) and Long Short-Term Memory (LSTM) networks. In the VAR model, each variable is a linear function of not only its past values but also the past values of all the other variables. On applying VAR model over HDFC Bank’s historical data we obtain a Mean Absolute Percentage Error (MAPE) of 10.02% for prediction of close price. Long Short-Term Memory (LSTM) has a chain-

like structure with repeating blocks called LSTM cell, and data is passed through the network with cycles. So the LSTM model takes into account previous inputs in addition to the current input. On applying LSTM model over HDFC Bank's historical data, we obtain a Mean Absolute Percentage Error (MAPE) of 2.27% for prediction of close price. For our dataset LSTM model outperformed the ARIMA model; thus, LSTM was selected to perform stock price prediction for the proposed system.

**Applying LSTM on Integrated Data** For predicting stock price, we selected six features from the integrated dataset – open, high, low, close price, volume, and FinBert compound score. These features are scaled to obtain the data values in a standard format. We generate a training and test dataset using the sliding window algorithm with a window size of 50 days. Our LSTM network has four layers. The first layer of the network has 300 neurons which is equal to the size of our input dataset, and each batch of input data consists of a matrix with 50 steps and six features. So the first layer accepts these batches as input and returns the entire sequence. Second layer of network inputs the previous layer's returned sequence and returns five values. Third layer is a dense layer with five neurons which inputs the five values returned from previous layer, and fourth LSTM layer is the final dense layer that outputs the predicted stock price. After training this LSTM model, we reverse the scaling for predicted values and evaluate model performance. The evaluation metric used is Mean Absolute Percentage Error (MAPE). We predict all the four stock prices – open, high, low, and close values for the next 60 days using our trained model. For HDFC Bank's integrated dataset, the LSTM model gives good result, MAPE of 1.95%. This implies that the mean of our predicted stock prices deviates from actual stock prices by only 1.95%.

## 4.5 *Pattern Detection and Analysis*

**Pattern Detection** In our study we have included 56 patterns. For each company, the candlestick patterns were extracted from the historic OHLC values. TA-Lib creates individual columns for each pattern. If the column for a pattern contains 0 for a candle, it means no pattern was extracted, positive value in the column represents bullish pattern, and negative value represents bearish pattern for that candle.

The overall pattern for each candle was specified in a new column in the dataset called `candlestick_pattern`. If no pattern was found, then "NO\_PATTERN" was filled in the cell for that candle. If a single pattern was extracted, then the corresponding pattern name was filled with a bear or bull tag based on its value. "Overall performance rank" from [patternsites.com](http://patternsites.com) was used to create `candlestick_rankings` dictionary in which the keys were the patterns (with bull and bear versions) and their corresponding values were their rankings. For the case where multiple patterns



Fig. 3 Candlestick patterns' chart for HDFC Bank



Fig. 4 Bollinger Bands and volume chart for HDFC Bank

were found for a given candle, this dictionary was used to select the best ranking pattern.

**Analysis Using Technical Indicators** An interactive candlestick chart is generated for each company for its historic prices. The OHLC values, pattern name, and recent news for a candlestick are shown when the cursor is hovered over the candlestick. Figure 3 shows the candlestick patterns' chart for HDFC Bank.

For each company, a plot with two subplots is generated using the historical data. The first subplot shows candlesticks as described for the previous plot, along with Bollinger Bands. Bollinger Bands are two standard deviation bands, above and below the simple moving average of the price. They indicate the volatility of the stock and show underbought or overbought conditions. The second subplot shows the volume of shares traded. It can be used to determine the overall market sentiment and identify trends. Figure 4 shows the Bollinger Bands and volume chart for HDFC Bank.

Another type of plot is generated for each company. This plot uses both historic and predicted prices. A vertical dashed line divides the plot into historic and





Fig. 5 MACD and RSI chart for HDFC Bank

predicted sections. The plot consists of two subplots. The first subplot shows Moving Average Convergence Divergence (MACD) indicator with MACD signal and histogram. Signal line crossovers can be observed, and histogram can show potential changes in market momentum. The second subplot shows the RSI index. Usually, RSI value above 70 and below 30 is considered as overbought and oversold condition, respectively. Figure 5 shows the MACD and RSI chart for HDFC Bank. These visualizations will be included in the advisory report along with commentary.

## 5 Results and Discussion

### 5.1 Sentiment Analysis Result

Table 2 shows the comparison of the two sentiment analysis models on the chosen labeled datasets. The evaluation metric used was accuracy score.

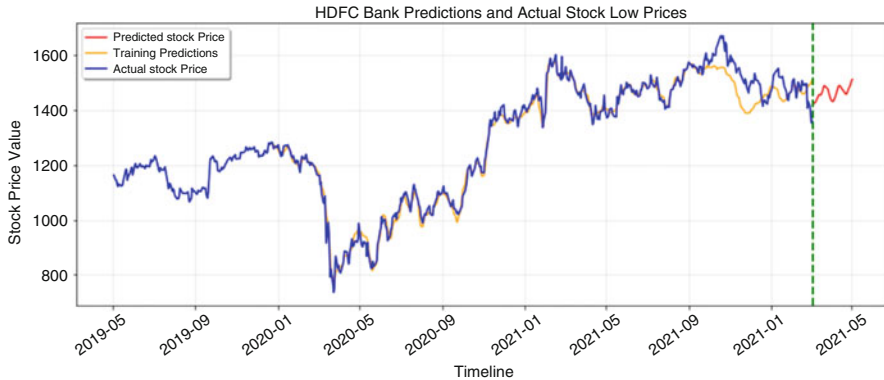
FinBert gave better accuracy than VADER on Financial PhraseBank. But, this may be because FinBert was trained on the same dataset. FinBert also performs better on the gold commodity dataset which was an unseen dataset for the model. So, we can infer that the FinBert model gives more accurate results on financial data compared to VADER.

### 5.2 Price Prediction Result

Table 3 shows results of the LSTM price prediction model. The Mean Absolute Percentage Error (MAPE) values for one company in each sector is shown. For every selected blue chip company, the open, high, low, and close prices are predicted thus producing four MAPE values. The last column of Table 3 represents the average of

**Table 3** Stock price prediction using LSTM

Company	Sector	MAPE (%)
Infosys	IT	2.36
Bajaj Auto	Automobile	3.12
HDFC	Financial services	1.95
Bharat Petroleum	Oil & Gas	5.49
Asian Paints	Consumer goods	2.43
Bharti Airtel	Telecom	6.57



**Fig. 6** Predicting low price for HDFC Bank

those four MAPE values. The MAPE for stock price prediction using LSTM ranges between 2% and 7%. Figure 6 shows the plot for actual training and predicted low prices for the next 60 days.

## 6 Conclusion

The proposed system provides a comprehensive analysis for blue chip companies. FinBert got 0.97 and 0.71 accuracy scores on the Financial PhraseBank and gold commodity dataset, respectively, while VADER got 0.63 and 0.15 accuracy scores. Based on these results, we conclude that models trained on domain-specific corpus perform better. MAPE ranges between 2% and 7% for stock price prediction using the LSTM model. Advisory reports generated as the output of this system will provide comment on the stocks. The limitations are that we have considered limited news sources to avoid false news and provided limited technical indicators. Future work can include more companies. Data can be collected from more news websites, social media, and blogs, and false news detection can be performed to get reliable news.

## References

1. Mohan, S., Mullapudi, S., Sammeta, S., Vijayvergia, P., Anastasiu, D.: Stock Price Prediction Using News Sentiment Analysis. In: 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService), pp. 205–208. IEEE (2019).
2. Kalra, S., Prasad, J.: Efficacy of News Sentiment for Stock Market Prediction. In: 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), pp. 491–496. IEEE (2019).
3. Elmasry, Abbas M.: Predicting Price Trend In The Stock Market Based On Data Analysis, News Sentiment And False-News Detection. *International Journal Of Scientific & Technology Research* 10(4), 401–407 (2021).
4. Shah, D., Isah, H. and Zulkernine, F.: Predicting the Effects of News Sentiments on the Stock Market. In: 2018 IEEE International Conference on Big Data (Big Data), pp. 4705–4708. IEEE (2018).
5. Esichaikul, V., Phumdontree, C.: Sentiment Analysis of Thai Financial News. In: 2nd International Conference on Software and e-Business (ICSEB '18), pp. 39–43. ACM (2018).
6. Samad, P., Mutalib, S., Rahman, S.: Analytics of stock market prices based on machine learning algorithms. *Indonesian Journal of Electrical Engineering and Computer Science* 16(2), 1050–1058 (2019).
7. Agarwal, A.: Sentiment Analysis of Financial News. In: 12th International Conference on Computational Intelligence and Communication Networks (CICN), pp. 312–315. IEEE (2020).
8. Kim, J., Seo, J., Lee, M., Seok, J.: Stock Price Prediction Through the Sentimental Analysis of News Articles. In: 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN), pp. 700–702. IEEE (2019).
9. Jariwala, G., Agarwal, H., Jadhav, V.: Sentimental Analysis of News Headlines for Stock Market. In: IEEE International Conference for Innovation in Technology (INOCON), pp. 1–5. IEEE (2020).
10. Hutto, C., Gilbert, E.: VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. In: Eighth International AAAI Conference on Web and Social Media, 8(1), 216–225 (2014).
11. Malo, P., Sinha, A., Korhonen, P., Wallenius, J., Takala, P.: Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology* 65(4), 782–796 (2014).
12. Sinha, A., Khandait, T.: Impact of News on the Commodity Market: Dataset and Results. In: Future of Information and Communication Conference, pp. 589–601. Springer (2021).

# Strengthening Deep-Learning-Based Malware Detection Models Against Adversarial Attacks



Rohit Pai, Mahipal Purohit, and Preetida Vinayakray-Jani

## 1 Introduction

In 2019, 2 billion computers were present globally, including servers, desktops, and laptops [15]. It means that there are many attack surfaces, and it is of utmost importance to protect these devices. One of the many ways in which attackers infiltrate a system is by using malware. Malware is any software intentionally designed to cause damage to a computer, server, client, or computer network [1]. Malware is one of the biggest threats to computer security [2] in this age, and we must find effective ways to detect and tackle it.

Two major approaches have been used in the past to detect malware: static analysis and dynamic analysis. Static analysis involves using metadata, unique data, or bits of code to identify malware and create signatures and heuristics. Dynamic analysis involves executing the malware in a sandbox environment such as a virtual machine to observe its behavior and determine its malignity. These traditional methods could not keep up with the growth of malware and their variants, and alternative faster methods were needed.

Machine learning and deep learning algorithms were explored as an alternative to traditional approaches as they gave exceptional results in detecting and predicting hidden patterns. Deep learning was preferred to standard machine learning due to its ability to learn complex features and deliver high-quality results. Classical deep learning models gave exceptional results in malware detection but failed against variants of the existing malware families and were susceptible to well-crafted adversarial samples.

---

R. Pai (✉) · M. Purohit · P. Vinayakray-Jani  
Sardar Patel Institute of Technology, Mumbai, India  
e-mail: [rohit.pai@spit.ac.in](mailto:rohit.pai@spit.ac.in); [mahipal.purohit@spit.ac.in](mailto:mahipal.purohit@spit.ac.in); [preeti.vinayakray@spit.ac.in](mailto:preeti.vinayakray@spit.ac.in)

Generative adversarial network or GAN is the perfect solution to the problem faced by classical deep learning models. Initially proposed by Goodfellow et al. [14] in 2014, GANs can be described as a machine learning system where two neural networks, a generator, and a discriminator, compete. In the end, the generator becomes capable of generating samples that are close to real-life samples, and the discriminator becomes a great classifier. Due to adversarial training, GANs prove to be very effective against variants and obfuscated malware [4, 5]. The existing GAN-based solutions are limited due to being trained on smaller and older datasets, and there is need for newer models.

We take inspiration from the architecture proposed by Kim et al. [4] and tune it to create a novel system that is trained on a state-of-the-art dataset [10]. Utilizing adversarial training of GANs and a variety of malicious samples allows our model to detect a wide range of malware and be viable for a long time. Our main contribution is devising a robust system that identifies 11 types of malware within a malicious sample (i.e., multilabel classification) with a high degree of certainty and a low response time.

## 2 Background

The main objective of the survey was to find the extent of research done in the malware detection domain across academia and the industry, compare them, discuss their pros and cons, and find solutions to the existing problems. Papers [11] and [12] are survey papers compiling the existing results and techniques in academia till date. Papers [3–5] represent the state-of-the-art techniques in this domain. Paper [7] provides insights into how malware detection models can be attacked and how one can protect and improve them. Lastly, paper [9] represents the progress of the industry in malware detection.

The survey done by D. Gilbert et al [11], provides a systematic review of machine learning and deep learning techniques used for malware detection and classification. The authors have studied 67 papers that use various static, dynamic, and hybrid approaches for malware analysis. They present the issues and challenges with each type of technique and provide several research gaps. First, class imbalance in the existing datasets was identified as a major gap. Second, there were no benchmark real-world datasets available for malware detection to train the machine learning models. There are services that provide malware binaries freely, but obtaining benign samples is a hassle. Additionally, classifying a file as benign or malicious and classifying a malicious file to its family are time-consuming processes, even for a security expert. Furthermore, there is discrepancy between each dataset's labeling approach that makes it impossible to meaningfully compare the accuracy across different works. Third, malware tends to evolve over time, and new variants and families appear periodically. The machine learning models need to be periodically retrained over time to keep up with this pattern. Lastly, malware authors make the feature representation of a malicious file very similar to that of a benign

file. Recently created classifiers could be easily fooled by well-crafted adversarial samples, and there is a need for adversarial training of the models.

R. Komatwar et al. [12] begin by surveying various categories of malware and classify them into 3 types: malware by platform, malware by fiction, and malware by stubs. A notable point is that across all categories, the attackers use various packing techniques to hide the presence of malware. The authors proceed to analyze the existing static and dynamic techniques of malware detection including various machine learning approaches such as K-means, decision tree, ANN, neuro-fuzzy networks, etc. They also provide a comprehensive analysis of the existing malware image creation and classification techniques. They emphasize on the need to classify malware as images. They state that existing techniques have loopholes that the hackers can exploit, and there is a need for new, complementary, and orthogonal techniques to defeat them.

Z. Cui et al. [3] propose a CNN-based approach to detect malware. Their model is trained using a dataset of over 9342 grayscale images. To avoid the problem of overfitting, they use image augmentation techniques such as rotation, width and height shift, rescale, shear, etc. The authors created models using images of sizes  $24 \times 24$ ,  $48 \times 48$ ,  $96 \times 96$ , and  $192 \times 192$  and compared their results. The authors found out that as the image size increases, the performance of the model becomes better, but the training time also increases. The authors settled on  $96 \times 96$  as the final image size due to an equal trade-off between performance and time. The authors also make use of the Bat algorithm to handle class imbalance over multiple families of malware. Their model classifies 25 malware families with an accuracy of 94.5%. The authors demonstrate that the model achieved better accuracy and speed when compared to other malware detection models. This model lacks due to being trained on a limited number of samples and having no adversarial training.

J.-Y. Kim et al. [4] propose a new GAN-based model called tDCGAN, capable of classifying malware and detecting zero-day attacks within 9 malware families. The authors first convert the malicious executable files into images of size  $63 \times 135$ . A deep autoencoder (DAE) is trained on the dataset, and it learns to pick up important features from the image and tries to reconstruct the same image from the latent representation. The decoder of the DAE is used as the generator of the GAN as it stabilizes the learning process of the GAN. The GAN is then trained, and the discriminator of the GAN is used for malware detection. This system achieves an average classification accuracy of 95.74%. The authors test the resistance of the model against zero-day attacks by adding noise to the existing malware and testing them against the system. The authors compare the performance of their system vs. other machine learning and deep learning techniques and conclude that their system is better even in case of zero-day attacks because of the inherent adversarial training. The main drawbacks of this chapter are the limited number of malware families and the inability to classify samples as benign or malicious.

LSC-GAN proposed by J. Kim and S. Cho [5] is capable of detecting and classifying malware within 9 families. It achieves an average classification accuracy of 96.97%. The drawbacks of this model are that it requires the input malware

images of the same size and that the malicious examples generated by the GAN for training may not be malicious in practice.

R. Podschwadt et al. [7] studied 12 techniques of crafting adversarial examples and 4 defensive techniques that may withstand such attacks on different datasets. The authors found that generating adversarial samples and attacking models are far easier tasks than defending them. The paper demonstrated that adversarial training was the most effective and efficient defense for image-based classifiers. The paper stated a need for approaches tailored toward classifiers based on binary data.

Kaspersky is one of the leading anti-virus vendors in the industry. In one of their white papers [9], they give an overview of how machine learning is used for malware detection. They highlight the salient challenges in building a machine-learning-based malware detection model. They give us an insight into the malware detection techniques used by Kaspersky.

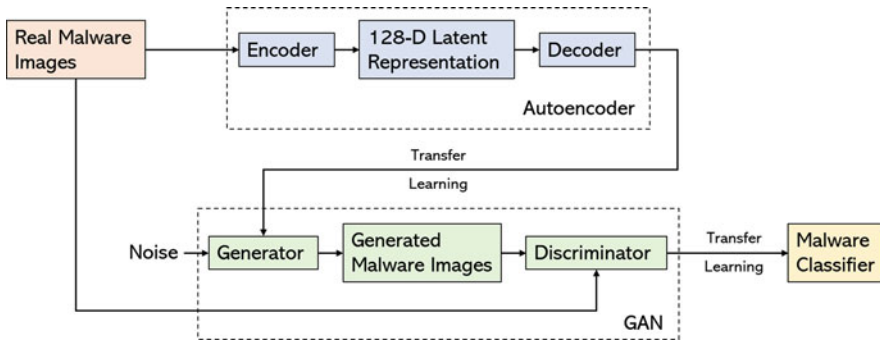
After a thorough literature survey, we observed the need to focus on classifying and detecting malware as images [12] and defending such models by using adversarial training [7]. This need was somewhat satisfied by the models proposed in papers [4] and [5], but these papers lack due to being trained on much smaller and older datasets. There is a need for models trained on newer, larger, and diverse malware datasets containing real-world malicious samples [11] to be viable in the future. There was a stark difference in the quality of models between the anti-malware industry [9] and academia due to the lack of high-quality and large datasets. This gap significantly narrowed when SOPHOS released the SOREL-20M dataset. This gave researchers like us the access to 10 million malicious samples to create drastically better models. We apply the learnings gained through the literature survey on this dataset and propose our system to enhance their work.

The remaining chapter is organized as follows. Section 3 describes the design of our proposed system. In Sect. 4, the technical details and relevant theory of the dataset and the components of our system are presented. The results obtained from training, testing, and validation of the malware classification system are presented in Sect. 5. The user facing component of this system is also shown here. Section 6 summarizes the proposed work and the principal findings of our research. The promising research directions and further improvements in this project are also presented here.

### 3 Proposed Design

Figure 1 shows the overview of our proposed malware classification model. It consists of 4 main components: creation of malware images, autoencoder, GAN, and malware classifier.

The process begins by creating malware images. The binary malware executable files are collected from the SOREL- 20M dataset repository, and we convert them into images using a custom algorithm. We proceed to train the autoencoder using these images. The autoencoder converts these images into a latent representation of



**Fig. 1** Proposed model

128 dimensions and converts them back into images. The weights of the decoder are then transferred to the generator. This process provides stability to the training process of the GAN and helps it converge. The GAN model is now trained, and the discriminator of the GAN model is taken out and used for the malware classification. We proceed to describe in detail each phase of the model.

### 3.1 Creation of Malware Images

As stated earlier, we use SOREL-20M as the dataset for our model. We obtained the binary executable malware files from the dataset's online repository. These files are stored in zlib compressed format to avoid accidental execution of malware. We began by decompressing these files and obtained binary strings (consisting of 0s and 1s) of the malicious code. These strings were converted into grayscale images by forming groups of 8 bits (1 byte) and converting each group to its decimal representation (an integer from 0 to 255). This integer represents the pixel value in the images. Since the length of each binary string may vary, we cannot convert them to images as it is. We need a standardized width or height for the images and pad the missing bits to the strings. We choose to fix the width and vary the height to ensure that the horizontal features of image are preserved. The width of the image is calculated using Table 1. Note that the file size and the length of the binary string are one and the same.

Once the width of the image is obtained, the number of padded bits is obtained using the following equation:

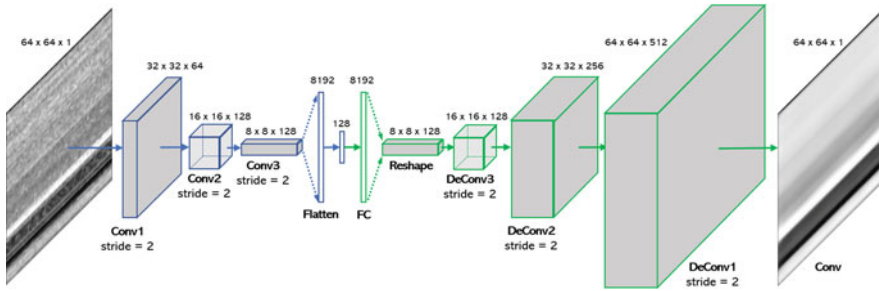
$$padding\ bits = image\ width - (file\ size \% image\ width).$$

After these calculations, we pad the bits to the binary string and create a grayscale image with the calculated width. Since we use convolutional layers in the subsequent phases, we need all the malware images to be of the same size. After



**Table 1** Image width for different file sizes

File size	Image width	File size	Image width
$\leq 10$ KB	32	200 KB $\sim$ 500 KB	512
10 KB $\sim$ 30 KB	64	500 KB $\sim$ 1000 KB	768
30 KB $\sim$ 60 KB	128	1000 KB $\sim$ 2000 KB	1024
60 KB $\sim$ 100 KB	256	$\geq 2000$ KB	2048
100 KB $\sim$ 200 KB	384		



**Fig. 2** Architecture of autoencoder

working with several image sizes, we concluded that smaller image sizes provided less precision, recall, and F1 score, while larger image sizes took more time to train. We finally chose  $64 \times 64$  as our final image size since it was best trade-off between the training time and the evaluation metrics. We resized all the images using cubic and area interpolation. Cubic interpolation was used when image width was less than 64 and area interpolation was used in all other cases. The resized images are then fed to the autoencoder, GAN, and malware classifier for training, testing, and validation.

### 3.2 Autoencoder

Autoencoders are a type of neural network used to compress the raw data (usually images) and reconstruct it from the compressed form [16]. It consists of two parts: an encoder and a decoder. The encoder compresses the input data to the latent representation by retaining only the most important features, and the decoder tries to reconstruct the original data from this latent representation. Once fully trained, the decoder has the ability to independently generate instances of the original dataset from the latent representation. The architecture of the proposed system’s autoencoder is shown in Fig. 2.

As done previously by J.-Y. Kim et al. [4], we leverage this ability of the autoencoder to create a decoder capable of generating malware images from a latent representation of 128 dimensions. We transfer the weights of this decoder to the

generator of the GAN. This gives it a great starting point and helps stabilize the training process of the GAN.

### 3.3 GAN

A GAN consists of two parts: the generative network and the discriminative network, called generator and discriminator, respectively. These two halves train by competing with each other. The task of generator is to generate images that are very close to the dataset provided and fool the discriminator. It creates fake data, and this data along with the actual data from the dataset is fed into the discriminator. The task of the discriminator is to correctly distinguish between the real and fake data. This is very similar to a zero-sum game [17].

We identified the need of using adversarial training on machine learning models in our literature survey. We found that GANs inherently make use of adversarial training and that using them would make our models more robust and resistant to adversarial attacks. This feature is lacking in any other deep learning algorithm and motivated us to use GANs. Hence, we use the GAN in our proposed system to perform adversarial training on the discriminator.

The generator generates fake images using noise vector of size 128 dimensions. Since we previously performed transfer learning on the generator, the generator is capable of producing images that are very close to real malware images and may even resemble variants of malware families. The discriminator is trained using these generated and real malware images. After sufficient epochs, the discriminator is able to recognize these types of malware images with ease. Transferring these weights to the malware classifier helps it in classifying malware present in the dataset as well as unseen variants of malware that may occur in the future. Figures 3 and 4 show the architecture of our system's generator and discriminator.

### 3.4 Malware Classifier

The malware classifier is the final module of the proposed system and performs the classification of the malicious samples to their respective types of malware. Figure 5 shows the general structure of the malware classifier model in our system.

We transfer the weights obtained from the discriminator to the malware classifier. The last layer of the discriminator is replaced by a new fully connected dense layer of size 11, as seen in the figure. This allows the classifier to predict the 11 types of malware and perform multilabel classification. This classifier is much better than any CNN counterpart due to the use of adversarial training.

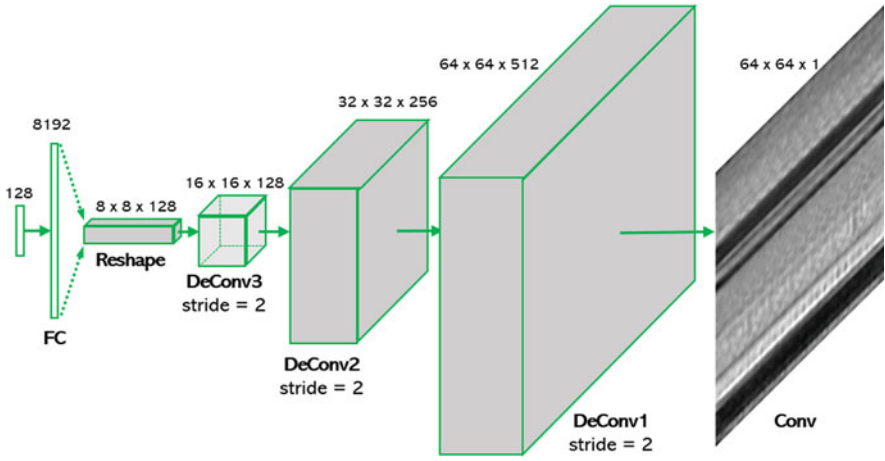


Fig. 3 Architecture of generator

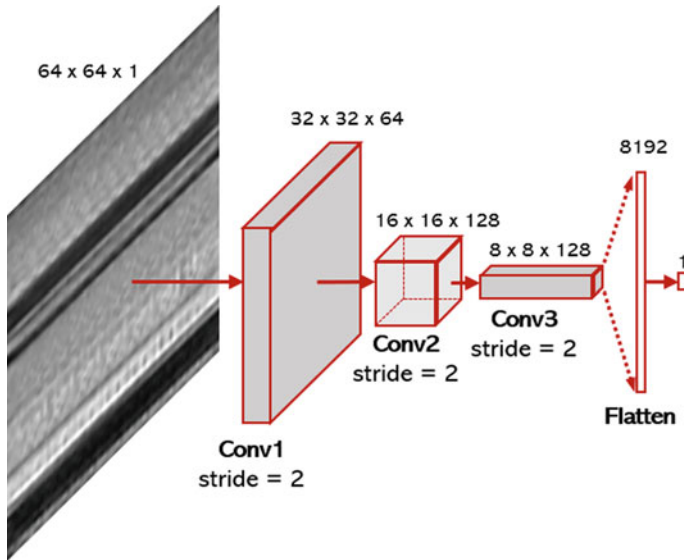


Fig. 4 Architecture of discriminator

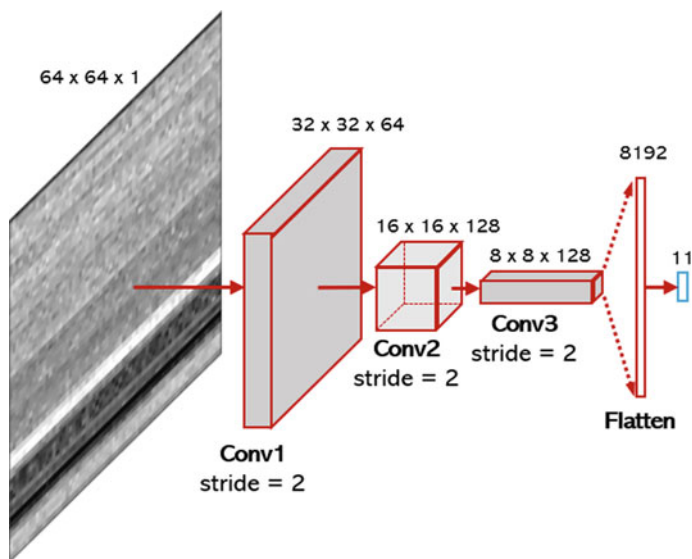


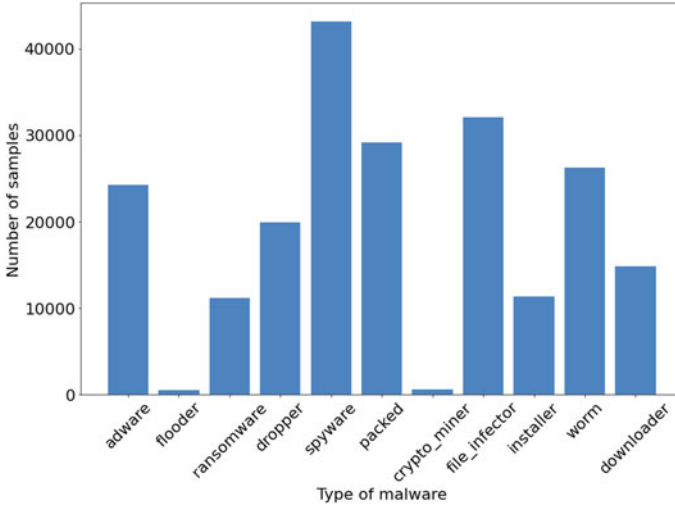
Fig. 5 Architecture of malware classifier

## 4 Experiments

### 4.1 Dataset

The SOREL-20M dataset consists of 10 million malicious binary executable files for Windows operating system. It is one of its kind since it provides access to such a large quantity of data for academic research. The dataset provides a database file consisting of the *hash value*, *first seen time*, and 11 columns for *adware*, *flooder*, *ransomware*, *dropper*, *spyware*, *packed*, *crypto\_miner*, *file\_infector*, *installer*, *worm*, and *downloader* for each malicious file. The columns of the types of malware in these records show the certainty with which each file belongs to a particular type. The higher the number, the higher the certainty. We converted each non-zero value in these columns to a 1 and used it as the labels. Since this dataset contains files with more than one type of malware in it, we chose to perform multilabel classification.

It was infeasible for us to use the entire dataset at this moment since its size is about 8 terabytes. Hence, we chose to train our model on 2 subsets of the dataset: random 10,000 files and the first 100,000 files by first seen time of the malware. The count of each label in the first 100,000 files is shown in Fig. 6. We set aside the first 76.6% of the files as the training set, the next 9.7% as the testing set, and the remaining 13.7% as the validation set [10]. Some of the sample types of files (containing 1 or more types of malware in it) are shown in Table 2 along with their images and its frequency within our subset of 100,000 files. Files containing the malware Spyware, File Infector, and Worm were the most frequent in this subset,



**Fig. 6** Label counts for the first 100,000 files

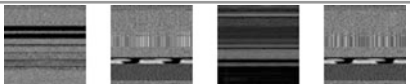
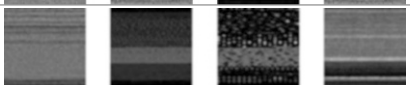
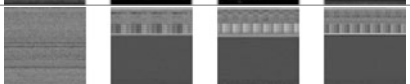
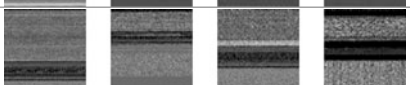
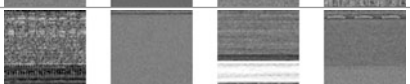
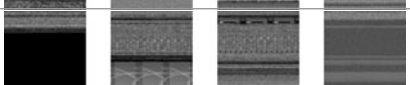
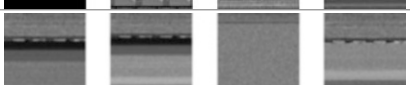


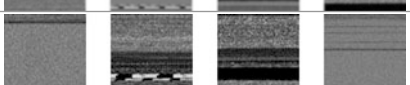
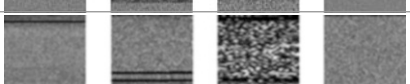

while those containing solely Cryptominer or Flooder were the least frequent. We can see from Table 2 that even though files have the same type of malware, their images are very different.

## 4.2 Experiment Details

All of our experiments were performed using free cloud resources on Kaggle, Google Colab, and Google Cloud. We used the available GPUs to drastically speed up our training times.

We used the Keras library to create all of our models. Convolutional layers form the basis of all of our models since they are key in extracting the important features from the malware images. LeakyReLU was used as the activation function for all convolutional layers except the output layer. LeakyReLU was used instead of ReLU to avoid the dying ReLU problem. Sigmoid function is used as the activation function for the output layer since we are performing multilabel classification. Dropout layer is used after each convolutional layer for regularization and stable learning. Batch normalization layers are added after the dropout layers in the encoder, discriminator, and malware classifier to avoid overfitting. Deconvolution layers were used to rescale and obtain back the images of actual size. We used a stride instead of pooling layers to perform downsampling while training the models:

**Table 2** Sample types of files with their frequency and images

Type of malware	No. of data (no. of test data)	Sample images of each type
Spyware, File Infector, and Worm	11,257 (1086)	
Adware	8228 (814)	
Ransomware, Packed, and File Infector	6566 (641)	
Spyware and File Infector	6187 (573)	
Adware and Installer	3719 (364)	
Dropper, Spyware, and Packed	3150 (310)	
Spyware and Worm	2430 (234)	
File Infector	2219 (218)	
Packed	2073 (196)	
Adware, Installer, and Downloader	2046 (226)	
Flooder and Packed	135 (7)	
Cryptominer	20 (3)	

#### – Autoencoder

- Encoder:  $4 \times 4\text{Conv}@64 \rightarrow 4 \times 4\text{Conv}@128 \rightarrow 4 \times 4\text{Conv}@128 \rightarrow \text{Flatten}$  to size 8192  $\rightarrow$  Fully connected layer of size 128
- Decoder: Fully connected layer of size 8192  $\rightarrow 4 \times 4\text{DeConv}@128 \rightarrow 4 \times 4\text{DeConv}@256 \rightarrow 4 \times 4\text{DeConv}@512 \rightarrow 5 \times 5\text{Conv}@1$

– GAN

- Generator: Same as autoencoder’s decoder
- Discriminator:  $4 \times 4\text{Conv}@64 \rightarrow 4 \times 4\text{Conv}@128 \rightarrow 4 \times 4\text{Conv}@128 \rightarrow$   
Flatten to size 8192  $\rightarrow$  Fully connected layer of size 1

– Malware Classifier

- Decoder:  $4 \times 4\text{Conv}@64 \rightarrow 4 \times 4\text{Conv}@128 \rightarrow 4 \times 4\text{Conv}@128 \rightarrow$  Flatten  
to size 8192  $\rightarrow$  Fully connected layer of size 11

The batch size was set at 32 for all the models. We used the Adam optimizer for training the models with  $\beta_1 = 0.5$  and loss function set as Binary Crossentropy. The learning rate was set to  $10^{-4}$  for the autoencoder, the generator, and the discriminator. The autoencoder is trained for 150 epochs, while the GAN is trained for 200 epochs. In case of the malware classifier, the training is done in 2 phases. First, all the layers except the dense layer of size 11 are frozen. The model is then trained for 500 epochs with learning rate set to  $10^{-3}$ . Second, the remaining layers are unfrozen, and the model is trained again for 400 epochs at a much lower learning rate of  $10^{-5}$  to fine-tune it. We note that fine-tuning improved the results of the malware classifier.

## 5 System Evaluation and Results

We trained our proposed model on random 10,000 samples and the first 100,000 samples by first seen time of the malware. We also compared the performance of our model with a CNN model trained on the same random 10,000 samples. Referring to similar works on multilabel classification [18, 19], we chose the average per-class precision (CP), recall (CR), F1 score (CF1), false-positive rate (CFPR), the average overall precision (OP), recall (OR), F1 score (OF1), false-positive rate (OFPR) as the metrics to evaluate the performance of our classifier.

Table 3 shows detailed class-wise metrics for the model trained on 100,000 samples for reference. Spyware and File Infector show the best results as they are most dominating types in the dataset. Flooder and Cryptominer perform badly due to the lack of samples in this subset, as seen in Fig. 6. This behavior may change when the number of samples is increased.

Table 4 shows the average and overall metrics of all the models. Our proposed models perform much better than the standard CNN model. This justifies the usage of the autoencoder and the GAN in our system.

The metrics for both the proposed models show that our model is able to perform multilabel classification on the given dataset with a relatively high precision, recall, and F1 score and low false-positive rate. The proposed model trained on 100,000 samples performs better than the model trained on 10,000 samples in all metrics except CR and CF1. This is due to drop in precision and recall of the Flooder and

**Table 3** Metrics of the model created using 100,000 samples

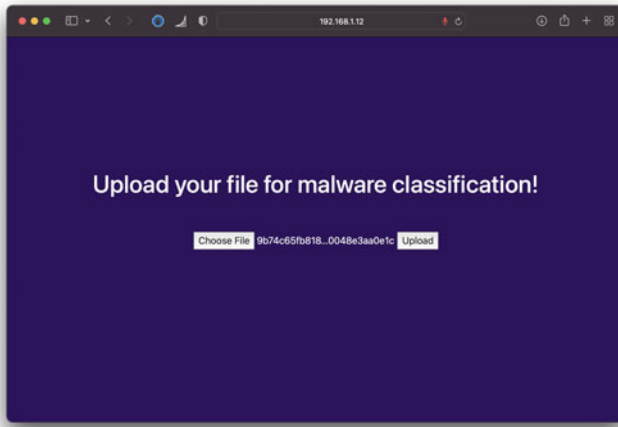
	Adware	Flooder	Ransomware	Dropper	Spyware	Packed	Crypto Miner	File Infector	Installer	Worm	Downloader
Precision	0.8474	0.4872	0.8620	0.8043	0.9019	0.8311	0.2292	0.8966	0.7753	0.8873	0.7585
Recall	0.8267	0.4043	0.8677	0.7696	0.8736	0.8459	0.1746	0.8894	0.7873	0.8852	0.7391
F1 score	0.837	0.4419	0.8648	0.7866	0.8875	0.8385	0.1982	0.8930	0.7812	0.8863	0.7487
False-Positive rate	0.064	0.0025	0.0218	0.0601	0.1080	0.092	0.0047	0.0658	0.0368	0.0522	0.0524



**Table 4** Comparing the results of various models

	CP	CR	CF1	CFPR	OP	OR	OF1	OFPR
CNN—10k Samples	0.7141	0.6808	0.6962	0.0730	0.7961	0.7703	0.7830	0.0613
Proposed model—10k samples	0.7368	<b>0.7627</b>	<b>0.7489</b>	0.0654	0.8165	0.7984	0.8074	0.0558
Proposed Model—100k samples	<b>0.7528</b>	0.7330	0.7422	<b>0.0509</b>	<b>0.8410</b>	<b>0.8531</b>	<b>0.8470</b>	<b>0.0449</b>

The bold values indicate the best value in a given metric amongst the 3 models



**Fig. 7** Landing page of the Web portal

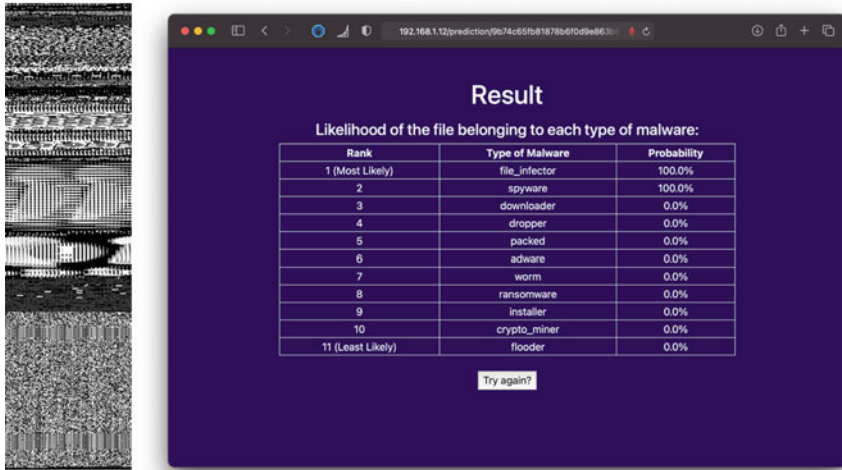
Crypto types of malware, caused by the low count. Hence, this affects only the average metrics and not the overall metrics.

We also deployed the proposed model trained on 100k samples on a website where end users can upload any binary executable file and obtain detailed results on the probability of the file belonging to each type of malware. We created this lightweight web application using Flask. Figure 7 shows the landing page of our website.

By clicking on upload file option, the user can upload a file from their own machine to the website’s server. The website converts the file into a 64 × 64 image using the algorithm stated in Sect. 3.1 and predicts the types of malware present in the file using the malware classification model.

We obtained results for some samples from the test dataset. Here, we discuss the results for one such sample. Figure 8 shows the actual image of the malware on the left side, before resizing it to 64 × 64, and the images on the right side are the classification results that are obtained.

The results show 11 probabilities of a malware belonging to a specific type. As can be seen from the figure, the classifier predicts with 100% probability that the file belongs to File Infector and Spyware types and does not belong to any other type with 100% probability, which is the ideal result.



**Fig. 8** Results of classification of a malware belonging to Spyware and File Infector types

The website has been designed to be easy-to-use and hassle-free, allowing ordinary users to check the malignity of any file quickly without taking the risk of executing the file. We observe that the main bottleneck of the website is while uploading the file to the server and once a file is uploaded, the system can produce results rapidly.

## 6 Conclusion and Future Scope

After a thorough literature survey, we observed that the latest malware detection methods have various machine learning and deep learning models at their core. Most of these models have the common issue of failing against specially crafted adversarial samples and variants of the existing malware families. Moreover, the existing models were trained on older and much smaller datasets. There was a need for models that could withstand such attacks while providing a low false-positive rate, accurate results, scalability, and a fast response time.

Our proposed system is trained on a massive dataset with various types and families of malware and utilizes the power of autoencoders and the adversarial training of GANs to solve this problem. It is designed to identify all types of malware present within a malicious sample with a high degree of certainty. Our best model achieves an overall precision of 84.1%, an overall recall of 85.31%, an overall F1 score of 84.7%, and a false-positive rate of 4.49%, outperforming conventional neural network models. It is deployed on a website, allowing the end users to upload any executable file to the website and check its malignity. Due to the use of GANs,

the model will withstand future variants within the malware families and be viable for a long time in the future.

In the future, we will explore the effect of larger image sizes such as  $192 \times 192$  and  $384 \times 384$  on the model. Further, we used a subset and not the entire SOREL-20M dataset, which limited our model. We plan to explore the effect of training a model with such a large number of samples. We also need to manually test our model against well-known adversarial attacks to strengthen it. Lastly we intend to expand our model by integrating benign samples into the training process. This would ensure that the model becomes a full-fledged malware detection system and can distinguish between malicious and benign files.

## References

1. Wikipedia Contributors. “Malware”, in Wikipedia, The Free Encyclopedia. Accessed: Jul. 13, 2021. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=Malware&oldid=1033368871>
2. “What Happens If Your Computer Is Infected by Malware?”, Consolidated Technologies, Inc. Accessed: Jun. 15, 2021. [Online]. Available: <https://consoltech.com/blog/what-happens-if-your-computer-is-infected-by-malware>
3. Z. Cui, F. Xue, X. Cai, Y. Cao, G. Wang and J. Chen, “Detection of Malicious Code Variants Based on Deep Learning”, IEEE Transactions on Industrial Informatics, vol. 14, no. 7, pp. 3187–3196, July 2018, <https://doi.org/10.1109/TII.2018.2822680>.
4. J.-Y. Kim, S.-J. Bu, and S.-B. Cho, “Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders”, Information Sciences, vol. 460–461, pp. 83–102, Sep. 2018, <https://doi.org/10.1016/j.ins.2018.04.092>.
5. J.-Y. Kim and S.-B. Cho, “Detecting Intrusive Malware with a Hybrid Generative Deep Learning Model”, in Intelligent Data Engineering and Automated Learning - IDEAL 2018, Springer International Publishing, 2018, pp. 499–507. [https://doi.org/10.1007/978-3-030-03493-1\\_52](https://doi.org/10.1007/978-3-030-03493-1_52).
6. J. Yuan, S. Zhou, L. Lin, F. Wang, and J. Cui, “Black-box adversarial attacks against deep learning based malware binaries detection with GAN,” in the 24th European Conference on Artificial Intelligence (ECAI 2020), pp. 2536–2542, <https://doi.org/10.3233/FAIA200388>.
7. R. Podschwadt and H. Takabi, “Effectiveness of Adversarial Examples and Defenses for Malware Classification,” arXiv:1909.04778 [cs], Sep. 2019.
8. H. Li, S. Zhou, W. Yuan, J. Li and H. Leung, “Adversarial-Example Attacks Toward Android Malware Detection System,” in IEEE Systems Journal, vol. 14, no. 1, pp. 653–656, March 2020, <https://doi.org/10.1109/JSYST.2019.2906120>.
9. “Machine Learning Methods for Malware Detection,” Kaspersky Lab, Moscow, Russia. Accessed: Jun. 14, 2021. [Online]. Available: <https://media.kaspersky.com/en/enterprise-security/Kaspersky-Lab-Whitepaper-Machine-Learning.pdf>
10. R. Harang and E. M. Ruddy, “SOREL-20M: A Large Scale Benchmark Dataset For Malicious PE Detection,” arXiv:2012.07634v1 [cs.CR], Dec. 2020.
11. D. Gibert, C. Mateu, and J. Planes, “The rise of machine learning for detection and classification of malware: Research developments, trends and challenges,” Journal of Network and Computer Applications, vol. 153, p. 102526, Mar. 2020, <https://doi.org/10.1016/j.jnca.2019.102526>
12. R. Komatwar and M. Kokare, “A Survey on Malware Detection and Classification,” Journal of Applied Security Research, vol. 16, no. 3, pp. 390–420, Aug. 2020, <https://doi.org/10.1080/19361610.2020.1796162>

13. "Internet Security Threat Report". NortonLifeLock Inc., Tempe, Arizona, U.S. Accessed: Jun. 15, 2021. [Online]. Available: <https://docs.broadcom.com/doc/istr-22-2017-en>
14. I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., "Generative Adversarial Networks", arXiv:1406.2661 [stat.ML], Jun. 2014.
15. "How Many Computers Are There in the World?", Aug. 9, 2019. Accessed: Aug. 20, 2021. [Online]. Available: <https://www.scmo.net/faq/2019/8/9/how-many-computers-is-there-in-the-world>
16. Wikipedia Contributors. "Autoencoder", in Wikipedia, The Free Encyclopedia. Accessed: Nov. 24, 2021. [Online]. Available: <https://en.wikipedia.org/wiki/Autoencoder>
17. Wikipedia Contributors. "Generative adversarial network", in Wikipedia, The Free Encyclopedia. Accessed: Nov. 24, 2021. [Online]. Available: [https://en.wikipedia.org/wiki/Generative\\_adversarial\\_network](https://en.wikipedia.org/wiki/Generative_adversarial_network)
18. A. Shalaginov and K. Franke, "A deep neuro-fuzzy method for multi-label malware classification and fuzzy rules extraction," 2017 IEEE Symposium Series on Computational Intelligence (SSCI), 2017, pp. 1–8, <https://doi.org/10.1109/SSCI.2017.8280788>.
19. J. Lanchantin, T. Wang, V. Ordonez, et al., "General Multi-label Image Classification with Transformers", arXiv:2011.14027 [cs.CV], Nov. 2020.

# Video-Based Micro Expressions Recognition Using Deep Learning and Transfer Learning



Samit Kapadia, Ujjwal Praladhka, Utsav Unadkat, Virang Parekh,  
and Ruhina Karani

## 1 Introduction

In the field of computer vision, the study of facial expressions is an important and rapidly growing issue. They are the key characteristics in non-verbal communication and are important to extract to better understand the emotions and intentions which need to be conveyed. They are extremely important in daily social interactions.

Facial expressions have two subcategories. Firstly, Macro Expressions – These are apparent and distinct facial expressions. They happen in a half-second to four-second time range. They're easy to spot and generally correspond to the content and tone of what's being spoken. Secondly, Micro Expressions – Facial Expressions that occur in a fraction of a second. They have an extremely tiny time period, ranging from 1/25 to 1/2 of a second. They are hidden emotions that develop unconsciously and go unrecognised or misconstrued as a result. There is no method to prevent people from displaying micro-expressions involuntarily and without their knowledge. Emotional intelligence requires the ability to recognise this leaking.

There are two benefits of extraction of micro-expressions. Firstly, Emotional Awareness – because facial expressions are universal, learning to read faces and identifying when an emotional response is beginning, when emotion is being suppressed, and when a person is ignorant to their feelings is critical. Secondly, Detecting Deception – When a person tries to hide his or her emotions, the emotion often manifests itself in the face. The leakage can be limited to a single area of the face (a modest or subtle emotion), or it can be a broad, rapid expression (a micro-

---

S. Kapadia (✉) · U. Praladhka · U. Unadkat · V. Parekh · R. Karani  
Department of Computer Engineering, Dwarkadas J Sanghvi College of Engineering, Mumbai,  
India  
e-mail: [ruhina.karani@djsce.ac.in](mailto:ruhina.karani@djsce.ac.in)

expression). At a speed of 1/25th of a second, micro expressions can be difficult to observe and discern. We may, however, learn to recognise them with practise.

Micro-expression detection is crucial in domains like psychology [1, 2], lie detection [3–5], melancholy detection [6, 7], and Smart Driving applications like detecting the driver’s emotional status while driving, among others.

This rest of the paper is structured as follows – Sect. 2 provides a literature review of different methods used for extraction and the influence of deep learning and transfer learning. Detailed methodology of our evaluation of the performance of all models is discussed in Sects. 3 and 4. Finally, Sect. 5 is the conclusion.

## 2 Literature Review

In the realm of psychotherapy, Haggard et al. [8] was the first to notice the requirement for micro-momentary facial expression recognition. Ekman has been a key figure in the study of facial expressions since 1957. Ekman wrote important books and papers about facial expressions, emotions, and deceit [9, 10]. His discoveries encouraged the development of the Facial Action Coding System (FACS), the first and only complete tool for objectively measuring facial movement [11]. Several researchers have worked on automatic micro-expression identification and have had a lot of success so far. Using FACS, many datasets were created and programmed.

The Facial Action Coding System is the most comprehensive approach for analysing facial expressions currently available. Anatomist Hjorstjo (1970) set the foundation by identifying the units of action based on facial muscle groups, on which Ekman and Friesen created FACS as a measurement system [11]. There have been several revisions, with the most recent one being the 2002 manual, which should be followed today.

Micro-expression detection has been approached in a variety of ways. To uncover hidden information and attenuation of small changes in micro-expression motion, Wang et al. [12] suggested a video motion magnification solution based on the Eulerian motion magnification technique, on CASME || [13] dataset, they had a high accuracy of 75.3%. Li et al. [14] demonstrates a method for detecting micro-expressions by recognising the local and temporal patterns (LTP) of facial movement. Videos are used to produce temporal local features, which are then extracted from projects in the PCA space. To differentiate the micro-expressions, they employ an SVM classifier.

In 2004, a revolutionary technique called Local Binary Patterns (LBP) [15] was proposed, in which LBP histograms were recovered from small face regions and concatenated into a single histo-gram to efficiently describe the facial image. In the computed feature space, a neighbour classifier was used to recognise the objects, with Chi-square as the dissimilarity measure. The Local Binary Pattern on Three Orthogonal Planes (LBP-TOP) [16], an enhanced variant of LBP, has been demonstrated to be particularly successful in describing and identifying dynamic textures in [17].

With the recent advancements in deep learning, it is now possible to train complicated models on massive datasets. For image and video-based applications, learning-based algorithms are the preferred models. In the realm of image recognition, neural networks have considerably improved accuracy performance. Ayyalasomayajula et al. [18] proposes a Convolutional Neural Networks model for micro-expression classification. CNN is used for face detection and Eulerian Video Magnification (EVM) [19] is applied to amplify the micro-expression to a calculated threshold. Further a separate CNN model is used to classify the formerly detected micro-expression into one of the seven universal micro-expressions.

Li et al. [20] discussed a deep multi-task learning model with HOOF (Histograms of oriented optical flow) feature to detect micro-expressions. The study by Li et al. [21] employs a deep multi-task convolutional network to divide the facial region based on detected facial landmarks. It also employs a fused convolutional network to capture optical flow patterns containing muscle changes over a short period of time when micro-expression is present. Finally, the improved optical flow was employed to refine feature information, and the refined optical flow characteristics were detected using a Support Vector Machine classifier to identify micro-expressions. In the work of Xia et al. [22] spatiotemporal recurrent neural networks have been used to capture the spatiotemporal deformations of micro-expression sequences.

Transfer learning has recently had a significant impact on increasing classification task performance. The use of pre-trained weights on some big datasets on a smaller dataset proves to have significant results. Many attempts [23–25] have shown promising results in micro-expression detection.

### 3 Methodology

This study reviews the effectiveness of five pre-trained deep learning models on the MEVIEW (MicroExpression VIdEos in-the-Wild) [26] dataset. Using the concept of transfer learning, the pre-trained weights of the 5 models were trained on the ImageNet [27] dataset, which contains 15 million high-resolution images and 1000 classification categories. These models can be used for feature extraction, prediction, and fine tuning. In our work the models used are MobileNet [28], VGG19 [29], ResNet50 [30], Xception [31] and EfficientNetB7 [32] etc.

#### 3.1 Base Model

VGG19 – It is a successor to AlexNet [33], in which the massive kernel-sized filters of AlexNet are regularly replaced by  $3 \times 3$  kernel-sized filters in the first two convolutional layers. It includes 19 layers and 143.7 million trainable parameters.

ResNet50 – ResNet stands for deep residual networks. It is a convolutional neural network that is 50-layers deep (48 Convolution layers along with 1 MaxPool and 1

Average Pool layer) and has 25.6 million trainable parameters. It is faster than VGG while maintaining a low of complexity. It is a great model for computer vision and image classification tasks where it can be trained without increasing the training error percentage.

**MobileNetV2** – This version of MobileNet is nearly identical to the original, with the exception that it uses inverted residual blocks with bottlenecking features. When compared to the original MobileNet, it contains significantly less parameters of 3.5 million. It has 105 layers of depth. It's utilised for classification, segmentation, embedding, and detection, among other things.

**Xception** – Xception is an acronym for Extreme Inception. This model architecture was inspired by Inception [34], with depthwise separable convolutions replacing the Inception modules. There are 81 layers in the model, with a total of 22.9 million trainable parameters.

**EfficientNetB7** – It has a depth of 438 layers and has over 66.7 million trainable parameters. It is an effective method of scaling up MobileNets and ResNets to obtain better accuracies while being smaller in size and faster on inference.

## 3.2 Preprocessing

The MEVIEW dataset contains data in the form of videos. Frames are used to break down the videos. The **onset frame**, when the micro-expression begins to form, the **apex frame**, where it is most visible, and the **offset frame**, where it begins to diminish, are the three key frames of micro-expressions. For training the model, all frames from onset to offset are used as input. To improve the model's efficiency, the remaining frames are removed. The Haar cascade is used to find the bounding box for the person's facial position in the image. The original images are re-shaped into  $256 \times 256$  dimensions to fit the input requirements of the base models. The classification labels are covered to One-hot encoded values and passed to the training model along with the images.

## 3.3 Model Architecture

The base model is connected to the Fully Connected (FC) layers via a sequential model. The architecture for all 5 models is the same. The optimal batch size for this experiment is 32 and the number of epochs is calculated as 4. To make our models more computationally efficient, we use an optimizer. The only thing that varies in the different models is their hyper-parameters.

Figure 1 describes the model architecture. Firstly, the raw data is preprocessed i.e., the video files in the datasets are converted to their respective frames. The



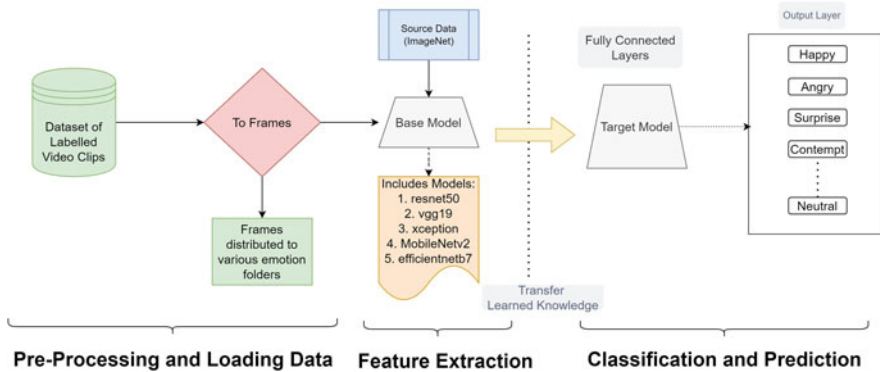


Fig. 1 Model architecture

preprocessing block’s result is used as the input to the compiled model. The compiled model’s output is subsequently transferred to the fully connected (FC) layers as specified. Lastly, the softmax activation layer is coupled to the final fully connected layer, giving us one of the potential categories.

Because the models are loaded with pre-trained weights from the ImageNet dataset, the layers of the base model must be frozen. As a result, the weights do not change and only the subsequent output is changed. The layers must be unfrozen as they progress further into the network and become more data specific. The results on a smaller dataset are noteworthy since we employ transfer learning, and the actual model is trained on a massive dataset.

Not all models could be trained with the same hyperparameters. With Stochastic Gradient Descent Optimizer and a learning rate of 0.0001, models of VGG19 and ResNet50 as their base model performed better. Adam Optimizer is used in the other three models of MobileNetV2, Xception, and EfficientNetB7. At 0.0001, the learning rate of all the models delivered the best results. The loss function was calculated using *categorical cross entropy*, which works well with multi-class classification applications. The loss function can be calculated using the following formula.

$$Loss = - \sum_{i=1}^{output\ size} y_i \cdot \log \hat{y}_i$$

Here  $y_i$  stands for the probability that an event  $i$  occurs. The sum of all  $y_i$  equal to 1. The minus sign indicates that the loss value will drop as the distribution gets closer to each other.

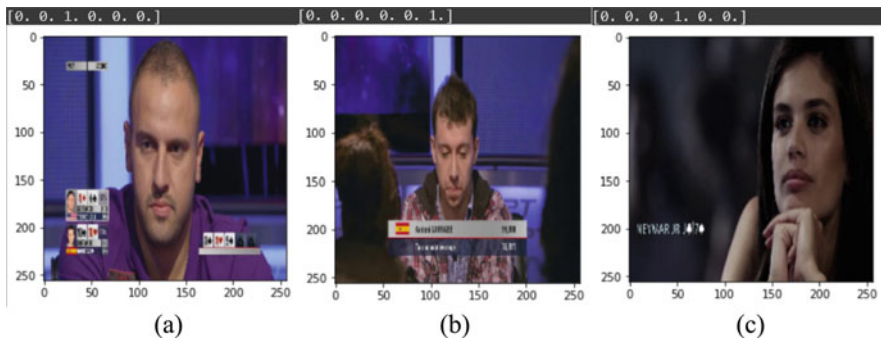
### 4 Results and Analysis

In this section, we have discussed the performance of pre-trained weights of various deep learning models on multiple spontaneous micro-expression datasets. The experiments are performed using Google colab’s GPU server with NVIDIA T4 16GB graphic processor.

This study uses the MEVIEW (MicroExpression VIdEos in-the-Wild) dataset. It consists of 31 videos of 16 people downloaded from the internet, mostly from poker games and TV interviews. It depicts five emotions and contains macro- and micro-expressions. In long movies, the onset and offset frames of the micro-expressions were tagged, followed by further annotation using FACS coding and the determination of emotion kinds. The datasets were split into training data (80%) and test data (20%). We discussed the accuracy, F1 score, and AUC of our model to assess its performance. A detailed explanation of each metrics is discussed below, and the performance summary of all models is shown in Table 1. Figure 2 shows the results of our predictions.

**Table 1** Results of different models

Model	Accuracy	Loss	Precision	F1 score	Recall	AUC
Xception	0.983	0.099	0.983	0.988	0.983	0.994
ResNet50	0.976	0.232	0.976	0.982	0.976	0.986
MobileNetV2	0.976	0.094	0.978	0.978	0.974	0.999
VGG19	0.967	0.261	0.967	0.968	0.967	0.985
EfficientNetB7	0.967	0.125	0.967	0.977	0.967	0.998



**Fig. 2** Show 3 frames labelled (a), (b) and (c). [‘Anger’, ‘Contempt’, ‘Fear’, ‘Happiness’, ‘Neutral’, ‘Surprise’] is the emotion classification. The frame (a) shows the micro-expression recognised as ‘Fear’, for (b) the micro-expression is ‘Surprise’ and for (c) ‘Happiness’

## 4.1 Accuracy

Accuracy is a critical evaluation criterion. It can be defined as the percentage of correct predictions made by our model. The following formula can be used to calculate accuracy.

$$\text{Accuracy} = \frac{\text{Number of Correct Prediction}}{\text{Total number of Prediction}} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where  $TP$  = True Positives,  $TN$  = True Negatives,  $FP$  = False Positives, and  $FN$  = False Negatives.

## 4.2 F1-Score

The F-score, also known as the F1-score, is an evaluation metric for a model's accuracy on a dataset. It's a tool for assessing classification systems. It is defined as the harmonic mean of precision and recall, and is a method of combining precision and recall in a model. It is defined mathematically as follows.

$$F_1 = \frac{2}{\frac{1}{recall} + \frac{1}{precision}} = 2 \times \frac{precision \times recall}{precision + recall} = \frac{2TP}{2TP + FN + FP}$$

## 4.3 Area Under ROC Curve (AUC)

The ROC (receiver operating characteristic curve) is a graph that shows how well a classification model performs across all categorization levels. AUC assesses the full two-dimensional area beneath the complete ROC curve from (0,0) to (1,1). It is a composite measure of performance that considers all possible categorization levels. The trained models are tested on the 20% test split performed earlier and there are further evaluated. Their results are shown in Table 1.

Table 1 indicates that results obtained from the Xception model are better than the other transfer learning models. MobileNetV2 and ResNet50 show high performances too and can also be used to achieve high accuracy in classification of micro-expressions.

Figure 3 posits a comparison of results achieved from the 5 transfer learning models based on evaluation metrics such as accuracy, loss, F1-score, precision and recall.

To further evaluate the best model i.e. Xception, we plot a confusion matrix as shown in Fig. 4. The X- and Y-axes represent the six emotions utilised in the classification procedure.

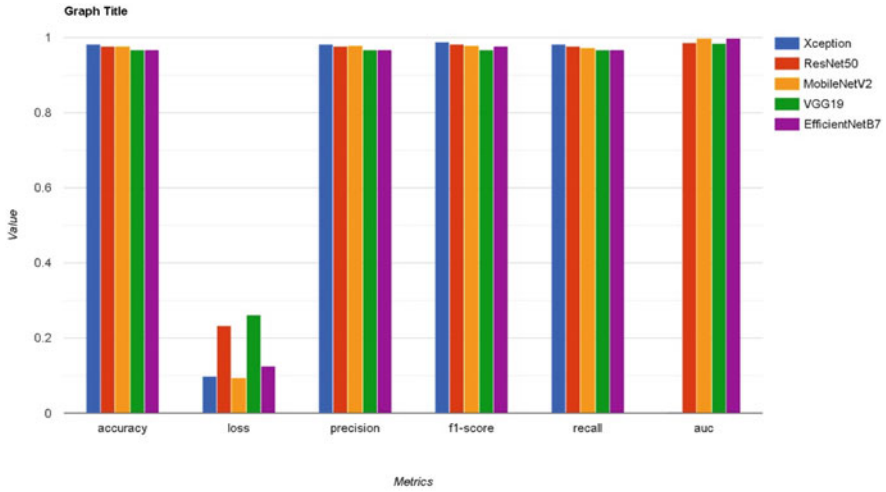


Fig. 3 Graphical representation of different model metrics

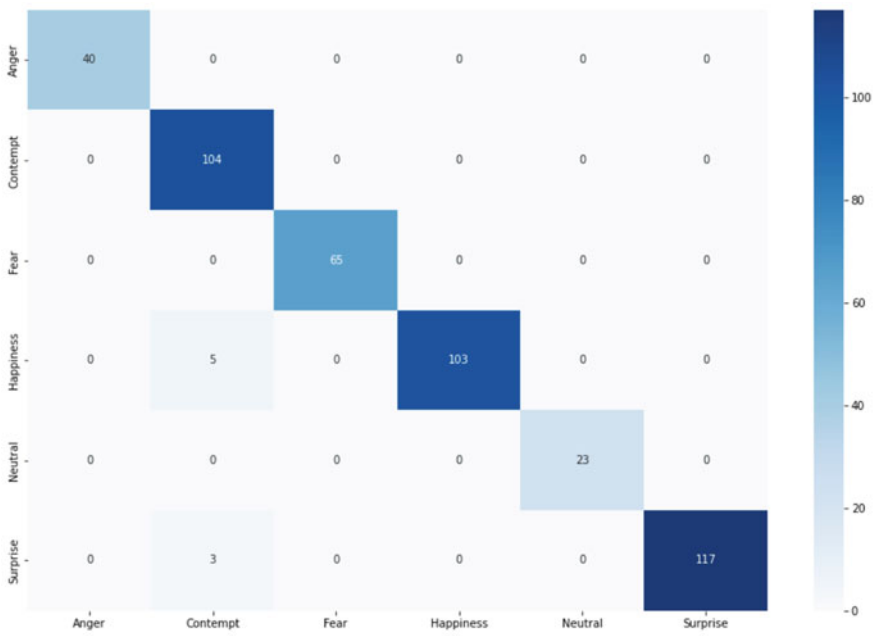


Fig. 4 Confusion matrix for Xception based model

## 5 Conclusion

This study performs an overall analysis as well as in-depth analysis of the video-based micro-expression classification. An in-depth analysis and overview of different methods for Micro-expression recognition and classification is discussed. The study discusses the necessary pre-processing steps required for video-based micro-expression detection tasks. Further, a careful study and analysis of effectiveness of 5 transfer learning models on the in-the-wild dataset is conducted. The models that were learned on the ImageNet dataset were further trained on the MEVIEW dataset using transfer learning and pre-trained weights. The Xception model resulted the best accuracy of 98.26% surpassing all previous results. The results indicated that transfer learning can have high performance and is an optimal methods when applied to this specific domain.

## References

1. Datz, F., Wong, G. and Löffler-Stastka, H., 2019. Interpretation and working through contemptuous facial micro-expressions benefits the patient-therapist relationship. *International Journal of Environmental Research and Public Health*, 16(24), p. 4901.
2. Bhushan, B., 2015. Study of facial micro-expressions in psychology. In *Understanding facial expressions in communication* (pp. 265–286). Springer, New Delhi.
3. Owayjan, M., Kashour, A., Al Haddad, N., Fadel, M. and Al Souki, G., 2012, December. The design and development of a lie detection system using facial micro-expressions. In *2012 2nd international conference on advances in computational tools for engineering applications (ACTEA)* (pp. 33–38). IEEE.
4. Barathi, C.S., 2016. Lie detection based on facial micro expression body language and speech analysis. *International Journal of Engineering Research & Technology*.
5. Jordan, S., Brimbal, L., Wallace, D.B., Kassim, S.M., Hartwig, M. and Street, C.N., 2019. A test of the micro-expressions training tool: Does it improve lie detection?. *Journal of Investigative Psychology and Offender Profiling*, 16(3), pp.222–235.
6. Grobova, J., Colovic, M., Marjanovic, M., Njegus, A., Demire, H. and Anbarjafari, G., 2017, May. Automatic hidden sadness detection using micro-expressions. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)* (pp. 828–832). IEEE.
7. Grobova, J., Colovic, M., Marjanovic, M., Njegus, A. and Anbarjafari, G., 2019. Going deeper in hidden sadness recognition using spontaneous micro expressions database. *Multimedia tools and applications*, 78(16), pp. 23161–23178.
8. Haggard, E.A. and Isaacs, K.S., 1966. Micromomentary facial expressions as indicators of ego mechanisms in psychotherapy. In *Methods of research in psychotherapy* (pp. 154–165). Springer, Boston, MA.
9. Ekman, P. and Friesen, W.V., 1971. Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17(2), p. 124.
10. Ekman, P. and Keltner, D., 1997. Universal facial expressions of emotion. Segerstrale U, P. Molnar P, eds. *Nonverbal communication: Where nature meets culture*, 27, p.46.
11. Ekman, P. and Friesen, W.V., 1978. Facial action coding system. *Environmental Psychology & Nonverbal Behavior*.
12. Wang, Y., See, J., Oh, Y.H., Phan, R.C.W., Rahulamathavan, Y., Ling, H.C., Tan, S.W. and Li, X., 2017. Effective recognition of facial micro-expressions with video motion magnification. *Multimedia Tools and Applications*, 76(20), pp. 21665–21690.

13. Yan, W.J., Li, X., Wang, S.J., Zhao, G., Liu, Y.J., Chen, Y.H. and Fu, X., 2014. CASME II: An improved spontaneous micro-expression database and the baseline evaluation. *PLoS one*, 9(1), p. e86041.
14. Li, J., Soladie, C. and Segulier, R., 2018, May. Ltp-ml: Micro-expression detection by recognition of local temporal pattern of facial movements. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)* (pp. 634–641). IEEE.
15. Ahonen, T., Hadid, A. and Pietikäinen, M., 2004, May. Face recognition with local binary patterns. In *European conference on computer vision* (pp. 469–481). Springer, Berlin, Heidelberg
16. Mattivi, R. and Shao, L., 2009, September. Human action recognition using LBP-TOP as sparse spatio-temporal feature descriptor. In *International Conference on Computer Analysis of Images and Patterns* (pp. 740–747). Springer, Berlin, Heidelberg.
17. Li, J., Soladie, C., Segulier, R., Wang, S.J. and Yap, M.H., 2019, May. Spotting micro-expressions on long videos sequences. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)* (pp. 1–5). IEEE.
18. Ayyalasomayajula, S.C., Ionescu, B. and Ionescu, D., 2021, May. A CNN Approach to Micro-Expressions Detection. In *2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI)* (pp. 345–350). IEEE.
19. Wu, H.Y., Rubinstein, M., Shih, E., Guttag, J., Durand, F. and Freeman, W., 2012. Eulerian video magnification for revealing subtle changes in the world. *ACM transactions on graphics (TOG)*, 31(4), pp. 1–8.
20. Li, X., Yu, J. and Zhan, S., 2016, November. Spontaneous facial micro-expression detection based on deep learning. In *2016 IEEE 13th International Conference on Signal Processing (ICSP)* (pp. 1130–1134). IEEE.
21. Li, Q., Zhan, S., Xu, L. and Wu, C., 2019. Facial micro-expression recognition based on the fusion of deep learning and enhanced optical flow. *Multimedia Tools and Applications*, 78(20), pp. 29307–29322.
22. Xia, Z., Hong, X., Gao, X., Feng, X. and Zhao, G., 2019. Spatiotemporal recurrent convolutional networks for recognizing spontaneous micro-expressions. *IEEE Transactions on Multimedia*, 22(3), pp. 626–640.
23. Kadakia, R., Kalkotwar, P., Jhaveri, P., Patanwadia, R. and Srivastava, K., 2021, October. Comparative Analysis of Micro Expression Recognition using Deep Learning and Transfer Learning. In *2021 2nd Global Conference for Advancement in Technology (GCAT)* (pp. 1–7). IEEE.
24. Peng, M., Wu, Z., Zhang, Z. and Chen, T., 2018, May. From macro to micro expression recognition: Deep learning on small datasets using transfer learning. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)* (pp. 657–661). IEEE.
25. Wang, S.J., Li, B.J., Liu, Y.J., Yan, W.J., Ou, X., Huang, X., Xu, F. and Fu, X., 2018. Micro-expression recognition with small sample size by transferring long-term convolutional neural network. *Neurocomputing*, 312, pp. 251–262.
26. Husák, P., Cech, J. and Matas, J., 2017, February. Spotting facial micro-expressions “in the wild”. In *22nd Computer Vision Winter Workshop (Retz)* (pp. 1–9).
27. Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
28. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510–4520).
29. Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
30. He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).
31. Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251–1258).

32. Tan, M. and Le, Q., 2019, May. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105–6114). PMLR.
33. Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
34. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818–2826).

# Trustworthiness of COVID-19 News and Guidelines



Shubhanshu Singh, Lalit Nagar, Anupam Lal, and B R Chandavarkar

## 1 Introduction

The worldwide pandemic COVID-19 is an infectious illness that has caused a tragedy all over the world over past few years and has put human lives in jeopardy. In the whole world, various social media platforms ([Whatsapp](#), [Snapchat](#), Twitter, etc.) became the primary source of information transmission about COVID-19 disease. It includes the impact, a variety of transmissions through various means, publicizing precautions, and do's and don'ts to be followed to deal with COVID-19 and its effects [1]. People get manipulated and misled by these social media's fake news and false guidelines that go viral all over there. It results in serious panic situations among the people, and to deal with it, a solution is much needed. Some attackers or hackers put their fake website links with manipulated headings that breach the privacy of confidential data or create economical loss to people. It will identify such cases and report about the same [2].

This chapter tries to implement an optimal solution using various kinds of layers and different optimization functions. It particularly gives better performance in the case of sequential data using ML and DL frameworks trained with the dataset for identifying the fake news and guidelines spread over on COVID-19. To train the model, a dataset was taken from the Twitter API. We first analyze what could be the way of identifying the fake news and guidelines and train our model accordingly [3].

The workflow of the chapter goes as this chapter consists of eight sections. In Sect. 1, the introductory part of the paper is presented. In Sect. 2, the impact of non-trustworthy COVID-19 news and guidelines is discussed that how fake news or

---

S. Singh (✉) · L. Nagar · A. Lal · B. R. Chandavarkar  
Department of Computer Science and Engineering, National Institute of Technology Karnataka,  
Surathkal, India



false guidelines are spreading, and how it can impact adversely to society's health. In Sect. 3, it is defined how to measure the trustworthiness of news through different steps such as authenticated source, transparency over data, user focus on it, and data quality. In Sect. 4, related work is described thoroughly. In Sect. 5, the detailed description of the used dataset is presented with what steps to follow for data preparation such as cleaning and preprocessing, feature extraction, feature selection, and sampling of the trained set. In Sect. 6, the fake detection model is presented where training and performance analysis is discussed. In Sect. 7, the implication and its future work are discussed with what could be the future research direction over it, and we finally concluded the paper in Sect. 8.

## **2 Impact of Non-Trustworthy News and Guidelines**

Fake news or guidelines regarding the COVID-19 situation in any area may arise drastic changes in the health situation of that area. But as in the current scenario, the new generation's dependencies on Internet and social media platforms increase drastically, which is leading to the security and authenticity issue of any information rolling over there. Readily and rapidly access to content leads the people to follow it blindly irrespective of verifying the source that is influencing the people in anyhow manner, being unknown of the facts and truthiness of any information. Being confined to the house in lockdown, people tend to spend time being online, and also that is the major source to know the pandemic situation that raises the concern of assuring that authenticity and correctness of any info are maintained. In this section, it is explained how society may be impacted adversely by fake news or guidelines.

### ***2.1 Impact of Fake News***

In today's time, people are more influenced by social media news. Spending many hours on social media platforms, people tend to believe in the news spread over there. Without cross-checking the source, people mislead society or themselves by sharing the contents present over there. It is a serious need to understand the difference between opinion and news [4]. Junk news is a more common word that encompasses a wide range of materials, such as making party-influenced propaganda or party-oriented journalism. Fake information may provide inappropriate or wrong information to people. It may create a panic situation in society. As a student, we are expected to find, evaluate, and reference trustworthy sources in a variety of ways. If we include fake news as proof for our proposal or as part of our research, it may raise doubts about the integrity of the sources we use as a whole and our ability to identify quality news.

Spammers of false news may alter the material to make it appear to be fresh and current. Through fake news, many spammers and hackers can manipulate public perception toward sensitive news such as COVID-19 awareness and precautions. And sometimes they get private information from the public through fake links present in the news. Despite being on the true part, the spreading of fake news questions the credibility of sources [5]. With social media and the Internet uses extensively, it is now possible for a few groups to push fake news that aligns with their beliefs and disparege that which does not. It may be harmful to proceed without knowing all facts, but it may be drastic adverse to step on in an action of or on any fake news. Whatever type of news is getting watched, it is a need to know what type of news is visited and whether it should be entrusted or not.

Informational overload can result in panic, fear, despair, and exhaustion [6]. It is even more difficult to question news that affirms what we believe or lends itself as proof to an argument we are making. But just as something that we do not agree with is not always fake news, news that we do agree with is not always real. With an unbiased mind, we leave our thoughts and prejudices out of it when checking the news. As more people go for news online that directly opposes scientific research, researchers are put in the situation of having to defend the proof of their doing. When the news was conveyed through print, television, and radio, there was less possibility for people to publicly comment, evaluate, or disagree with the info given by researchers.

## ***2.2 Impact of False Guidelines***

It is simple for everyone to contact their loved ones, publish about their lives without filter, and keep up with current events by using social networking sites [7]. For the past few years, a majority of daily hot issues have been tied to the COVID-19 scenario, daily case news, and advice recommendations to follow, among other things. When the number of people infected with the COVID-19 virus rises, the government publishes recommendations on limits, face masking, social distance, testing, and other standard operating procedures (SOPs). In many nations, the COVID-19 impacted socially, economically, physically health perspective. Afterward, the research result is in the hope to yield outcomes related to creating or inventing the novel treating procedures, medications, and different stages of vaccines in the near future. In the pandemic pressure to obtain an instant result, much research is going on publicly and to be famous, many things are not done properly and issued.

These rules are to be communicated to the public via news outlets and through different social media news feeds. The propagation of COVID-19 resulted in a frenzy of social media platforms. The majority of venues was utilized to inform society about important news, rules, and safeguards. Unfiltered conspiracy theories and agendas, according to WHO, are rumoring faster than the pandemic, generating psychological fear, false medical advice, and economic disruption [8]. While

disseminating these guidelines to the public, social media platforms play a critical role. As a result, many people edit or manipulate the genuinely issued instructions for their own gain, spreading false or misleading guidelines that have a large-scale negative impact on society's health as a medical pandemic.

If information or a guideline earns trust and confidence, it is reliable [9]. When information such as coronavirus will get destroyed at higher temperature or the COVID-19 virus does not propagate through air medium began to circulate, many people and governments in hot-climate countries realized that COVID-19 would not affect them. Following this, the World Health Organization (WHO) stated, based on facts, that coronavirus may be propagated everywhere, even if the climate is hot and humid. It gives rise to the theory that people with incorrect information are not taking timely precautions, which causes the disease to spread widely in that location. After explanation also, it does not get conveyed to all in proper time due to overspreading of wrong info earlier that resulted in serious loss.

### **3 Measurement of Trustworthiness**

News and information are all around. We consume a lot of pieces of information through various social media platforms (Facebook, Twitter, etc.), web content, mobile applications, news, and various online streams. But there is a problem with the content's trustworthiness as more than 80% of the information on news and social media is not correct. Sometimes frauds make these platforms to play with people's sentiments. So there is a lot to define and achieve the trustworthiness of the news. Various criteria are responsible to make content trustworthy, which include proper authentication of information generation source. Is news or guideline transparent or not? The quality of data is also important. In this section, further, all these criteria are covered in more detail.

#### **3.1 *Authenticated Source***

In this section, we will talk about authenticated sources from where we can verify whether this source of generation is authenticated or not. It can be specialist, researchers, WHO, and many other authority and institutions. The file must give a clear, authenticate, and trustworthy flow of processes and way to demonstrate to the using people that it can ensure data integrity, authenticity, correctness, dependability, and accessibility through time. There are a variety of methods for determining whether or not a source is reliable. (a) People can visit the official website of the government of a particular region for the advisory and info provided. (b) People can use social media platforms also but only through authenticating

channels that are having the blue tick sharing own source news. (c) People also can get news from other social media sources or channels. But they should bind themselves to cross-verify once with the source provided or through the official authorized to authenticate the source.

### **3.2 *Transparency***

Giving accurate information about the repository, the targeted user demographic, the objective and technological capabilities, as well as the terms of service, is another aspects of transparency. Non-verified articles created with the negative goal mentioning one's or a group's benefits are examples of fake news. These are news with misleading info intended to harm or use to gain financially or politically [10]. Adding a policy that every news, advisory, or shared guideline to the public should have associated with the source provided in it will lead to transparency over the info provided. The reader can cross-verify from the source provided we can easily anticipate believing or not over the source given. Also enclosing the authorized documents for the public may increase the transparency in any process and thus may intend to believe over any source. This methodology can be used to guide the identification through the transparency criteria.

### **3.3 *User Focus***

Data files must have the basis to get the info regarding the community to prioritize and understand their need about and on different situations that will be community- or group-oriented and differ from one another; it should connect with the target user group's data flow and its managing practices and thus should be able to predict the community's evolving demands. It discovers the values of people who participate in the dissemination of bogus news, such as publishing, enjoying, sharing, and commenting. Unlike information such as misleading reviews, fake news may attract both malicious and unintentional users [11]. To guarantee that supplied information is getting diverse views, one should read news from a range of sources. Do not choose one source over another because it is more trustworthy or confirms what you already know to be true. Fake news headlines are frequently written with the intent of eliciting a strong reaction. If a headline promotes something that appears to be too good to be true or provokes fury about a certain issue or incident, be skeptical. One way that the distributor of fake news incites its spread is by evoking strong emotions.

### 3.4 Data Quality

The file contains relevant methods for checking data and metadata, as well as ensuring that enough information has been provided for the users to make assessment of the quality [11]. Files must be well enough to validate the data along with metadata for completeness and quality, as well as guarantee that there is a sufficient amount of info of the data for the chosen group or community to judge data purity. Data fluffiness and ease of data intercept are ensured by quality control tests on the data. The ability of scientists to comment on or rate data along with metadata and the availability of citations to relevant publications or links to citation indices are all examples of good practice. Together, these proposed data management best practices help to improve the quality and integrity of research data files, for both data deposit and reuse. However, certification necessitates a significant amount of self-evaluation, as well as a long-term activity, monitoring, maintenance, and continuous development.

## 4 Related Work

Nowadays, researchers used text generated from various information sources to analyze the sentiment in the field of natural language processing (NLP), since the early 1990s. Sentiment analysis (SA) has gained popularity as a sub-field of NLPs for determining the researcher's flow. SA can classify text into various categories: neutral, not specific about the content in either positive or negative, it can be out of context too; positive, this kind of category gives positive sense about information; negative, negative categories give just opposite sense of what positive categories give or assess sentiment strength [12]. Machine learning is the method of learning machines on some sort of dataset to predict the future without iteration of the human being. To analyze sentiment analysis in today's time, machine learning models are used massively for a variety of purposes [13, 14]. Tourism [15], languages on media platform [16], visitors feedback, third-party feedback, and review, investing in the right and highest return giving asset, real estate, and movie reviews are all examples of where SA is employed. The value of SA is based on its capacity to provide improved insights, pursue a competitive edge, and arrive at optimal solutions. For example, [12] utilized SA to learn about people's attitudes regarding hospitality companies. Furthermore, the author in [17] explored the impact of SA on physical and online sales services, the implications for product sales and purchases, and price strategy modifications, as well as the loss between feelings and investment returns on a various time frame basis. In climate change case [18], if we used SA to determine the amount of agreement between climate experts and policymakers, and recommended that they collaborate, it can directly impact their return of business. Another use of SA is evaluating clients in order to make purchase choices [19].

While the significance and ramifications of SA are immeasurable, false information is a huge problem in the field.

Shu et al. [19] presented a model-based method for constructing an unknown model that captures changes in textual information that is the most relevant postings with time. They used 5M approx. postings from Weibo micro and Twitter—blogs to perform their studies. They compared various popular and efficient machine learning and deep learning algorithms with some modifications such as regression, naive Bayes, random forest, recurrent neural network (RNN), and convolutional neural network (CNN). Another research proposed a hybrid deep learning model based on the same dataset and published the FakeNewsNet dataset, which was used to test several machine algorithms such as support vector machine (SVM) and various complex and modified neural networks on a dataset. A recurrent network model for identifying fake information and unrealistic fraudulent claims was introduced by DeClare [20]. It employs proofs and counter-proofs acquired from the huge source of information on the Internet to support or refute a claim. The scientists used at least four separate datasets to train a bidirectional LSTM model and got an overall accuracy of 80%. Ksieniewicz and colleagues and Singh et al. [21] developed an attention-based model whose backbone is LSTM that uses 13 different semantic and user factors to differentiate between false and non-fake tweets using tweeted information. They compare and contrast the attention-based LSTM recurrent neural network to a variety of classical machines and deep learning models. The attention-based LSTM recurrent neural network offers the best effect, according to the solution.

## 5 Dataset

This section covers the various steps involved in gathering Twitter datasets for use in generating sentiment ratings and training the model. The dataset contains features such as ID, URL, title, and tweet. We have to construct a new dataset by merging four files and adding a new column called a label. The label column has two values: zero for false news and one for true news. Then we divided the dataset into training and testing datasets into 7:3 ratio [22]. In data preparation, the following procedures are followed:

- **Data Cleaning and Preprocessing:** Unwanted data, irrelevant data, noise, and missing values must be removed during data cleaning and preprocessing in order to obtain data in the desired and appropriate format, which will increase data accuracy. We remove hypertext transfer protocol (HTTP), www, hashtags, punctuations, and irrelevant numerals in this phase. Data were preprocessed after cleaning, including removal of stop word, stemming, and lemmatization. Only the stop words were eliminated in this case.
- **Feature Extraction:** After data cleaning and preprocessing, it is very much important to get the relevant features to get a better and more concise dataset to

train. Since data has many features that are not relevant or less relevant, we can drop them. Most relevant feature includes words and sentences, after extracting features convert them into ELMo vectors using ELMo word embedding technique.

The ELMo (Embeddings from Language Models) vector assigned to a token is usually a function of the entire sentence containing that word unlike traditional methods such as word2vec, GloVe, which give the same vector for the same word. They have only one representation of each word, so they cannot detect how the meaning of each word can change based on the context, whereas ELMo allows the same word to exist in different vectors under different contexts, an ability to model polysemy [23]. ELMo embeddings are context-sensitive, producing different representations for words that share the same spelling but have different meanings (homonyms) such as “bank” in “river bank” and “bank balance.”

- **Feature selection and dimensionality reduction:** In text classification applications, dimensionality reduction is a key aspect in improving model results by lowering dimensions and getting more relevant characteristics that contribute the most to the final output.
- **Sampling the training set:** To re-balance class distributions, sampling is done on an unbalanced collection of data. Undersampling and oversampling are two forms of sampling that can be done. We generally remove or combine the data when we undersample. Because we are removing data, it is probable that we will be deleting user data as well, and this strategy is best used when there is a lot of data or duplicate data. The other strategy is oversampling, which we will use to expand the training dataset. This technique is used when we need to balance data class variation.

## 6 Fake Detection Model Using Recurrent Neural Network (RNN)

This chapter majorly focused on the long short-term memory machine learning model that is a type of RNN. This model uses previous output and is used in input for the next phase RNN. Schmidhuber and Hochreiter introduced the concept of the long short-term memory model. It addressed the problem of RNN long-term dependency, in which the RNN cannot tell the word stored in long-term memory but may make more accurate predictions based on the most recent data [24]. The performance of the recurrent neural network degrades as the gap length rises. By default, long short-term memory can store information for a very long time. It is used for time-series data processing, prediction, and classification.

In Fig. 1, it can be seen that first the data has to be extracted from Twitter API. Then the text is changed in single case, ideally lower case, using text preprocessing. In this step, the text is split into smaller units. We can use sentence tokenization on

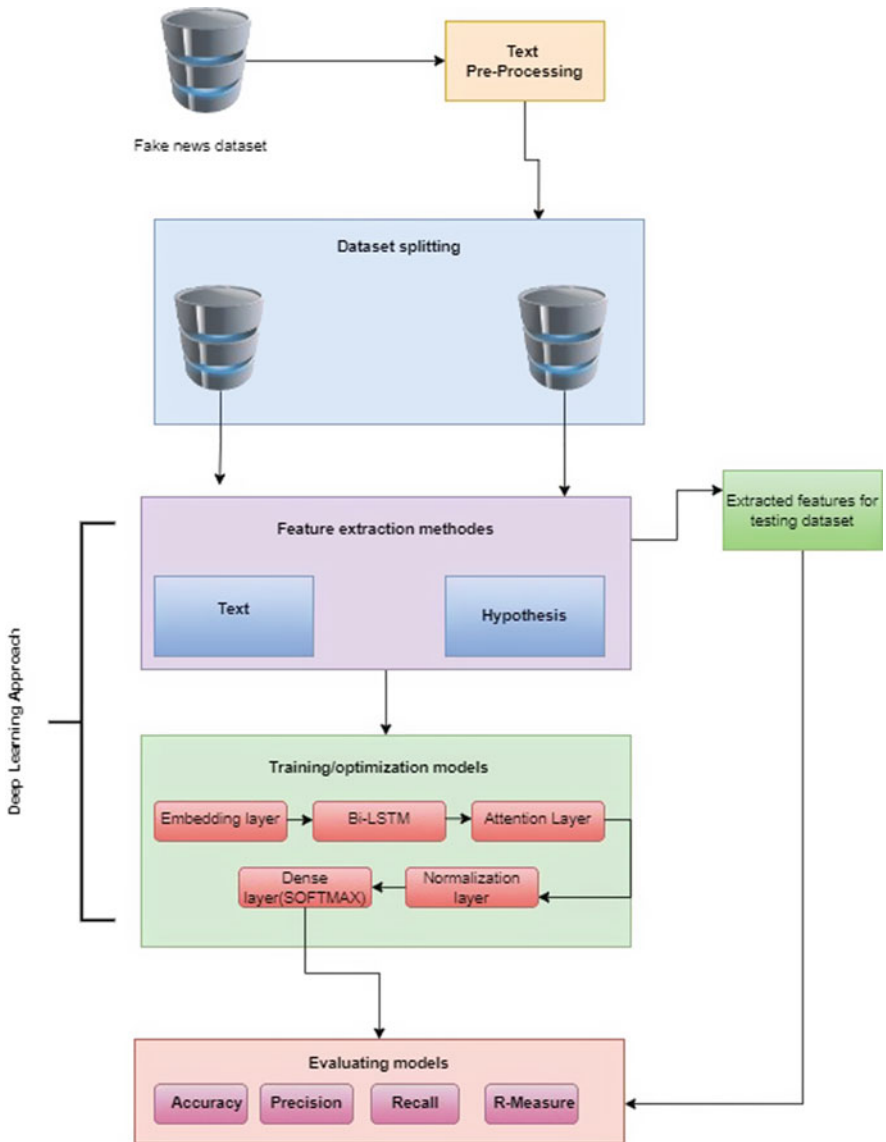


Fig. 1 Architecture of model

our problem statement. Stop words are often used words that are taken out of the text since they bring no value to the analysis. These words carry less or no meaning. If necessary, words are stemmed or reduced to their root/base form. Lemmatize the word while ensuring that it retains its meaning. Then split the data into train, validate, and test datasets. In the next step, relevant features are extracted, which



have most impact on output. In the figure, the next step is model layer architecture in which each word is turned into a fixed length vector of a specific size using the embedding layer. The method of making any neural network to have sequence information in both ways backward (future to past) or forward (ahead to future) is known as bidirectional long short-term memory (BI-LSTM). In bidirectional, our input flows in two directions, making a BI-LSTM different from the regular LSTM. Attention layer in neural networks is a technique that mimics cognitive attention. The effect amplifies specific sections of the input data while detracting from others, with the idea being that the network should pay greater attention to that tiny but significant portion of the data. Normalization is a network layer that allows each layer to learn more independently. It is used to make the output of the preceding layers more natural. In normalization, the activations scale the input layer. A dense layer is one that is intimately linked to its previous layer, meaning that each of the layers' neurons is coupled to each of the preceding layers' neurons.

## 6.1 Training Model

After doing cleaning, preprocessing, and sampling, we get data into the desired format. Then it comes to ELMo model that is implemented using TensorFlow and TensorFlow Hub. The pre-trained ELMo model can be imported using TensorFlow Hub to convert the trained set and validated set into ELMo vectors. Now split the data into the training, validation, and testing datasets and set and train the model using the training dataset. Steps used for training the trained textual entailment (TE) recognizer are as follows: [20]

1. In the training dataset, words are tokenized for both the hypothesis and the premise, and the token ID of each word is mapped to the associated multidimensional word. Set the weights of the layer in Fig. 1 as the word embedding matrix and mark them as non-trainable.
2. Set the neural network's settings. The neural network is trained using the training and validation datasets. Calculate the classification report.
3. Test the model on the testing dataset, which consists of sentence pairs after it has been trained on the training and validation datasets. We utilize the same token for testing that we used in STEP 1.
4. Compare the data from training and testing with the final outcome.

Here we describe the algorithm for non-trustworthy COVID-19 news:

- (a) For each reference and trustworthy news combination, we label the trustworthy (hypothesis) as false if the likelihood of contradiction exceeds the chance of entailment, regardless of the neutral component's probability.
- (b) We classify the headline (hypothesis) as legitimate for each reference news item and headline pair if the probability of entailment is larger than the probability of contradiction, regardless of the likelihood of the neutral component.

$$isFake = (TER_{pred}(R_{tok}, T_{tok})). \quad (1)$$

TER stands for the trained textual entailment recognizer, where R-tok and T-tok have tokenized references and trustworthy news items.

## 6.2 Performance Analysis

Different evaluation metrics, including accuracy, F1 score, precision, and recall, are used to assess the model's performance on trained, validation, and test datasets. In Fig. 1, at the last step, model performance is analyzed using these measurements. To gain a notion of how the model performs, the findings are analyzed and compared to the end result to some existing model performance.

The following are the performance parameters that would be used to evaluate a new architecture:

- Accuracy: It is the ratio of the number of correct predictions to the total number of inputs.

$$ACCURACY(A) = \frac{CorrectClassification}{TotalNumberOfInputs}.$$

- Precision: It defines the percentage of positive predictions that were correct.

$$PRECISION(P) = \frac{TP}{TP + FP},$$

where TP is true positive and FP is false positive.

- Recall: It defines what proportion of actual positives was identified correctly.

$$RECALL(R) = \frac{TP}{TP + FN},$$

where TP is true positive and FN is false negative.

- The harmonic mean of accuracy and recall is the F1 score. Because it penalizes extreme values, the harmonic mean is frequently utilized instead of a simple average. This statistic is employed when we are trying to discover the best balance of accuracy and recall.

$$F1 - SCORE = 2 * \frac{(P * R)}{(P + R)},$$

where P is the Precision and R is the Recall.

### **6.3 *Hyperparameter Optimization***

Hyperparameters are noteworthy because they have a direct influence on the properties of a particular model and may also be used to modify the model's performance. Hyperparameters are tweaked to improve the efficacy of a particular model. The process is known as "searching" the hyperparameter space for the best values.

## **7 Implication and Future Work**

Our work is mainly on finding trustworthy news using sources of tweets, analysis on outreaching, and impacts of wrong information based on likes and retweets. There is another possibility also to investigate the trustworthiness of news based on world authenticated research data and more advanced method such as image processing and finding information from the image and applying some analysis on that data it can be achieved using NLP (natural language processing).

### **7.1 *Application of Solution***

As earlier, we have to train and validate our model on Twitter-based news dataset so it can predict the trustworthiness of news and guidelines on Twitter. Since our model is a more general one, we can also use it on other social media platforms or any other Web content. Similarly, we can train our model in different contexts such as in trending or spamming of wrong information, it will be more proficient if we can get an effective result and accuracy.

### **7.2 *Future Research Direction***

We discuss the differentiation of non-computing and computing, as well as approval, limits of our methodology, and further ways of findings to this part. We can gain more information on image processing utilizing natural language processing, and model reach will expand from a text-based approach to an image-based method. There are a plenty of additional options in this situation.

## 8 Conclusion

In this chapter, we look at the issue of the COVID-19 recommendation's reliability, and how changing them might lead to dangerous health issues such as psychological health, wrong medical advice, and economic inequity. It has been used to compare the performance of un-optimized ML- and DL-based models for determining trustworthiness for tweets at a pre-viral stage. By extracting ID, URL, title, and user attributes from tweets, the LSTM model has been utilized to get 100 features from the texts and build a hybrid feature. Machine learning models were trained using alternative actual and false news datasets as well as user attributes, whereas deep learning models were developed using hybrid features. The viable amount of features from the hybrid feature set was determined using a population-based optimization approach capable of reducing total features by more than 15%. The acquired findings demonstrate the superiority of the deep-learning-based model for false detection over typical machine learning methods. All current models are outperformed by the provided optimized LSTM model with hybrid characteristics.

## References

1. WHO, Wunderman Thompson, Social media & COVID-19: A global study of digital crisis interaction among Gen Z and millennials, <https://www.who.int/news-room/feature-stories/detail/social-media-covid-19-a-global-study-of-digital-crisis-interaction-among-gen-z-and-millennials>, [Accessed: 29-01-2022] (2021).
2. Ireton, Cheryl, Posetti, Julie, Hyperpartisan News and Articles Detection Using BERT and ELMo, <https://unesdoc.unesco.org/ark:/48223/pf0000265552> (2018).
3. W. H. Bangyal, R. Qasim, Z. Ahmad, H. Dar, L. Rukhsar, Z. Aman, J. Ahmad, et al., Detection of fake news text classification on COVID-19 using deep learning approaches, Computational and Mathematical Methods in Medicine 2021. <https://doi.org/10.1155/2021/5514220>.
4. Francesco Pierri, Stefano Ceri, "False News on Social Media: A Data-Driven Survey Share on", <https://doi.org/10.1145/3377330.3377334>, [Accessed: 05-01-2021] (2019).
5. Liang Wu, Huan Liu, "Tracing Fake-News Footprints: Characterizing Social Media Messages by How They Propagate", <https://doi.org/10.1145/3159652.3159677>, [Accessed: 15-01-2022] (2018).
6. Yasmim Mendes Rocha, Gabriel Acácio de Moura, Gabriel Alves Desidério, Carlos Henrique de Oliveira, "The impact of fake news on social media and its influence on health during the COVID-19 pandemic: a systematic review", <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8502082/>, [Accessed: 05-02-2022] (2021).
7. Hua SHEN, Xinyue LIU, "Detecting Spammers on Twitter Based on Content and Social Interaction", <https://ieeexplore.ieee.org/document/7311917>, [Accessed: 15-01-2022] (2016).
8. J. Górski, G. Gołaszewski, J. Miler, M. Piechówka, "Trustworthiness: safety, security and privacy issues", <https://ieeexplore.ieee.org/document/4511073>, [Accessed: 20-01-2022] (2007).
9. M. V. Octaviano, "Fake News Detection Using Machine Learning", <https://doi.org/10.1145/3485768.3485774>, [Accessed: 01-01-2022] (2021).
10. Hnin Ei Wynne, Zar Zar Wint, "Content Based Fake News Detection Using N-Gram Models", <https://doi.org/10.1145/3366030.3366116>, [Accessed: 05-06-2022] (2019).
11. O. Azeroual, J. Schöpfel, Trustworthy or not? research data on COVID-19 in data repositories, in: Libraries, Digital Information, and COVID, Elsevier, 2021, pp. 169–182. <https://doi.org/10.1016/B978-0-323-88493-8.00027-6>.

12. Thota, A., Tilak., Ahluwalia, S., Lohia, N., Fake news detection: A deep learning approach., <https://scholar.smu.edu/cgi/viewcontent.cgi?article=1036&context=datasciencereview> (2018).
13. D. Sun, Y. Du, W. Xu, M. Y. Zuo, C. Zhang, J. Zhou, Combining online news articles and web search to predict the fluctuation of real estate market in big data context, *Pacific Asia Journal of the Association for Information Systems* 6 (4) (2014) 2. <https://doi.org/10.17705/1pais.06403>.
14. Alnawas, A., Arici, N, Sentiment analysis of Iraqi Arabic dialect on Facebook based on distributed representations of documents., *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* (2019). <https://doi.org/10.1145/3278605>.
15. Srivastava, D.P., Anand, O., Rakshit, A., Assessment, implication, and analysis of online consumer reviews: A literature review., *Pacific Asia Journal of the Association for Information Systems* (2017). <https://doi.org/10.17705/1pais.09203>.
16. Deng, S, Huang, Z.J., Sinha, A.P., Zhao, H., The interaction between microblog sentiment and stock returns: An empirical examination., *MIS Quarterly* (2018). <https://doi.org/10.25300/MISQ/2018/14268>.
17. Lak, P., Turetken, O., The impact of sentiment analysis output on decision outcomes: An empirical evaluation., *AIS Transactions on Human-Computer Interaction* (2017). <https://doi.org/10.17705/1thci.00086>.
18. J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, Detecting rumors from microblogs with recurrent neural networks, [https://ink.library.smu.edu.sg/sis\\_research/4630/](https://ink.library.smu.edu.sg/sis_research/4630/), [Accessed: 02-02-2022] (2016).
19. K. Shu, D. Mahudeswaran, S. Wang, D. Lee and H. Liu, FakeNewsNet: A data repository with news content social context and spatial temporal information for studying fake news on social media, <https://www.liebertpub.com>. <https://doi.org/10.1089/big.2020.0062>, [Accessed: 04-02-2022] (2018).
20. C. Janze and M. Risius, Automatic detection of fake news on social media platforms, <http://www.ttccenter.ir/ArticleFiles/ENARTICLE/3884.pdf>, [Accessed: 05-02-2022] (2017).
21. J. P. Singh, A. Kumar, N. P. Rana, and Y. K. Dwivedi, "Attention-based LSTM network for rumor veracity estimation of tweets, *Information Systems Frontiers* (2020). <https://doi.org/10.1007/s10796-020-10040-5>.
22. Durier F., Vieira R., Garcia A.C., Can Machines Learn to Detect Fake News? A Survey Focused on Social Media, <https://www.researchgate.net/publication/330364905CanMachinesLearnToDetectFakeNewsASurveyFocusedonSocialMedia>, [Accessed: 19-05-2019] (2019).
23. Gerald Ki Wei Huang and Jun Choi Lee, Hyperpartisan News and Articles Detection Using BERT and ELMo, <https://ieeexplore.ieee.org/document/9034917/authors#authors> (2022).
24. J. Brownlee, A Gentle Introduction to Long Short-Term Memory Networks by the Experts, <https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/>, [Accessed: 05-02-2022] (2021).
25. Philander K, Zhong Y., Twitter sentiment analysis: Capturing sentiment from integrated resort tweets, <https://www.sciencedirect.com/science/article/abs/pii/S027843191630007X?via%3Dihub>, [Accessed: 09-02-2022] (2016).
26. Alaei, A.R, Becken, S., Stantic, B., Sentiment analysis in tourism: Capitalizing on big data., *Journal of Travel Research* (2017). <https://doi.org/10.1177/0047287517747753>.
27. Jost, F., Dale, A., Schwebel, S., How positive is "change" in climate change? a sentiment analysis., *Environmental Science & Policy* (2019). <https://doi.org/10.1016/j.envsci.2019.02.007>
28. K. Popat, S. Mukherjee, A. Yates and G. Weikum, DeClarE: Debunking fake news and false claims using evidence-aware deep learning, <https://www.researchgate.net/publication/327743297DeClarEDebunkingFakeNewsandFalseClaimsusingEvidence-AwareDeepLearning>, [Accessed: 02-02-2022] (2018).
29. A. I. F. AI, COVID-19 Open Research Dataset Challenge (CORD-19) An AI challenge with AI2, CZI, MSR, Georgetown, NIH & The White House, <https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge>, [Accessed: 01-02-2022] (2022).

# Detection of Moving Object Using Modified Fuzzy C-Means Clustering from the Complex and Non-stationary Background Scenes



Ravindra Sangale and Ashok Kumar Jetawat

## 1 Introduction

Observation of object tracking in a scene is unquestionably a significant problem in computer vision. In many vision-based systems, it is one of the first stages. HCI, robot visions and ISS, for example, all need the identification. In fixed camera, several approaches for detecting moving objects have been presented and demonstrated to be successful; however, unforeseen elements exist employing a moveable camera [1]. Many approaches for using a camera have been presented, but their usefulness is still debatable. A simple method for clustering a large number of rapidly shifting things is to be used on a regular basis. The act of recognising those items would cause alteration in scenes known as moving object detection. There are two methods for accomplishing this: (1) motion detection and (2) estimate motion. The procedure of identifying altered and unaffected areas from the returned video picture frames appears to be called “change detection,” and the only stationary object is the camera [2]. However, if the time period is short, this strategy is excessively costly, owing to the fact that the earlier clustering effort is not leveraged. If the duration is long, large periods are accessible. Additionally, brute force technique takes into account for static objects, ignoring any information about their motion. This means, for example, that it is impossible to recognise when certain sets are travelling together. Clustering continually, on the other hand, should include not just the objects’ present locations but also their expected motions. As we will see, this gives us a deeper understanding of how datasets of constantly moving objects cluster [3].

---

R. Sangale (✉) · A. K. Jetawat  
PAHER, Udaipur, India  
e-mail: [ravindra.sangale@vit.edu.in](mailto:ravindra.sangale@vit.edu.in)

Whereas lens cameras record the actual light of a picture, event-based cameras acquire the per-pixel brightness asynchronously, rendering typical computer vision algorithms ineffective for interpreting event data [4]. Detecting the moving things is a crucial problem in automation, since it allows a computer to distinguish between moving and immovable items. Various techniques for recognising moving items in a scene have been developed over time. Only a handful of them, however, are appropriate for event-based data. One of the numerous motives is that event data lacks significant visual qualities like colour and texture, as event-based sensors simply capture relative changes in brightness. Other factors include a shortage of training data (which limits the use of deep learning methods), plenty of noise (which is caused by sensor faults owing to technological innovation of event-based sensors) and so on [5].

## 2 Related Works

The amount of computing these approaches take to work is the most important factor limiting their application. The authors of [6] said that their unoptimised MATLAB implementation took 30–60 s each frame and that of [7] took over 2.6 s. Because they employ dense optical fluxes and non-parametric backpropagation (BP) to optimise a Markov random field (MRF), the approach suggested in [8] also takes a lot of computing and cannot run in real time (MRF). As a result, while these techniques may produce impressive outcomes offline, they are worthless in real time unless a computer with a lot of processing power is available. Although the approaches provided in [9] function in real time, they are nevertheless insufficient when further visual inference tasks are frequently done after detection or when platforms with limited computational capability are used. Non-hierarchical and hierarchical techniques are represented, respectively. K-means method divides the items into K clusters with the purpose of minimising some measure relative to the cluster centroids. The Birch approach uses the concepts of a cluster element and a cluster feature to gradually group static objects. These ideas are expanded in our approach. Birch's static data was adjusted without being introduced, but because objects are always moving, the overall information changed over time [10]. [11], who propose and resolve the issue of object clustering based on network distance, has another interesting clustering approach. In its envisioned situation, when items are limited to a geographic network, network distance, which is so exact, usually used for evaluating item similarity. Despite substantial work on static databases, moving object clustering has just a few algorithms. We will now go through each of these in detail. [12] tries to demonstrate that object's motion may group and resultant can compete at any point. However, in 2D space, static clustering approaches appear to produce about 8 times as many clusters using the OC method, and fifteen times as many clusters as those produced using the standard clustering method. Furthermore, the efficiency of I/O is not taken into account in this approach. In [13], they suggest a clustering-based histogram approach. They use the K-center clustering method [14]

for histogram creation, and they use a ‘distance’ function that integrates together position and velocity differences. Histogram maintenance, on the other hand, is inefficient—as noted in the study, a histogram should be recreated if there are too many changes. Because moving object databases often have a significant number of changes at each timestamp, histogram reconstruction occurs frequently; hence strategy may be not practicable. In [15], micro-clustering is used to moving objects, resulting in algorithms that constantly retain cluster bounding boxes. On the other hand, dominates the algorithm’s overall running time, and the quantity is often very large. The items switch up to  $O(n)$  throughout the moving micro-cluster with  $n$  objects, and each change equates to an event [16].

### 3 Proposed Methodology

This section discuss the proposed video frame segmentation and feature fusion techniques. The overall proposed model is given in Fig. 1.

#### Modelling of Moving Objects

To begin, we will create a collection of object tracking, each of which may broadcast their actual position and velocities to a centralised computer. The object communicates updated motion data to a server when the difference between its

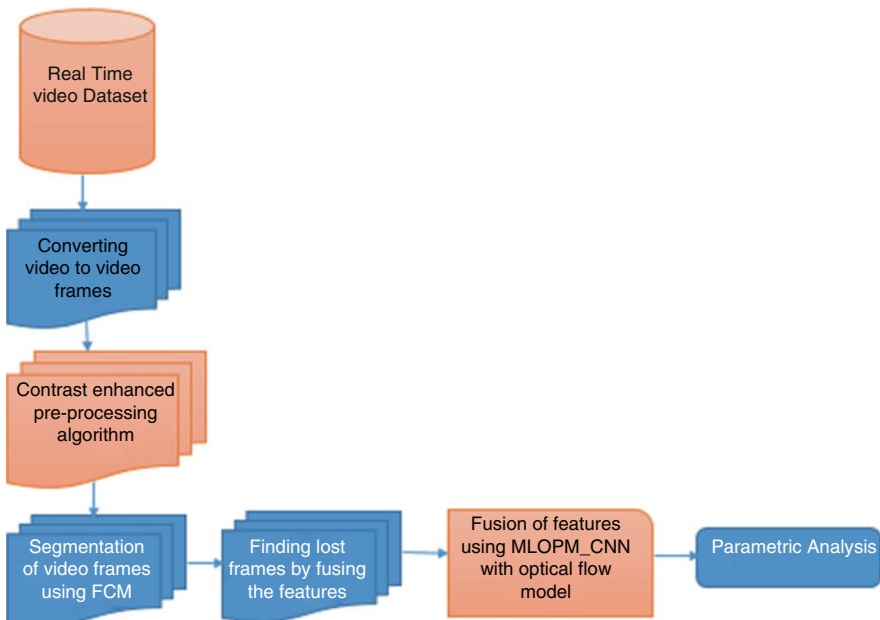


Fig. 1 Proposed overall model





**Fig. 2** (a, b) Original video frame and preprocessed video frame

current location and server-side position exceeds a specified threshold determined by the service to be supported. As time goes on, the gap between the actual position and the server's assumed location widens. As a result, the largest time delay between any two changes to any object is represented by the issue parameter maximum update time ( $U$ ). The  $U$  parameter in the system may be used to force each item to update at least once per  $U$  time units. This is sensible since it is difficult to know if an item is still travelling in the exact direction or has disappeared if it has not connected with the servers in a long period of time.

The video frames are subjected to data preprocessing tasks for cleaning the data and making it suitable to increase the accuracy and efficiency of a deep learning model. Here contrast enhanced preprocessing method is used to suppress the noise over the video frames. The video frame before and after preprocessing is represented in Fig. 2a, b.

### 3.1 Clustering of Video Frames Using MFCM

The primary feature of the MFCM approach is that it gives outstanding results even when the data is overlapping, and it also distributes each data point across several clusters. However, in addition to computing time and accuracy, it necessitates higher number of iterations, and the use of Euclidean distance calculates load as uneven. As a result, encoder-decoder-based CNN can help to lessen this:

Dataset  $Z = \{z_1, z_2, \dots, z_q\}$  with cluster set  $X = \{x_1, x_2, \dots, x_p\}$  and membership

$$\text{set } W = \left\{ w_{kl} \mid 1 \leq k \leq e, 1 \leq l \leq p \right\}$$

It is possible to formulate these three FCM in more detail. The suggested mechanism’s general concept combines IFCM, creating efficient encoder retraining, for example:

$$\begin{aligned} \min : & \sum_{k=1}^e \sum_{l=1}^p w_{kl}^o \| z_l - x_k \|^2 \\ & \sum_{l=1}^e w_{kl} = 1, w_{kl} \geq 0 \end{aligned} \tag{1}$$

Modified FCM is constructed in the equation to eliminate spurious grouping.

$$L_o(W, X) = \sum_{k=1}^e \eta_i \sum_{k=1}^p (1 - u_{kl}^o)^o + \sum_{k=1}^e \sum_{k=1}^p w_{kl}^m \| z_l - z_i \|^2 \tag{2}$$

As a result, improving the equation aids in the updating of the membership matrix along with cluster centres:

$$x_k = \sum_{l=1}^p w_{kl}^o z_l / \sum_{l=1}^p w_{kl} \tag{3}$$

Membership matrix

$$w_{kl} = \left( 1 + \left( \frac{e_{kl}}{\eta_k} \right)^{-1/(o-1)} \right)^{-1} \tag{4}$$

The separation seen between cluster and the membership matrix is indicated by ekl in the preceding equation.

**Function Parameter**

Because FCM has a substantial distance disadvantage, the function parameter is introduced in this section to measure the difference across instances and CC for improved clustering. Improvised FCM treats each instance as a multidimensional array for recording correlation across multiple modes. Furthermore, the model must be trained prior to the implementation of FCM-optimised encoder-decoder, developed in the subsequent sections to train the model.

**Computational Model**

For pre-training the parameters, the CNN is utilised as a core component in the computational model, which is time-consuming and computational. We have produced a new version that saves time and money while maintaining the same quality of results; NN is designed with input:  $Z \in T^{K_1} \times K_2 \cdots \times K_P$ , and the re-creation of the same is denoted by  $Z \in \bar{T}^{K_1} \times K_2 \cdots \times K_P$ .

$$\text{hidden\_layer}_{l_1, \dots, l_p} = \text{enc}(\psi) \left( \sum_{k_1, \dots, k_p}^{K_1, \dots, P} d_{l_1, \dots, l_p}^{(1)} + Y_{\alpha k_1, \dots, p}^{(1)} Z_{k_1, \dots, k_p} \right) \quad (5)$$

$$\text{output\_layer}_{k_1, \dots, k_p} = \text{dec}(\psi) \left( \sum_{l_1, \dots, l_o}^{L_1, \dots, L_p} d_{k_1, \dots, k_p}^{(1)} + Y_{\beta l_1, \dots, l_o}^{(1)} \text{hid\_layer}_{l_1, \dots, l_o} \right) \quad (6)$$

In the following equation,  $K_1$  denotes number of dimensions, and  $L_1$  denotes hidden layer. The following equation gives the reconstruction goal. The current study’s goal is Eq. (8), which is a reconstruction goal.

$$\begin{aligned} L_{V \text{ encdec}}(\Psi) = & \left[ \frac{1}{o} \sum_{m=1}^o \left( \sum_{s=1}^{K_1 \times \dots \times K_o} \sum_{l_1=1}^{L_1} \dots \sum_{l_o=1}^{L_p} \left( Y_{s l_1, \dots, l_o}^{(2)} \right)^2 \right. \right. \\ & \left. \left. + (0.5(\text{out\_layer}_m - Z_m))^y I(\text{out\_layer}_m - Z_m) \right) \right. \\ & \left. + 0.5\zeta \left( \sum_{r=1}^{L_1 \times \dots \times L_n} \sum_{k_1=1}^{K_1} \dots \sum_{j_p=1}^{K_o} Y_{r k_1, \dots, k_o}^{(1)} \right)^2 \right] \quad (7) \end{aligned}$$

**Fusion of Features Using MLOPM\_CNN**

Our technique is based on MLO pattern matching, a set of template edge pixels  $M = \{\mu_1, \dots, \mu_m\}$  and a set of image edge pixels  $N = \{v_1, \dots, v_n\}$  to formalise the problem. The elements of  $M$  and  $N$  are  $x$  and  $y$  image coordinate vectors. We will call  $pT$  a random variable that describes the template’s location in the picture. While this assumes that the model appears only once in the image, we may impose a threshold on the probability at each point to account for scenarios when the model does not appear or appears several times.

**Map Similarity Measure**

At this set of measurements, we utilise the distance between each template pixel (at the location provided by some  $p = [xy]^T$ ). We use the term ‘distance’ to describe these measurements:  $\overline{D}_1(p), \dots, D_m(p)$ . In general, to pre-compute the image’s distance transform, we can get these distances rapidly for any  $p$ . The probability density functions for the distance variables are combined to generate the likelihood function for  $p$ . As a result, the distance measurements are approximated to be independent. We discovered that method produces correct results because, as indicated in Eq. 1, the correlation between the distances levels off rapidly as the locations get farther apart (1):

$$L(p) = \prod_{i=1}^m f(D_i(p)) \tag{8}$$

This probability function is unaffected by the number of template alterations that are permitted. The locations where the template position transfers the template edges in to picture define it by  $f(D_i(p))$ .

**Frame Extraction**

Each video or animation you see on TV, on your computer, on your phone, on your tablet or even at the theatre is made out of a series of still pictures. These pictures are then repeatedly presented one after the other, fooling our sight into believing the item is moving. The smoother and more fluid the movement appears, the faster the images are played. The majority of movies and television shows are shot at a rate of 24–30 FPS. A normal movie’s total frame count can be in the hundreds of thousands. Capturing the photo one at a time is wasteful. If user wants to extract a sequence and range, a software has been created for this purpose that could capture quite so many video frames as the user likes and save them to picture files automatically.

The classifier is merged with features that are created automatically by using CNN. The advantages of CNN include the whole classifier, layer dataset, where the output loudness is obtained by changing the input accent and this method is considered very difficult. By means of distinguished function, each layer of input is converted into output through a few unique layers. The limitation of this classifier is that it did not encode the article’s orientation and position into its predictions. Convolution is taken as a relaxed operation, e.g. max pool, both forward and backward. For an extensive network, each training process is envisioned towards longer consideration.

In the beginning, CNN is established for developing the CNN creation and for categorising the images in video frames. Later towards the CNN establishment from scratch, this method is improved by the image augmentation approach. Finally, CNN model is pre-trained in order to classify the images, and precision for training and validation data is examined.

**Pool** It is known as pooling layer. Max pooling in CNN is only utilised and usually the pool size is in the dimension of 2 \* 2 along two strides.

**Fully Connected (FC) Layer** Convolution is executed by the fully connected layer. n1 \* n2 is the size configuration, input tensor size is n1 and the output tensor size is n2. 7 \* 7 \* 512 is considered as N1, which is a triplet and n2 is considered as integers.

**Dropout** Enhancement of deep learning mechanism is utilised by the ‘Dropout’ layer. Some quantity is located by linking the node percentage of specific node. Dropout layer percentage varies from 0 to 0.5.

CNN with the convolution layer is followed by ReLU layer, which enhances the CNN’s non-linearity. Pooling comprises the convolution layers having equivalent

number of channel, strides and kernel size. In fact, two  $3 \times 3$  convolution layers and three  $3 \times 3$  convolution kernels are collected, and it is equivalent to only  $5 \times 5$  and  $7 \times 7$  convolution layers correspondingly. Two or three stacked processes work more efficiently than a single large convolutional core. Additionally, numeric parameters have been reduced. Unnecessary convolutional layers insert ReLU layers and are much needed.

The images of input video frames and their respective images of video frame are represented as  $S = (S^{(1)} \dots S^{(N)})$  and  $M = (M^{(1)} \dots M^{(N)})$ , respectively. The main aim is to model and design the maps  $S$  to  $M$  that helps few training data. This is established by the probabilistic technique towards the learnt distribution model that is indicated as:

$$P(n(M, i, w_m) | n(S, i, w_s)) \quad (9)$$

For image size  $I$  of  $w \times w$ , the patch is indicated by  $n(I, i, w)$  and pixel  $i$  is focused. For high  $w_s$ , extraction of higher contextual information is preferred. The functional form  $f$  is provided as:

$$f_i(s) = \sigma(a_i(s)) = P(m_i = 1 | s) \quad (10)$$

where  $f_i$  represents  $i$ th output component's significance and  $a_i$  denotes the sum of input. Logistic utility is given as  $\sigma(x)$  and it is denoted as:

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \quad (11)$$

The softmax output unit is along the side of CNN and it is utilised for multi-class marking. The size of the vector  $S$ , called the softmax output, determines the promotion higher than the potential marker at the  $i$ th pixel under the multi-class marker, conditional on the  $i$ th pixel path to the  $l$ th output unit is considered as condition of re-composition called as:

$$f_{il}(s) = \frac{\exp(a_{il}(S))}{Z} = P(m_i = l | s) \quad (12)$$

where  $f_{il}(s)$  is the probability of prediction where label  $j$  is mapped with pixel  $i$ .

The real-time working of the proposed model is as follows:

Here the input video frame is obtained from MOT20 dataset for detecting the moving objects. Figure 3a, b describes the original video frame, which is taken as input, and the extracted video frame for object detection, respectively.

The advantages and disadvantages of the proposed method are outlined as below:

- First, a huge amount of labelled data are possibly handled by CNN from numerous domains.



**Fig. 3** (a, b) The original video frame, which is taken as input, and the extracted video frame for object detection

- Graphics Processing Unit (GPU) is used for paralleling, to make the process faster. Henceforth, a huge number of pixel is used for extension. To train the data, kernel's size is minimised by the proposed procedure of computational learning.
- Each training data patch is provided with an initiative sigma. The process of optimisation is very difficult due to the huge number of training patches. Binary classifiers remove those difficulties by using minimum patches. Hyperparameters are used to define the sensitivity examination, therefore achieving higher accuracy.
- Even though binary classifier is used for optimisation, the optimisation process is slightly difficult, and only a few alterations in hyperparameters change the sensitivity of the proposed system.

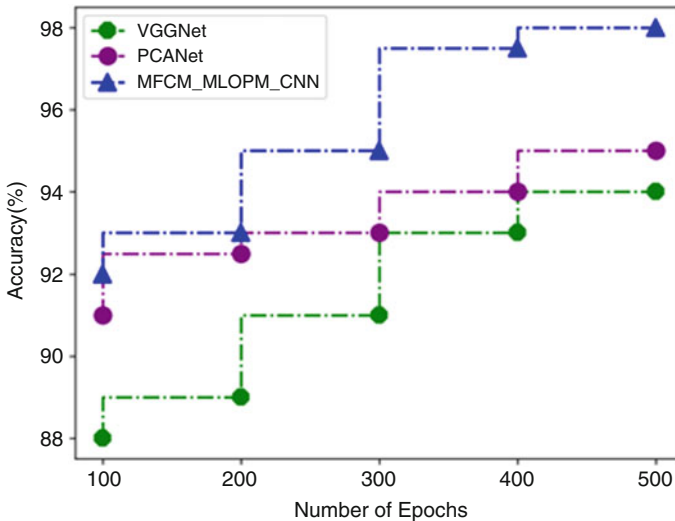
## 4 Performance Examination

Python was used to test the suggested technique on Windows 10. In addition, the OpenCV library was used to recognise and evaluate datasets. In this experiment, we assess the relevance of max, margin training for MFCM\_ MLOPM CNN, and compare objects using different tracking strategies. We tested the suggested FCM\_ MLOPM CNN tracker based on an understanding for a variety of real-time moving pictures in this study.

Table 1 shows comparative examination of proposed and existing technique in terms of accuracy, precision, recall, F-1 score and RMSE. The existing techniques compared are VGGNet and PCANet.

**Table 1** Comparative examination of proposed technique and existing technique

Parameters	VGGNet	PCANet	MFCM_MLOPM_CNN
Accuracy	94	95	98
Precision	88	92	93
Recall	90	93	96
F-1 score	75	80	83
RMSE	55	51	43



**Fig. 4** Comparative examination of accuracy

Figures 4, 5, 6, 7, and 8 showed the comparative examination of proposed and existing technique based on the parameters. Here the proposed technique obtained optimal results in processing the video frames and fusing them.

## 5 Conclusion

This paper proposed a novel technique in real-time video frame processing and feature fusing by multiple object detection based on segmentation and feature fusion techniques. Here the video dataset has been processed and segmented for clustering process using modified fuzzy C-means (MFCM) clustering, and then the clustered video frames have been processed for feature fusion. So feature fusion is carried out using Maximum Likelihood Optimisation Pattern Matching (MLOPM)-based convolutional neural network (CNN) fusion technique. Here the experimental results have been carried out based on accuracy, precision, recall, F-1 score and RMSE. From the comparative examination the proposed technique obtained optimal results in multiple object detection from real-time dataset.

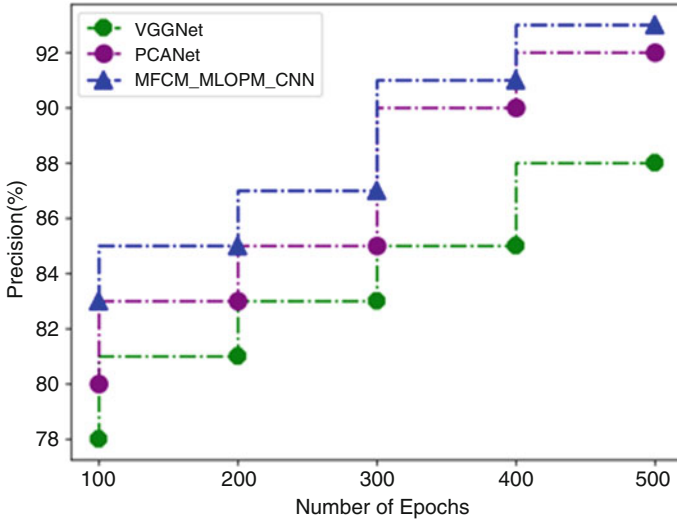


Fig. 5 Comparative examination of precision

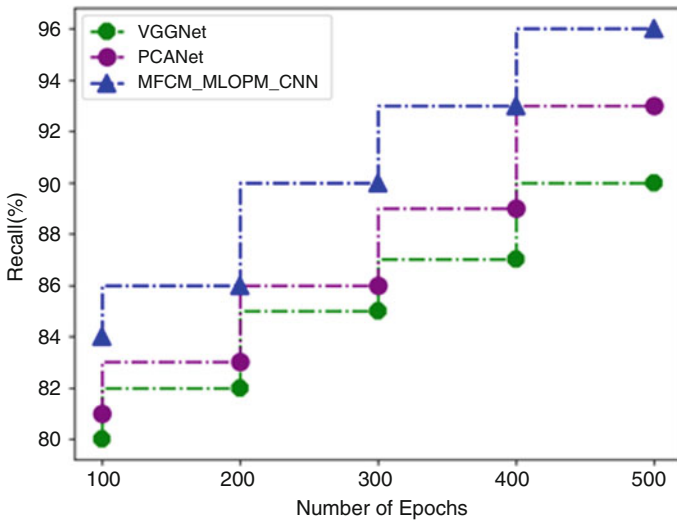


Fig. 6 Comparative examination of recall



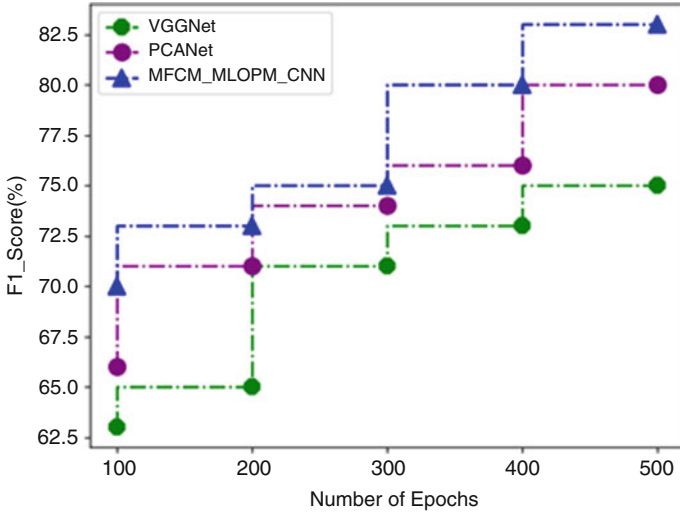


Fig. 7 Comparative examination of F-1 score

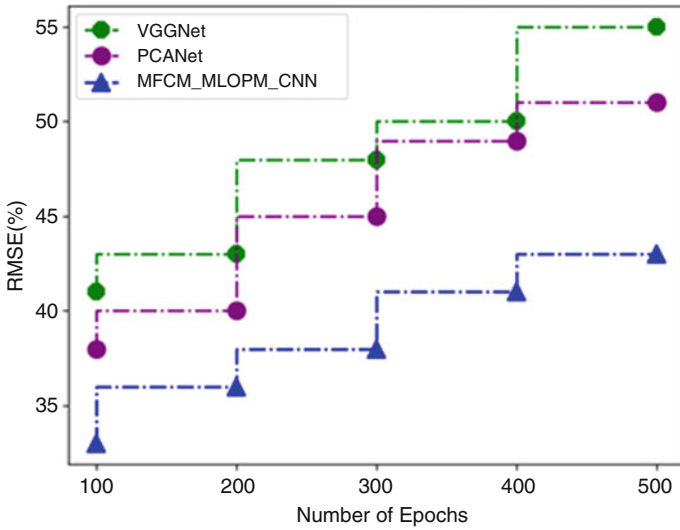


Fig. 8 Comparative examination of RMSE

## References

1. Ciaparrone, G., Sánchez, F. L., Tabik, S., Troiano, L., Tagliaferri, R., & Herrera, F. (2020). Deep learning in video multi-object tracking: A survey. *Neurocomputing*, 381, 61–88.
2. Naik, U. P., Rajesh, V., & Kumar, R. (2021, September). Implementation of YOLOv4 Algorithm for Multiple Object Detection in Image and Video Dataset using Deep Learning and Artificial Intelligence for Urban Traffic Video Surveillance Application. In *2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)* (pp. 1–6). IEEE.
3. Kim, B., & Lee, J. (2019). A video-based fire detection using deep learning models. *Applied Sciences*, 9(14), 2862.
4. Duggal, S., Manik, S., & Ghai, M. (2017, November). Amalgamation of video description and multiple object localization using single deep learning model. In *Proceedings of the 9th International Conference on Signal Processing Systems* (pp. 109–115).
5. Wu, D., Sharma, N., & Blumenstein, M. (2017, May). Recent advances in video-based human action recognition using deep learning: A review. In *2017 International Joint Conference on Neural Networks (IJCNN)* (pp. 2865–2872). IEEE.
6. Pradhymna, P., & Shreya, G. P. (2021, August). Graph Neural Network (GNN) in Image and Video Understanding Using Deep Learning for Computer Vision Applications. In *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)* (pp. 1183–1189). IEEE.
7. Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11), 3212–3232.
8. Pérez-Hernández, F., Tabik, S., Lamas, A., Olmos, R., Fujita, H., & Herrera, F. (2020). Object detection binary classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance. *Knowledge-Based Systems*, 194, 105590.
9. Lou, L., Zhang, Q., Liu, C., Sheng, M., Zheng, Y., & Liu, X. (2019, May). Vehicles detection of traffic flow video using deep learning. In *2019 IEEE 8th Data Driven Control and Learning Systems Conference (DDCLS)* (pp. 1012–1017). IEEE.
10. Sreenu, G., & Durai, M. S. (2019). Intelligent video surveillance: a review through deep learning techniques for crowd examination. *Journal of Big Data*, 6(1), 1–27.
11. Mhalla, A., Chateau, T., & Amara, N. E. B. (2019). Spatio-temporal object detection by deep learning: Video-interlacing to improve multi-object tracking. *Image and Vision Computing*, 88, 120–131.
12. Raj, J. R., & Srinivasulu, S. (2021). Object Detection in Live Streaming Video Using Deep Learning Approach. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1020, No. 1, p. 012028). IOP Publishing.
13. Shreyas, E., & Sheth, M. H. (2021, August). 3D Object Detection and Tracking Methods using Deep Learning for Computer Vision Applications. In *2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)* (pp. 735–738). IEEE.
14. Rajjak, S. S. A., & Kureshi, A. K. (2021). Multiple-Object Detection and Segmentation Based on Deep Learning in High-Resolution Video Using Mask-RCNN. *International Journal of Pattern Recognition and Artificial Intelligence*, 2150038.
15. Jadhav, Y., & Farimani, A. B. (2021). Dominant motion identification of multi-particle system using deep learning from video. *arXiv preprint arXiv:2104.12722*.
16. Abdelali, H. A., Derrouz, H., Zennayi, Y., Thami, R. O. H., & Bourzeix, F. (2021). Multiple hypothesis detection and tracking using deep learning for video traffic surveillance. *IEEE Access*.

# Deterrence Pointer for Distributed Denial-of-Service (DDoS) Attack by Utilizing Watchdog Timer and Hybrid Routing Protocol



Sandya J. K., Ashwanth S., Aluri Prameela Manyatha,  
and V. Ceronmani Sharmila

## 1 Introduction

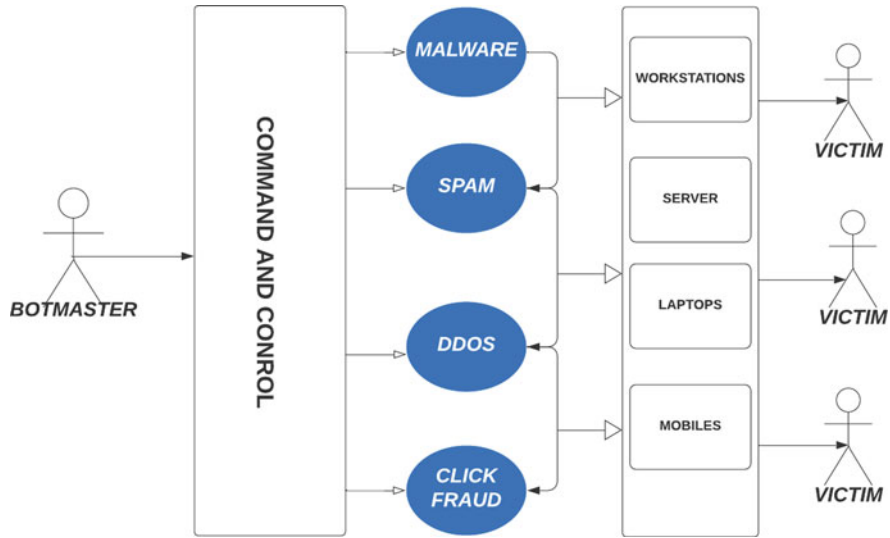
Denial-of-service (DOS) invasions have become recognized by the webbing research society after the initial 1980s. The foremost distributed denial-of-service hack circumstance was informed in the midsummer of 1999, and the majority of the denial-of-service hacks after then have existed and distributed in characteristic. [1]

The DDoS invasions are a malevolent endeavor to sabotage the legitimate access provider of directed cyberspace, service, or webwork by gaining control over the focus or its circumscribing substructure among a inundate of information superhighway traffic. When the target's webwork is focused by the botnet, one by one, the bot transmits invites to the target's address of a website, conceivably generating the server or the computer network to turn into overpowered, succeeding in a denial of service to legitimate traffic. The exceedingly evident manifestation of a DDoS hack is a website or service instantaneously seemly sluggish or inaccessible. On the other hand, after several motives – suchlike an actual impale in traffic – could establish comparable operation difficulties, the advance examination is generally necessary.

DDoS hacks depraved a reticulation by invading nodes put forward in the network, as a result obstructing succeeding traffic to sites and moreover jeopardizing the detriment of classified data. These invasions could close down a site consequently poignant the employment and enterprise straightaway. Computer criminals invade to benefit admittance to the records plus thieve the information of prospects to utilize it for them to concede advantages although the rest extort corporations since attacking their lattice by exacting a redemption to remedy the hack which is

---

Sandya J. K. · Ashwanth S. (✉) · A. P. Manyatha · V. C. Sharmila  
Hindustan Institute of Science and Technology, Information Technology, Chennai, Tamil nadu,  
India



**Fig. 1** Generalised UML diagram of a DDoS attack. (K. S. Bhosle [2])

recognized as cyber-extortion. The expense of authentic traffic being jammed by entering the sites isn't solely of pecuniary worth nevertheless comprises a large number of additional expenses such as client experience, reputation, financial value, repair, and rebuild. Every industry, whether minor or major, requires to scrupulously scheme to guarantee DDoS safeguard for their lattice.

Examination of the webwork, recognition of invaded nodes, and network modification are approaches to guarantee the hacked node is initially detached since the network excludes dissemination and excludes distressing the network function (Fig. 1).

Subsequently, the primary purpose of this paper is to mitigate the distributed denial-of-service attacks by implementing the energy harvesting routing algorithm with the assistance of a watchdog timer and by establishing a hybrid protocol as well as to hinder traffic congestion in the networking architecture in addition to increasing the data transmission speed. Furthermore, the secondary objective is to improve the following networking parameters: throughput, packet delivery ratio, energy consumption, and delay.

The execution of hybrid protocols was surpassed in comparison to alternative protocols because this type controls an enormous amount of traffic, is exceptionally flexible and dependable, compounds the advantages of various forms of the protocol in one protocol, and could be altered as per needs. Hybrid routing admits for expeditious confluence but needs fewer handling authority plus storage as equated to link-state routing. HRP is also utilized to establish optimal webwork terminus routes in addition to describing webwork topology information alterations.

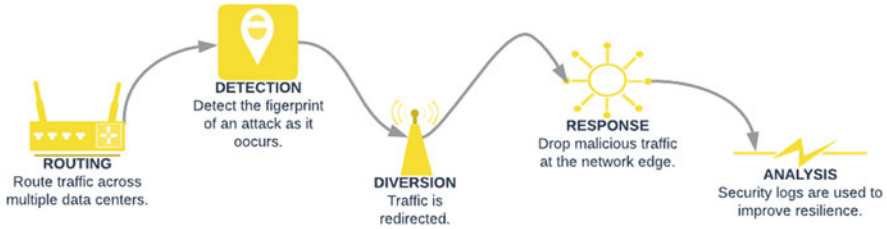


Fig. 2 System overview

The proposed approach in this paper is accomplished by utilizing a watchdog node for the reason that it recovers spontaneously with no individual interference as well as identifies the faults in the program while restarting the computer. In addition, the opted algorithm, i.e., the proposed energy hoarding routing algorithm, enhances the lifespan of sensing element nodes and the quality of service under mercurial traffic charges and power readiness circumstances (Fig. 2).

## 2 Related Works

This segment exhibits a study of recent literature on approaches to mitigating distributed denial-of-service attacks.

One method employed by Shoufeng Cao and Huan Lin [1] main goal is to constant a detailed pattern of authentic traffic, that is, the goal class; next, they tried to find that the consecutive deal is diluted with attack streams by equating the performance to that of the main goal class. They designed a new pattern, that is, the rhythm model used to describe eruption actions and virtualized three methods of DDoS invasion. A more systematic and theoretical analysis is required for the rhythm method with new virtual identification techniques.

A series of recent studies by Zhuotao Liu [3] has indicated the idea of applying mbox and CHM on the target as per the NetFilter Linux Kernel Module. In reference to Intel, the AES-NI library they have utilized AES-128 builds CBC-MAC for calculating because of its readiness and swiftness in these upcoming progressive CPUs. The author displayed the foremost DDoS prevention technique the MiddlePolice. However, this existing research addresses three challenges: Firstly, it demands finite distribution of cloud instead of worldwide network approaches. Secondly, it discourages the drawbacks of the procedures that use seller-specified traffic control policies based on the drive location. Lastly, it points out the bypass vulnerability of the current cloud-based services.

Chenxu Wang emphasizes [4] that at the beginning to decrease the influence upon network dynamics, they introduced a variable for estimating the resemblance among a pair of distributions to determine the difference among sketches in two successive sensing cycles. Then, to detect malevolent servers competently, they applied the

unusual sketch taken out of the previous pair sensing cycles to prevent the overturn computation of the Internet protocol numbers.

The research by Nabil Ali Alrajen [5] has operated with a common stochastic packet indicating method through a wide range of feasible applications that will assist to find IP location and cyberspace constriction as two mean events to indicate its resistance. This approach equates the functioning of identifying exactness of enhanced SVM method, and it utilizes the related variables plus the features, applying database KDDCup99 for analyzing and evaluating, furthermore to utilize all the features of the dataset to enhance detail about the swap in a modern world and to understand which classification method is best in the role of identifying precision by involving all variables of datasets. In the present studies, the multi-threading for bundle handling was constrained. The same will be expelled through SNORT3 in the later work.

Jesús Arturo Pérez-díaz [6] executed an architecture that utilizes an anomaly intrusion detection system to identify variations through a certain framework of machine learning models. They implemented a real-time model based on Mininet VM over VirtualBox and an open network operating system controller. A new approach is therefore needed for the sake of performance; a particular method of shifting to IDS over IPS will be inserted in the future.

Aqeel Sahil et al. [7] designed a novel technique named CS-DDoS to identify and mitigate Dos TCP attacks. CS-DDoS detects attacks precisely utilizing least squares support vector machines. The accuracy holds up to 97% along with a 0.89 kappa coefficient for a lone attack, whereas it is 94% and 0.9 kappa coefficient for the several attacks. The production utilizing K-fold cross-validation enhances the safety of the accounts, minimizes bandwidth usage, and reduces resource depletion. Later on, they want to detect the attack that fulfills the threshold limit along with conquering the challenge of phishing eventually.

Wangdong Jiang et al. [8] have analyzed a well-known prevention and mitigation technique that utilizes a proactive shifting method with the assistance of Bayes' hypothesis to work out restrictive genuine likelihood based on their success and failures. Furthermore, they have concluded that sign-based sensing systems depend on respect to marks that separate typical traffic from a malignant one.

Several studies by Hongbin Luo [9] show the outline, execution, and analysis of a framework for realized DDoS assaults to distinguish the assault D-PID. In their approach, PIDs are utilized as domain routing stuff, and this D-PID alters the paths of interdomain to avoid flooding attacks. This framework also saves the time of arranging PIDs and is productive. To estimate the expenses of initiating these attacks, they have carried out wide simulations and general expenses of D-PID. Also, they reported that D-PID considerably charges more during the attack initiation and also the upgrade value is less compared to the existing IP prefixes.

Sushmita Chakraborty et al. [10] concluded that DDoS is enhancing a large part of a persistent threat operation and the point of attack mechanism has increased. Furthermore, many vulnerabilities like the distributed and unevenly distributed structure of the Internet framework, industry procedures, retreat policies, and revert on funding have decreased the importance of ISPs in removing DDoS fully. On

the other hand, DDoS defense is itself enlarging as a modern business. In these conditions, it looks unfeasible to entirely remove DDoS from the band.

Previous research by Rajat Saxena and Somnath Dey [11] showed that third-party auditors utilize the Weibull probability distribution technique. In this approach, the IDS tools (Snort) are provided to the virtual nodes which are later customized with probability distribution techniques, and when exploited, cautions are initiated by these nodes regarding the packet flooding. Furthermore, the notifications get transformed into basic probability assessments that will be deposited in the database by the front-end server. Precision and susceptibility contrasts are weighed up with additional methods utilizing false alarm rates. The method was highly successful compared to other methods as it got maximum true positives and true negatives.

In a large number of existing studies by Salma A. Mostafa [12], the broader literature has examined the analysis of several methods for tracking and restraining DDoS attacks, based on artificial intelligence and mathematical methods which are sensible at the Open Systems Interconnection layer model. They get instant identification with great accuracy; along with it, the system is inundated, and the distribution of accuracy diminishes by applying MIB and SVM.

M. Poongodi et al. [13] initiated a method that observes the whole progressive transmission thresholds to combatively shut vigorous communications to avoid the system away from resource usage. Currently, this method can be utilized to notice vicious activity by equating existing information with threshold information in addition to several model variables that are implemented to the reCAPTCHA controller method. Their designed system is matched with another current method to measure the designed system's reCAPTCHA controller's capability. The various variables are applied to show the ability of the method to investigate the overflow of distributed denial-of-service attacks. They substantially proved that the designed reCAPTCHA controller method evidently enhanced the outcome competitively with a present model. Likewise, the unique system in their approach reduces the waiting time and usage of energy calculated by variables correspondingly and raises the PDR's value.

Studies by Ligu Chen and ZhiWei Yanb [14] are well documented; it is also well acknowledged by them as a new way to decrease the DDoS traffic utilizing the ML breakthroughs on the access providing filters of the TLD servers. The same is applicable to large recurrent network servers. This prototype is based upon the spark tool as it executes a 0% final performance review and 4% FNR implying that both performance and precision are satisfied according to the requirement. As the authors note earlier, more work is necessary to develop the characteristics and add a virtual way so the users can use streaming modus and an automatic alert system for traffic control plus detection.

Michał P. Karpowicz [15] researches the execution of an accommodative traffic rate controller intended to mitigate proportional DDoS hacks. The study indicates how to broaden the state-of-the-art scheme of webwork protection procedures, by interposing the adaptive procedures regulating the access provider rate policing backgrounds to the discovered webwork functioning circumstances. Additionally,

the arrangement has been intended to agree with the particular webwork access provider restriction.

Over time, an extensive literature has been developed by Onkar Thorat et al. [16] on identifying and categorizing the DDoS attacks that can be conveyed beyond TCP and UDP over the technique called TaxoDaCML – utilizing classification to separate and machine learning to overcome. This technique eradicates false categories in all layers. It cannot be considered conclusive because the shortcomings of this technique are if there is an error in at least one layer the overall result gets affected; hence, it is crucial to maintain a good certainty at all the levels (Table 1).

### **3 Topology Distribution**

Using static routing, minor networks might utilize physically arranged routing columns. Complex problems have larger layouts which can quickly change so that the usual structure of routing columns becomes unworkable. Dynamic routing tackles the issue using developing routing columns manually, in view of data conveyed by routing methods, permitting the network to behave independently by staying away from network breakdowns and obstacles. Dynamic routing controls the online network.

#### ***3.1 Distance Vector Routing***

The distance vector method operates the Bellman-Ford method. The mentioned procedure appoints a particular expense number to every one of the connections among every hub in the webwork. Hubs send data from spot A toward target B through the way that outcomes in the least expensive, for example, the amount of the expenses of the connections between the hubs used. At the point when a hub initially begins, it just is known of its nearby neighbors and the expense engaged with contacting them. Each hub, consistently, ships off to each neighbor hub its present evaluation of all expenses to reach out to all the destinations known by it. The adjoining hubs explore the data and contrast it with what they definitely know; whatever represents an enhancement for what they have, they embed in their table. After some time, every hub in the network finds the consecutive best next jump and maximum expense for all destination hubs. At the point when a network hub goes down, any hubs that pre-owned it as their next hub is disposed of from the passage and passed the information to every single contiguous hub, which follows the same process. At last, every hub in the network gets updated information and finds new ways to the target hub that don't include the failed hub.



**Table 1** Summarization of the main findings of the literature survey

Methods	Parameters used	Conclusion
[2] AL-DDoS detection method, quartile method	Gray-scale map	Future work will combine the rhythm matrix with other online detection methods
[3] mbox, packet filtering	Dp, Thdrop SLR, and $\beta$ parameters	Are fully destination-driven and also address the traffic-bypass vulnerability
[4] Reverse hashing, exponential weighted moving average (EWMA), multiple independent exponential weighted moving average (MIEWMA), hidden semi-Markov model (HsMM)	Sketch's parameter, Bloom filter's parameter, and MIEWMA's parameters	Propose a novel variant of Hellinger distance to calculate the divergence between sketches in two consecutive detection cycles
[5] Spoofing, switch-based, SVM	Trust-based Adversarial Scanner Delaying (TASD)	Lessening false positives without decreasing false negatives and opens a few ways of development that could additionally enhance the present systems of intrusion prevention
[6] Deep learning, reinforcement learning algorithm	Optimal training parameters, traffic flow parameter, and SVM accuracy score	Designed and implemented a modular and flexible security architecture to detect and mitigate LR-DDoS attacks in SDN environments
[7] LS-SVM, naïve Bayes, k-nearest, and multilayer perceptron	Accuracy for multiple attacks. Complexity times. Sixfold cross-validation	The incoming packets are classified to determine the behavior of the source within a time frame, in order to discover whether the sources are associated with a genuine client or an attacker.
[8] Source address validity enforcement protocol, Hop-count filtering, route-based packet filtering	Bandwidth depletion, protocol exploited, Fraggle	A description of the attacks and their key features is provided the advantages and disadvantages of different defense mechanisms
[9] IP traceback-based method	Empirical cumulative density function, datacenter trace, border routers	A dynamic framework called D-PID is evaluated PIDs are modified for interdomain paths according to order As a preventative measure against DDoS flooding

(continued)

Table 1 (continued)

Methods	Parameters used	Conclusion
[10] Flooding, protocol violation attacks, CPU power and service	Internet access, domain name registration, web hosting, USENET service, Co-location	Incomplete distribution and nonuniform architecture are weaknesses Business policies, infrastructures, and other factors ISPs are less interested in eradicating DDoS completely because of privacy policies and return on investment
[11] Key-based CAST-256 algorithm, Weibull distribution algorithm, Hadoop and MapReduce framework	Availability parameter, reliability parameter, median life parameter, traffic behavior, specificity, false alarm rate, detection rate	This solution was provided by an outside party auditor to identify the attack
[12] Bayesian, genetic algorithm, fuzzy logic, K-nearest neighbor technique	Traffic matrix parameter	A framework that dynamically analyzes PIDS and D-PID was evaluated PIDs are modified for interdomain paths according to order DDoS attacks can be prevented by preventing DDoS flood attacks
[13] reCAPTCHA controller, rule metrics, frequency distribution matrix, covariance analyzer	Packet delivery ratio, average latency (AL), detection rate (DR), and energy consumption (EC)	With hybrid mechanisms, security and performance can be improved by preventing and isolating attacks in the future
[14] Confusion matrix, botnet detection, query traffic Classification	Naive Bayes classifier, water torture attack	In order to reduce the amount of DDoS traffic, a novel method was developed and introduced, in which a traffic filter based on machine learning algorithms is applied to major recursive DNS servers on the Internet
[15] Polynomial-based, Traffic policing tuner	Power spectral density, Bode (magnitude) diagram, and adaptive traffic rate.	Analyzes Bounds are introduced during the estimation process and make the process more effective This will strengthen the learning process.
[16] TaxoDaCML (taxonomy-based divide and conquer) approach, meta-heuristic algorithm, KNN algorithm, decision tree algorithm	Precision, recall, F1 score, accuracy, batch size, learning rate	Dividing a larger classification with the help of a taxonomy Divide the problem into seven smaller classifications and solve them one by one Our algorithm uses machine learning methods to detect and classify DDoS attacks, and then we use TaxoDaCML to extract accuracy

### **3.2 *Link-State Routing***

While using link-state methods, a diagrammatic formed by the network is the key information given to every hub. To create its path, every hub blocks the whole network with data about many different hubs it can interface with. Every hub then, at that point, freely gathers this data into a map. Applying this map, every switch autonomously decides the most minimal expense way from itself to every other hub having a standard closest way method like Dijkstra's method. So the outcome is a tree chart established at the present hub, to an extent that they form the root through the tree to some other hub as the lowest expensive way to that hub. This tree then, at that point, builds the routing column, which determines the next successive best jump to get from the present hub to some other hub.

### **3.3 *Path Selection***

Path choosing includes implementing a routing measurement to numerous routes to choose the best way. Most directing methods apply just each network way in turn. Dual-path routing and explicitly equivalent expense multi-way routing methods allow the access of various elective ways. In a wide network, the measure is figured by routing method and can cover data, for example, transfer speed of bandwidth, webwork delay, hop count, way expense, load, and greatest conveyance unit. The routing table saves only the most ideal routes, while link-state datasets might save any remaining data too. If there is a rise of covering or equivalent routes, the algorithm method considers the components in need request to conclude which directs to introduce into the routing table:

1. **Prelude range:** A coordinating route table section with a more extended subnetwork shield is used all of the time as it determines the transfer more precisely.
2. **Metric:** When looking at routes educated through the equivalent routing method, a reduced metric is used mostly. Measurements couldn't measure up among routes gained from various routing methods.
3. **Managerial distance:** When looking at route table sections from various roots, for example, unique routing methods plus non-dynamic design, a lower regulatory distance gives an increased dependable source. Hence, these routes are used often.

## **4 *Route Analytics***

As the IP meshworks have turned into crucial business apparatuses, there has been expanded interest in strategies and techniques to screen the steering stance of organizations. Erroneous routing or directing issues cause unwanted execution debasement and fluttering.

Table-driven (proactive) directing kind of conventions continues with a fresh preparation of objections plus their courses by from time to time circulating directing tables all through the webwork. The fundamental impediments of such protocols are:

1. The corresponding criterion of data for support
2. Delay response on rebuilding and disappointments

On-demand (receptive) routing convention tracks down a course on request by deluging the organization with route request bundles. The principal burdens of such protocols are:

1. Elevated dormancy time in course finding.
2. Immoderate deluging can induce webwork to stop.

A hybrid protocol sort of convention consolidates the benefits of proactive plus receptive directing. The steering is at first settled for certain proactively legitimate courses and furthermore afterward serves the concerned from also commenced hubs through receptive deluging. The determination of either technique needs fate for common cases. With this kind of convention, the decision of proactive and responsive directing relies upon the chain of command level in which a hub dwells. The routing is at first settled for certain proactively legitimate courses and afterward serves the interest from also propelled hubs through receptive deluging on the reduced layers. The primary benefits of the half and half conventions are that they are incredibly adaptable and truly solid. The conventions are effectively adaptable as hybrid webworks are inherent in a design that empowers for a simple combination of new equipment parts. Besides, error distinguishing and investigating are simple.

Hence, the proposed approach is enabled with the assistance of the hybrid protocol.

## **5 Energy Hoarding Algorithm**

A certain way is chosen for the traffic to transmit the data from the root node to the target node, and this process is known as routing. It is carried out across various kinds of networks, along with circuit-switched packet networks, optical mesh networks, circuit-switched data, public switched telephone networks, and PC webworks, for instance, the World Wide Web. During packet switching in the webworks, directing is the more elevated level dynamic that coordinates network packages from the beginning to the end using medium nodes by explicit packet transmission devices. Packet switching plays a major role in networking by delivering the data packets across the network from one node to another. The network gadgets such as hubs, routers, switches, bridges, and firewalls act as the medium for data transmission. Personal computers do transmit packages along with executing directing, even though they are not exceptionally improved and equipped. Routing tables are considered to instruct this process, and they note the paths that lead to several network targets. Routing tables are proposed by the director who

identified them by analyzing traffic and alternatively produced with the help of protocols. It relates to the IP routing table and is appeared differently concerning crossing over. This process wants the addresses to be categorized in the network. To lead a bunch of gadgets or equipment, the categorized address provides a particular routing table path. In major webworks, routing beats vague bridging and is now the prevailing type of tending to on the Internet. Crossing over is still generally utilized inside small webworks.

Energy-harvesting-aware routing algorithm acts on the life span of hubs along with quality beneath traffic burden and energy accessibility conditions. It is an inquiry-based convention that is intended to regard either energy or distance during transmitting packets of data over a computer network.

To plan a successful routing convention, it is important to decide the energy spent by every hub to handle a packet. This comprises the energy expected to communicate, get, or advance the packet of their chosen way. What's more, the hub needs to exhaust energy to tune in for addressing an appearance packet or hang tight for an approaching occasion.

The energy consumption at the nodule  $i$  on link  $e(i, j) \in E$  for handling a packet is provided by:

$$\begin{aligned} E_c^i &= E_l^i + E_{tx}^i + E_{rx}^i + E_{sl}^i \\ &= (t_l^i I_l + (I_{tx} + I_{rx}) \frac{L}{R} + t_{sl}^i I_{sl}) U \end{aligned} \quad (1)$$

Here,  $E_l^i$ ,  $E_{tx}^i$ ,  $E_{rx}^i$ , and  $E_{sl}^i$  represent the energy exhausted over the time course of hearing, forwarding, acquiring, and resting states,  $U$  is the potential difference of the hubs,  $L$  is the distance of the data packet,  $R$  is the ratio in wireless sensor networks, and  $t_{sl}^i$  and  $t_l^i$  are estimated using:

$$t_{sl}^i = BI - SD = 2^{BO} - 2^{SO}, \quad (2)$$

$$t_l^i = BI - (t_{tx}^i + t_{rx}^i + t_{sl}^i + t_{CCA}) \quad (3)$$

To describe the beacon interval (BI) along with superframe duration (SD) of the beacon-enabled node, beacon order (BO) and superframe order (SO) are applied. Presume  $E_{rx}^i = 0$  when  $i$  is a source node and  $E_{tx}^i = 0$  when  $i$  is a target node. It can be written as:

$$E_c^i = \begin{cases} (t_l^i I_l + I_{tx} \frac{L}{R} + t_{sl}^i I_{sl}) U & \text{if } i \text{ is a } Tx \\ (t_l^i I_l + I_{rx} \frac{L}{R} + t_{sl}^i I_{sl}) U & \text{if } i \text{ is a } Rx \end{cases} \quad (4)$$

The residual energy of the nodule  $i$  is then:

$$E_r^i = E_0^i - E_c^i + E_h^i \quad (5)$$

$E_0^i$  and  $E_h^i$  represent the primary energy and the gathered energy of a node  $i$ .

## 5.1 Implementing Energy Hoarding Algorithm Using OTcl

```

proc eaack {} {
    {
        set $nsmessage*data=(message*)packet.data udp;
        set $nsnode = neighbornode_rep;
        set $nsreceiving time = rt(message*data);
        set $nsstransmission time=tt(message*data);
        set $assigner -> node_() (message*data);
        set $marker =1->configure timer
        set $dispatch =(message*)packet.data udp;
        set $accept =(message*)packet.data udp;
        set $ip =id "";
        select $clock =""; }
        if (ack())
        {
            msg*data=(msg*)pkt.data udp;
            sink = new_node_rep;
            allocator= node_->ip;
            public_key = m;
            primary_key =n;
            sign->offer iP(m,n);
            init_sink->address_alloc[0];
            config_(rp)set $ns address alloc;
            msg*data=(message*)pkt.data udp;
        }
        else
        {
            set $ns "attestation unavaible" at "";
        }
        if( threshold > 80)
        {
            select "$ns node_()message*data;
            select clock $ns at "";
            puts "$node_()-malicious_node add_mark . white square"
            update timer ();
        }
        else
        {
            select skip;
        }
        for(nsnode_(0) >= nsnode_(36))
        {
            set timer = rt(message*data) node "" -(tt(message*data)
            node ""+ size of(message*data nsnode "")
            check node_()*address_alloc[];
            select timer;
        }
    };proc attach-CBR-traffic { node sink size interval }
    {
        set ns [Simulator instance]
        set cbr [new Agent/CBR]
        $ns attach-agent $node $cbr
        $cbr set packetSize_ $size
    }
}

```

```

$cbr set interval_ $interval
$ns connect $cbr $sink
return $cbr
}
    
```

## 6 Methodology

To send a piece of information from the root node to the target node, initially, the message has to go through the base station to get encoded into binary, and when the data is in bulk, the message has to get encoded to ASCII values which may result in data loss. Hence, packetization takes place to prevent data loss. Furthermore, MAC and IP addresses have to be provided to convey the message to the accurate target. This process happens through intermediate nodes where each intermediate node contains the sink agent, i.e., the memory storage element. When the DDOS attack takes place in the storage element, the attack happens in different frequencies and that is when the watchdog timer gets enabled. It would identify the attacked node and would enable a watchdog node to replace the former and send the message through the replaced node in order to reduce the data loss and avoid data unavailability. The mentioned process would take place with the assistance of the energy-aware harvesting routing algorithm with the implication of the hybrid protocol. The algorithm functions to detect the nearest node which is at a shorter distance and is free to act as an agent node, but this increases the traffic. Finally, the information reaches the receiver’s base station to get decoded and sent to the receiver (Fig. 3).

## 7 Algorithm for Implementation

Step 1: Start.

Step 2: Current node =  $i$ , receiver nodes of route request control packets (intermediate nodes) =  $a_j$  ( $j=1,2,3,\dots$ ) -> Packetization will be implement.

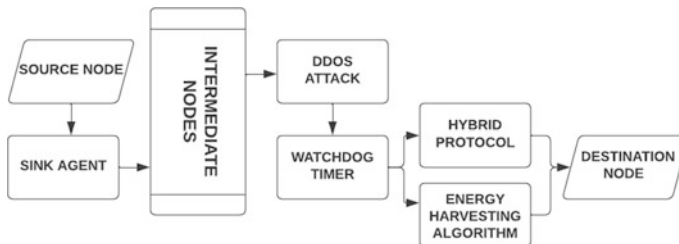


Fig. 3 System Architecture

- Step 3: Current node obtains the presence of its neighbor using the link-state routing protocol.
- Step 4: The watchdog timer is called and implemented for the current attacked node. (If no node is attacked, the information will reach the destination node without enabling the watchdog timer.)
- Step 5: The distance vector routing protocol specifies the links that route request packets are distributed on them.
- Step 6: The energy-aware harvesting algorithm is called.
- Step 7: Wait until all route request control packets are other routes in the destination.
- Step 8: Calculate the evaluation parameters (delay, PDR, throughput, energy consumption).
- Step 9: Choosing the optimal path based on the result of the compatibility.
- Step 10: Providing the final route.
- Step 11: Sending information finished.
- Step 12: End

## 8 Simulations and Analysis

We present to you the values that were set and simulation illustrations that result from the algorithm and analyze the various evaluation parameters. The simulation has been carried out in the NS2 simulator and is programmed using the OTcl (Table 2, Figs. 4, 5, 6, 7, 8, 9 and 10).

**Table 2** Simulation values

Attributes	Values
Channel type	Channel/wireless
Radio propagation model	Propagation/TwoRayGround
Network interface type	Pyh/WirelessPhy
MAC type	Mac/802_11
Interface queue type	Queue/DropTail/PriQueue
Link-layer type	LL
Antenna model	Antenna/OmniAntenna
IFQ length	50000
Number of mobile nodes	36
Routing protocol	EAACK
X dimension of topography	1500
Y dimension of topography	1500
Time of simulation end	10
Threshold	80
Energy	10



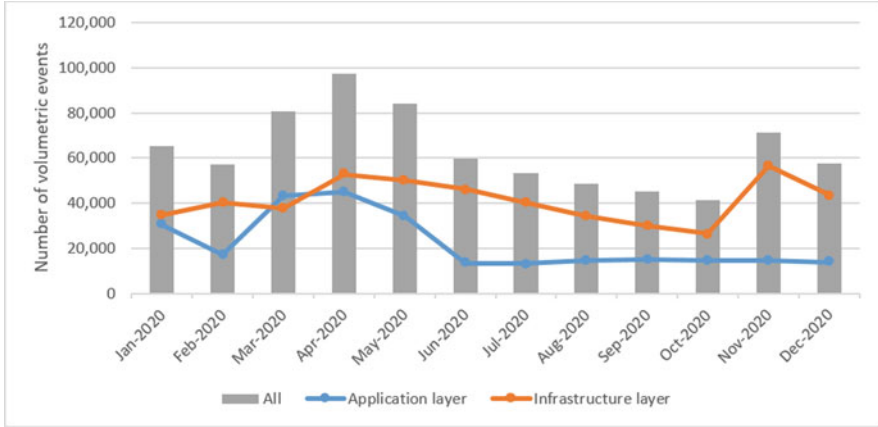


Fig. 4 A cyclical figure of quantitative DDoS hacks detected in 2020

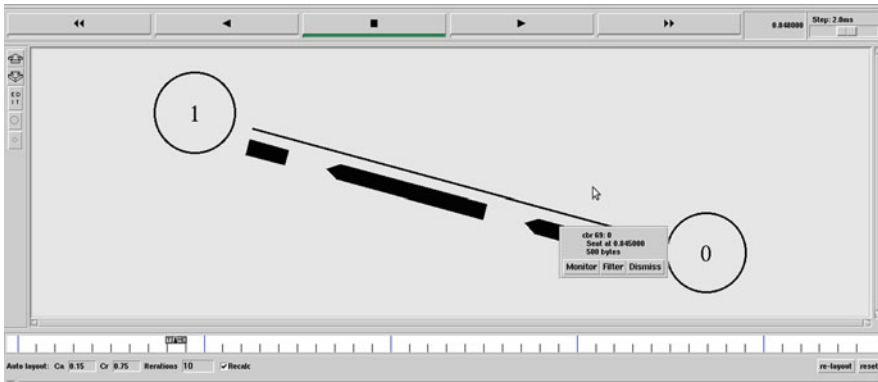


Fig. 5 Node link execution

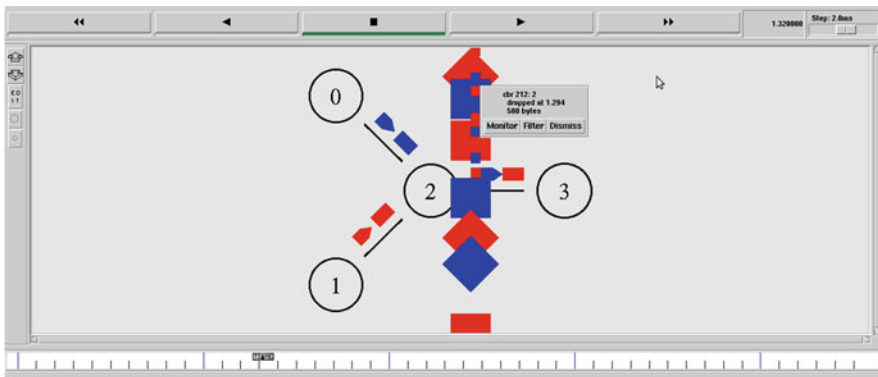
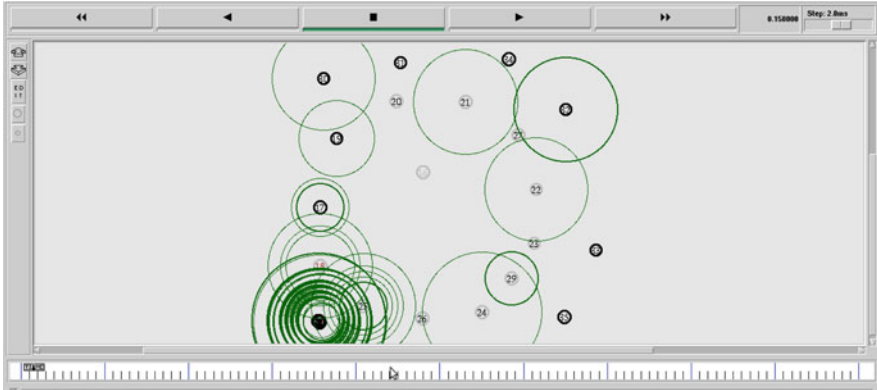
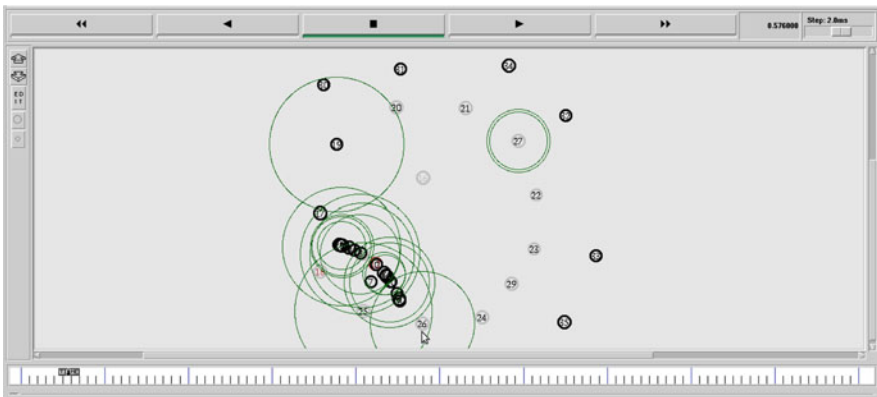


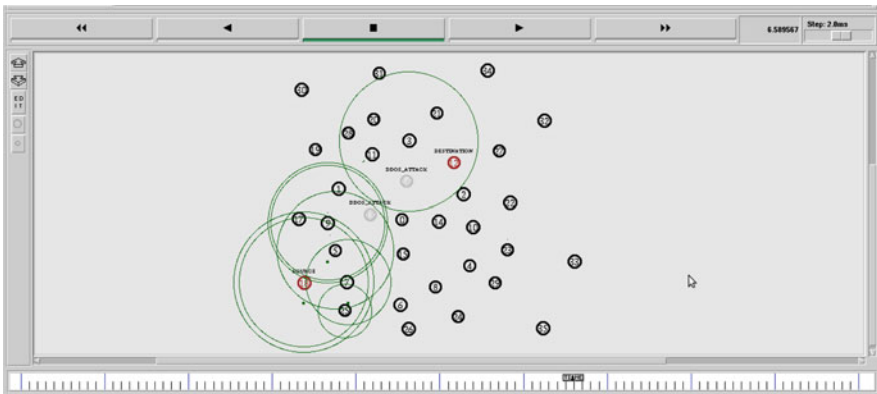
Fig. 6 Topology execution



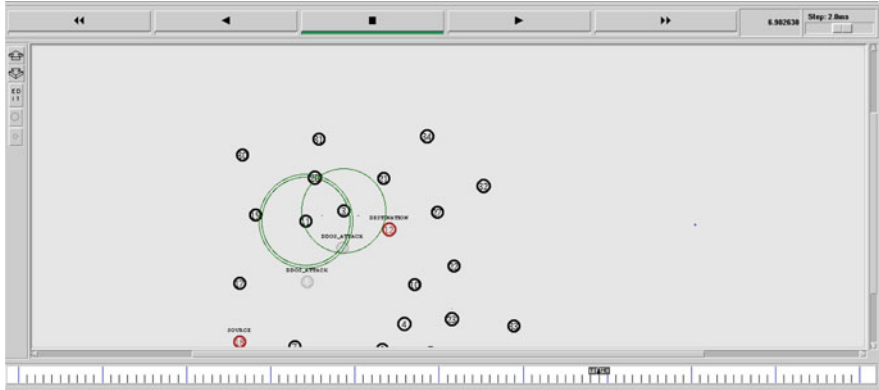
**Fig. 7** Nodes emitting signals for data transmission from root to target node through intermediate nodes



**Fig. 8** Dynamic mobility of nodes for data transmission



**Fig. 9** Source node enhances DSR (Dynamic Source Routing) and transmits data in alternative path to destination



**Fig. 10** Source node enhances DSR (Dynamic Source Routing) and transmits the data in the alternative path to destination

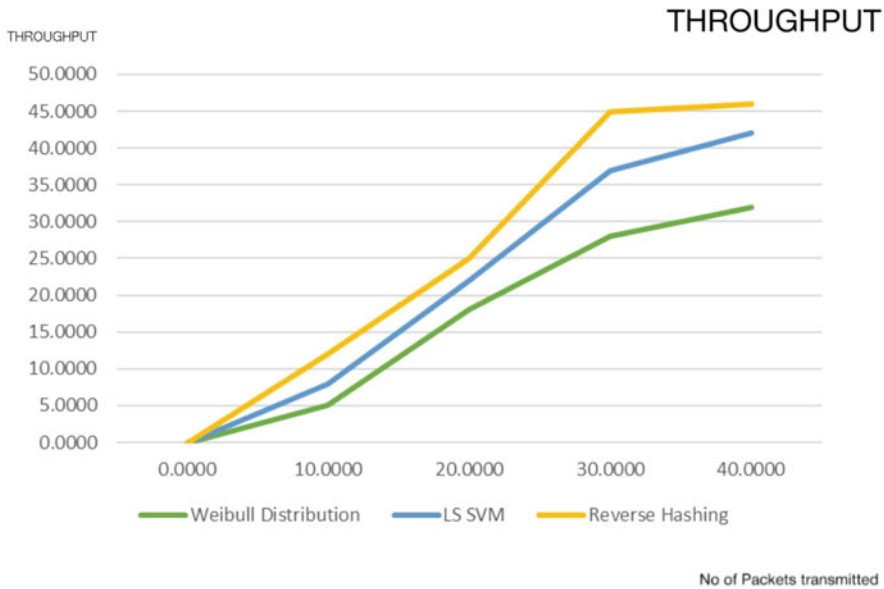
### 8.1 Analysis of the Evaluation Parameters

As per our survey and literature review, a few of the efficient methods to mitigate DDoS attacks used the following approaches, i.e., Weibull distribution, LS-SVM, and reverse hashing. We have compared the values of throughput, PDR, energy consumption, and delay of the mentioned existing approaches. Furthermore, we have taken into consideration the former average of the values and compared it with our proposed method which is using energy hoarding algorithm and hybrid protocol.

#### Throughput

Throughput refers to how much data can be sent from source to destination within the given time limit. It also counts the number of packets that arrived at the destination successfully. For the most part, the throughput value is calculated in bits per second, but it can also be calculated in data per second.

Figure 11 compares the values of throughput of the existing algorithms, and accordingly we can say that the approach that implemented reverse hashing produces the highest throughput. Additionally, we compared the proposed algorithm values of throughput and the average values of the same evaluation parameter of the existing methods in Fig. 12, i.e., the network simulator graph, and can be concluded that the proposed algorithm produces higher throughput comparatively.



**Fig. 11** A comparison between efficient existing approaches’ values of throughput

**Packet Delivery Ratio**

Packet delivery ratio plays a vital factor in performing routing protocol in any network. The same is taken from the overall number of information packets reached out destinations divided by the full data packets sent from sources. If the performance is good, the packet delivery ratio is always high.

Figure 13 compares the values of throughput of the existing algorithms, and accordingly, we can say that the approach that implemented reverse hashing produces the highest packet delivery ratio. Additionally, we compared the proposed algorithm values of packet delivery ratio and the average values of the same evaluation parameter of the existing methods in Fig. 14, i.e., the network simulator graph, and can be concluded that the proposed algorithm produces a higher packet delivery ratio comparatively.

**Energy Consumption**

Energy consumption plays always a major role. So, the results on energy consumption are utilized in routing protocols. While installing a watchdog timer to search out their nearby nodes, the energy cost for every transmission packet should be calculated.

Figure 15 compares the values of throughput of the existing algorithms, and accordingly, we can say that the approach that implemented reverse hashing produces the highest-energy consumption. Additionally, we compared the proposed

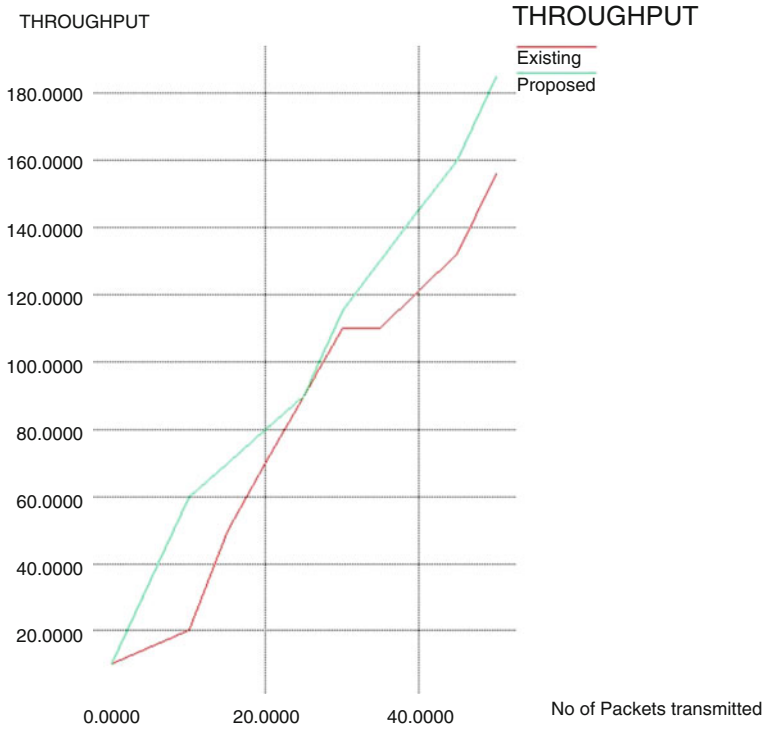


Fig. 12 Graph comparing the existing and the proposed method's values of throughput

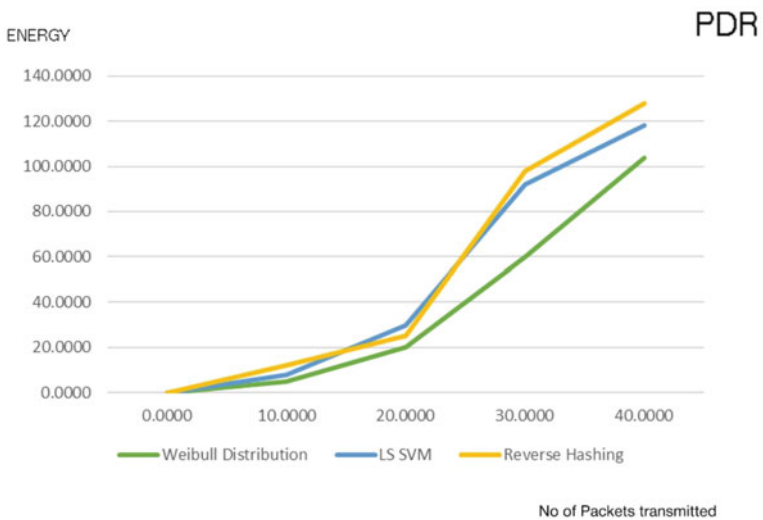


Fig. 13 A comparison between efficient existing approaches' values of PDR

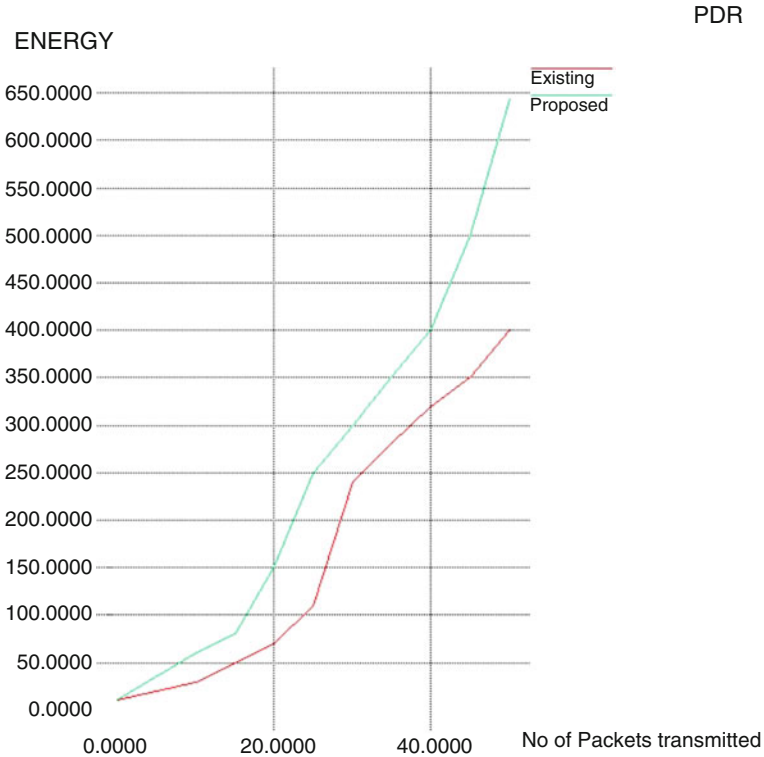


Fig. 14 Graph comparing the existing and the proposed method’s values of PDR

algorithm values of energy consumption and the average values of the same evaluation parameter of the existing methods in Fig. 16, i.e., the network simulator graph, and can be concluded that the proposed algorithm consumes less energy comparatively.

### Delay

Time spent by the information-carrying data packet waiting within the queue before it’s taken for execution is named as a delay. If the destination node is busy performing some heavy task, then there will be an increase in delay. If the destination node is free, then data packets are processed immediately, and these delays will decrease.

Figure 17 compares the values of throughput of the existing algorithms, and accordingly, we can say that the approach that implemented delay produces the highest delay. Additionally, we compared the proposed algorithm values of delay and the average values of the same evaluation parameter of the existing methods in Fig. 18, i.e., the network simulator graph, and can be concluded that the proposed algorithm produces a lower delay comparatively.

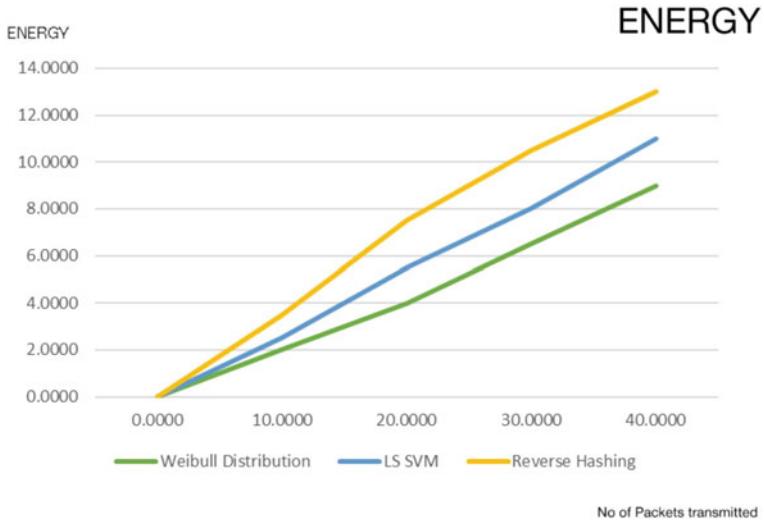


Fig. 15 A comparison between efficient existing approaches' values of throughput

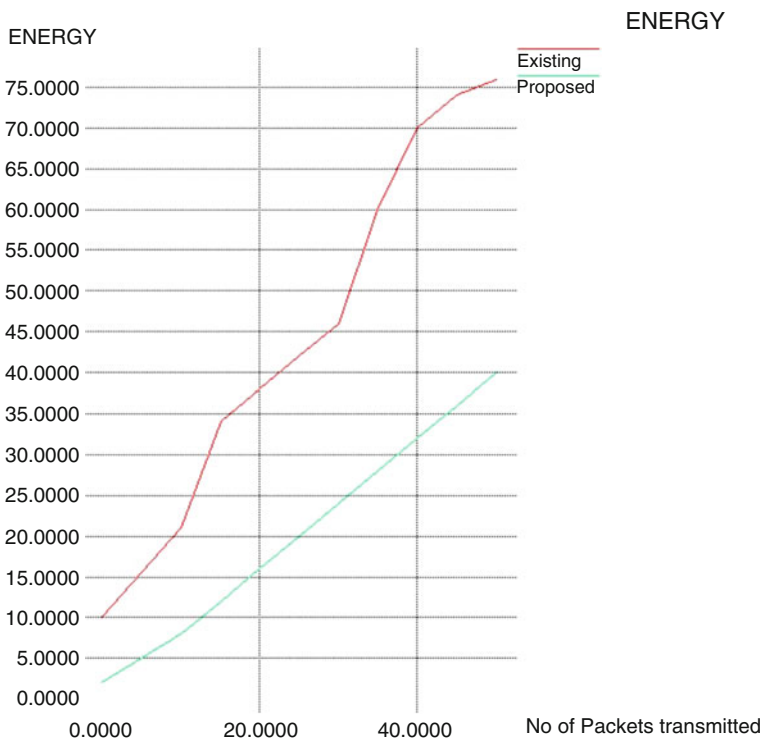


Fig. 16 Graph comparing the existing and the proposed method's values of energy consumption

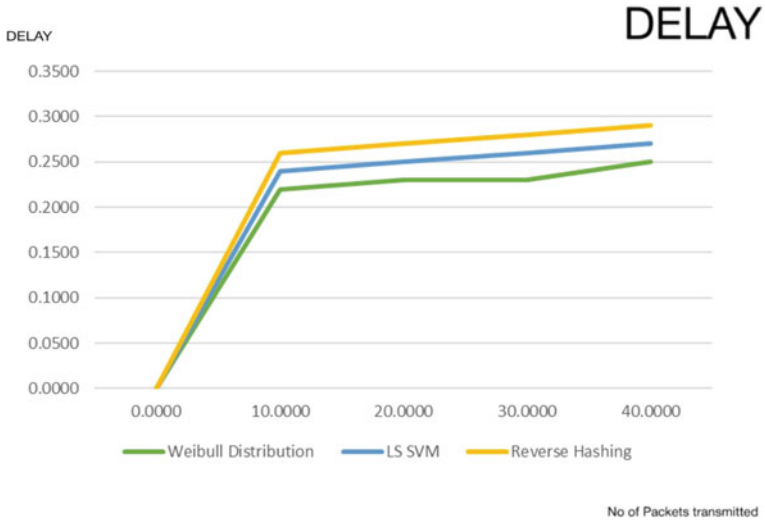


Fig. 17 A comparison between efficient existing approaches' values of throughput

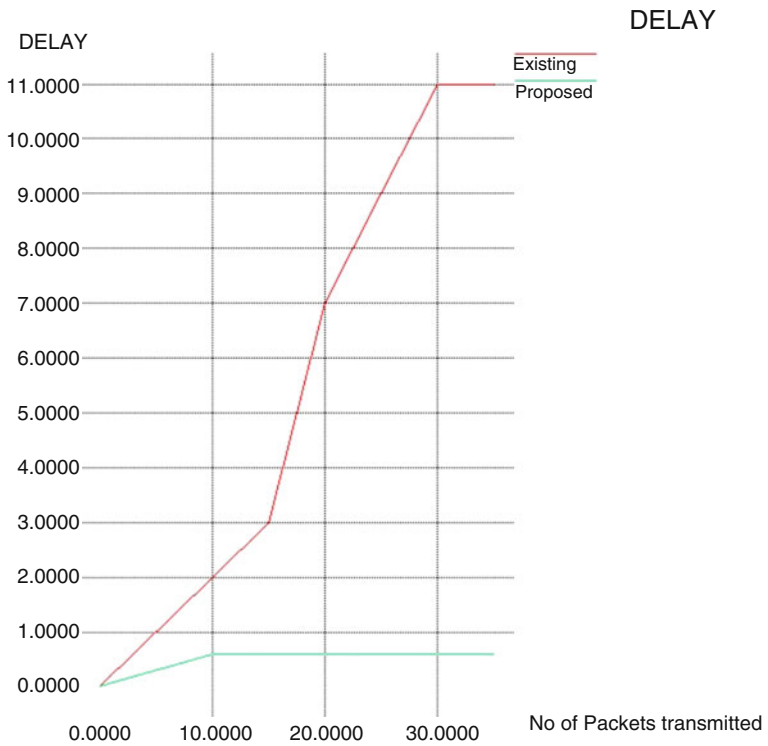


Fig. 18 Graph comparing the existing and the proposed method's values of delay



## 9 Conclusion and Future Work

It can be concluded from the study that DDoS attacks damage the network by attacking existing nodes within the network so that it puts hold on incoming traffic to the web forums and puts you in danger of losing confidential information. This attack can block the webserver and thus directly affect the business. It's important for every business with a web forum to be prepared to forestall DDoS attacks. In consequence, we created and executed a particular and adaptable security design to distinguish and relieve DDOS in software-defined network. The measured quality of the architecture permits each module to effectively supplant other modules without influencing different modules of the design. This proposed approach uses an energy hoarding algorithm and hybrid protocol along with a watchdog timer to detect the DDoS attack, post warnings, and prevent the attack by enabling a reagent node. Data loss is prevented along by reducing energy consumption when the proposed methodology is implemented. Throughput, packet delivery ratio, energy consumption, and delay are analyzed and made efficient as these parameters hinder traffic congestion in the networking architecture in addition to increasing the data transmission speed.

## References

1. Zhenzhong Cao, Huan Lin, Jiayan Wu, Fengyu Wang, and Shoufeng Cao, 2019, "Identifying Application-Layer DDoS Attacks Based on Request Rhythm Matrices", *IEEE Access*, vol. 7, pp. 164480–164491 <https://ieeexplore.ieee.org/abstract/document/8888259>
2. K. S. Bhosle, M. Nenova and G. Iliev, 2018, Generalised UML Diagram of a DDoS Attack,"Detection of Application Layer Ddos Attacks Based on Uml Modelling," 2018 International Conference on Smart City and Emerging Technology (ICSCET), 2018, pp. 1–6 <https://ieeexplore.ieee.org/document/8537355/authors#authors>
3. Zhuotao Liu, Hao Jin, Yih-Chun Hu, and Michael Bailey, 2018, "Practical Proactive DDoS-Attack Mitigation via Endpoint-Driven In-Network Traffic Control", *IEEE/ACM TRANSACTIONS ON NETWORKING*, vol. 26, no. 4, pp. 1948–1961 <https://ieeexplore.ieee.org/document/8418343>
4. Chenxu Wang , Tony T. N. Miu, Xiapu Luo, and Jinhe Wang, 2018, "SkyShield: A Sketch-Based Defense System Against Application Layer DDoS Attacks", *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY*, vol. 13, no. 3, pp. 559–573 <https://ieeexplore.ieee.org/document/8055579>
5. Rana Aubakar, Muhammad Faran Majeed, Amjad Mehmood, Hafsa Maryam, Abdulaziz Aldegeishem, Nabil Ali Alrajien, 2020, "An Effective Mechanism to Mitigate Real-Time DDoS Attack", *IEEE Access*, vol. 8, pp. 126215–126227 <https://ieeexplore.ieee.org/document/9097187>
6. Ismael Amezcua Valdovinos, Kim-kwang Dakai Zhu, Jesús Arturo Pérez-díaz1, Raymond Choo, 2020, "A Flexible SDN-Based Architecture for Identifying and Mitigating Low-Rate DDoS Attacks Using Machine Learning", *IEEE Access*, vol. 8, pp. 155864–155870 <https://ieeexplore.ieee.org/abstract/document/9177002>
7. Mohammed Diykh, Aqeel Sahi1, David Lai, Yan li, 2017, "An Efficient DDoS TCP Flood Attack Detection and Prevention System in a Cloud Environment", *IEEE Access*, vol. 5, pp. 6038–6047 <https://ieeexplore.ieee.org/document/7893798>

8. Wangdong Jiang, Tasnuva Mahjabin, Guang Sun, Yang Xiao, 2017, "A survey of a distributed denial-of-service attack, prevention, and mitigation techniques", *International Journal of Distributed Sensor Networks*, vol. 13, no. 12, pp. 1–33. <https://journals.sagepub.com/doi/full/10.1177/1550147717741463>
9. Athanasios V. Vasilakos, Hongbin Luo, Zhe Chen, and, Jiawei Li, 2017, "Preventing Distributed Denial-of-Service Flooding Attacks with Dynamic Path Identifiers", *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1801–1815. <https://ieeexplore.ieee.org/document/7888484>
10. Dr. Bhawna Sinha, Praveen Kumar, Sushmita Chakraborty, 2019, "A Study On Ddos Attacks, Danger And Its Prevention", *IJRAR*, Vol. 6, no. 2, pp. 10–15 <https://doi.org/10.1729/Journal.20847>
11. Rajat Saxena and Somnath Dey, 2019, "DDoS attack prevention using a collaborative approach for cloud computing", *Cluster Computing*, vol. 23, no.2, pp. 1329–1344 <https://link.springer.com/article/10.1007/s10586-019-02994-2>
12. Aida Mustapha, Mazin, Wafaa Mustafa Abdullah, Salma A. Mostafa, Abed Mohammed, and Bashar Ahmed Khalaf, 2019, "Comprehensive Review of Artificial Intelligence And Statistical Approaches In Distributed Denial of Service Attack And Defense Methods", *IEEE Access*, vol. 7, pp. 51691–51713 <https://ieeexplore.ieee.org/document/8692706>
13. Mounir Hamdi 1, M. Poongodi, Fadi Al-turjman, V. Vijayakumar, 2019, "Intrusion Prevention System For Ddos Attack On Vanet With Recaptcha Controller Using Information Based Metrics", *IEEE Access*, vol. 7, pp. 158481–158491 <https://ieeexplore.ieee.org/document/8859299>
14. Liguo Chen, et al., 2018, "Detection of DNS DDoS Attacks with Random Forest Algorithm on Spark", *Procedia Computer Science*, vol. 134, pp. 310–315. <https://doi.org/10.1016/j.procs.2018.07.177>
15. Michał P. and Karpowicz, 2021, "Adaptive tuning of network traffic policing mechanisms for DDoS attack mitigation systems", *European Journal of Control*, vol. 61, pp. 101–118. <https://doi.org/10.1016/j.ejcon.2021.07.001>
16. Ramchandra Mangrulkar, Nirali Parekh, Omkar Throat, 2021, "TaxoDaCML: Taxonomy based Divide and Conquer using machine learning approach for DDoS attack classification", *International Journal of Information in Management Data Insights*, vol.1, no.2, pp. 7–11. <https://doi.org/10.1016/j.jjimei.2021.100048>

# Modeling Logistic Regression and Neural Network for Stock Selection with BSE 500 – A Comparative Study



Selvan Simon and Hema Date

## 1 Introduction

Among the few studies in modeling machine learning (ML) for long-term financial decisions, neural network or popularly known as artificial neural network (ANN) appeared as the most preferred choice. ANN performed fundamental analysis for stock selection for improving long-term returns [15, 20, 28, 29, 35, 38, 39]. Furthermore, ANN performed ratio analysis for predicting company failure, decline, growth, etc. [1, 4, 12, 26, 40]. They applied ANN for financial performance classifications of companies using financial ratios as inputs, normally for a prediction horizon of one-year.

Applications of logistic regression (Logit) model in ratio analysis for financial performance classification of companies is still scarcer. As a standalone tool, Logit was applied to classify companies as performant or non-performant in stock market [24] and to predict company growth [30]. However, ANN models for companies financial health classifications were compared using traditional parametric Logit model as benchmark [9, 11, 19, 22, 23, 25].

Support vector machine (SVM) is another learning algorithm, often viewed as a competitor for ANN in classification problems. To capture the nonlinear relationships in training sets, SVM uses a kernel function, commonly the Gaussian radial basis function (RBF). SVM classified stocks as high return and normal return for stock selection [10]. SVM was compared in binary classifications for predicting financial risks one year advance [5, 14, 19, 32, 37].

---

S. Simon (✉) · H. Date  
National Institute of Industrial Engineering, Mumbai, India  
e-mail: [hemadate@nitie.ac.in](mailto:hemadate@nitie.ac.in)

## 2 Objectives

A study comparing Logit and ANN for the stock selection with Bombay Stock Exchange (BSE) was missing in the literature. Krishna Kumar, Subramanian, & Rao [20] used an ANN model for the point prediction of stock returns in subsequent year. However, they compared their model using BSE Sensex as benchmark, although BSE 500 constituted the investment universe. This study explored the effectiveness of ML models for stock selection with BSE 500. We developed stock selection models using Logit and ANN that are resilient to the noisy financial data. The models applied fundamental analysis for improving long-term returns. The aim was to identify an optimal model for the stock selection with BSE 500 using comparative studies. It comprised the objectives that involved iterative experiments: (i) optimizing hyper-parameters, (ii) optimizing parameters, (iii) optimizing training, and (iv) comparing models. We used the Orange 3.25 software tool [8] for the model development.

## 3 Data

### 3.1 Predictor Variables

In our study, we included nine financial ratios of companies as predictor variables, which frequently appeared in the literature as model inputs. Based on similar studies on stock selection [7, 17, 18], we list them and their standard acronyms under five categories (see Fig. 1).

Share Price Rationality	<ul style="list-style-type: none"> <li>• Price-to-Earnings ratio</li> <li>• Price-to-Book ratio</li> </ul>	PE PB
Profitability	<ul style="list-style-type: none"> <li>• Return on Equity</li> <li>• Return on Asset</li> </ul>	ROE ROA
Leverage	<ul style="list-style-type: none"> <li>• Debt-to-Equity ratio</li> </ul>	DE
Liquidity	<ul style="list-style-type: none"> <li>• Current Ratio</li> <li>• Quick Ratio</li> </ul>	CR QR
Share Valuation	<ul style="list-style-type: none"> <li>• Book Value per Share</li> <li>• Earnings per Share</li> </ul>	BVS EPS

Fig. 1 Financial ratios used as predictor variables

### 3.2 *Ranking Variable*

The annual Capital Gain Yield (CGY) can be used for ranking stocks. The stocks with higher annual CGY are assigned with higher rank [21]. We will use this ranking variable to derive the response variable, namely, “Class”, later. We computed CGY, the annual percentage increase in stock price, for the subsequent financial year using the following formula:

$$\text{CGY} = \text{rise in stock price}/\text{its original purchase price} \quad (1)$$

### 3.3 *Data Extraction*

We used the ProwessIQ database for accessing data. For our studies with BSE 500, we required a historical data on financial performances of companies spanning a contiguous period of 15 years. Thus, we extracted annual financial ratios of the companies listed on BSE 500 for financial years 2000–2014 and closing prices of their stocks for financial years 2001–2015. We used Microsoft Excel for preparing the datasets. We pre-processed the nine financial ratios for suitable formatting, naming attributes, and rearranging columns. We computed CGY of companies for the subsequent years and concatenated horizontally with their corresponding financial ratios.

### 3.4 *Datasets*

Since our goal is to achieve the maximum possible long-term return, we used the listing of companies in BSE during the entire period of study as the inclusion filter. That is, we included the set of companies that had long standings in stock market for well-established and reliable reporting procedures. It is reasonable to weed out the new entrants and old companies with incomplete data at the outset of the analysis for long-term investments to avoid the inherent high-risk with high-return stocks. Such restriction filters on completeness of data [10, 11, 28, 37, 39] are likely to reduce noise in financial data. Thus, we removed the tuples with missing value in any of their attribute columns. There were 5267 observations on BSE 500 available in 15 datasets with the sample sizes ranging from 243 for 2000–2001 to 433 for 2014–2015 for the experiments with BSE 500. We denote them by  $\mathcal{D}_0, \mathcal{D}_1, \dots, \mathcal{D}_{14}$ . For example,  $\mathcal{D}_{10}$  denotes the dataset of the year 2010–2011.

### 3.5 *Downsampling Training Samples*

Essentially, the fundamental analysis is guided by extreme cases: identifying patterns of companies in the past that performed extremely well (top) to buy stocks and companies that performed extremely bad (bottom) in sell stocks [3, 10, 29]. Therefore, we included only 200 companies with extreme performances as training examples from the dataset of each fiscal year. Based on our similar modeling with BSE 30, we expect the combination of top 50 and bottom 150 companies from BSE 500 will provide the optimal training sample.

### 3.6 *Response Variable*

We appended a binary categorical decision variable, namely, “Class” derived from the ranking variable as follows:

- Label the top 50 companies with high CGY values as Class “A”.
- Label the bottom 150 companies with low CGY values as Class “B”.

At the end of the above downsampling and categorizing, we had 15 sets of training samples obtained from the 15 datasets. We denote them by  $\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_{14}$ . Where,  $\mathcal{T}_j$  is the set of all training samples for the financial year  $[j, j + 1]$  obtained downsampling  $\mathcal{D}_j$ . For example,  $\mathcal{T}_5$  represents the training samples of the financial year 2005–2006.

## 4 **Cross Validation in Ratio Analysis: A Review**

### 4.1 *Set Aside Samples in Ratio Analysis*

To avoid the overfitting problems inherent to ANN and SVM modeling, similar studies in ratio analysis used set aside samples for optimizing their hyper-parameters. They used the simplest cross-validation splitting training sets to set aside some samples for validation and testing [2, 10, 22, 23, 28, 38, 39]. In our earlier study with BSE Sensex, we adopted this method while modeling ANN for stock selection using small sets of samples in the training sets. In this study, for the stock selection with BSE 500, we have a sufficiently large sample for training. Thus, we adopt a more reliable approach using a resampling method, k-fold cross validation.

**K-Fold Cross Validation** In k-fold cross-validation, the entire training set is first partitioned into k equal-size disjoint subsets. Sequentially one subset is used as the holdout set for testing the classifier trained on the remaining (k-1) subsets. The final performance of a classifier is generally evaluated by the average classification accuracy over all k subsets. In this resampling method, each instance of the training

set participates in training for  $k-1$  times and in testing exactly once. Therefore, the cross-validation accuracy is the percentage of the sample classified correctly [9, 32]. The cross-validation procedures can prevent the overfitting problem inherent to ANN, so that the classifiers can accurately predict the unseen data [13, 36].

## 4.2 *K-Fold Cross Validation in Ratio Analysis*

**Case Examples** The following studies adopted  $k$ -fold cross-validations in ratio analysis for binary classification problems in various scenarios:

*Ahn et al.* [1] developed a hybrid ANN model for business failure prediction. The dataset comprised of 1200 healthy and 1200 failed firms drawn randomly from a list of non-financial companies. With the 2400 test cases, they used a 12-fold cross-validation in the test for preventing overfitting.

*J. H. Min & Lee* [22] developed a SVM model for bankruptcy prediction using RBF kernel. The dataset comprised 944 bankrupt and 944 non-bankrupt cases in random order. They employed a grid-search with five-fold cross-validation for parameter optimization, to prevent overfitting.

*Ding et al.* [9] developed a SVM model for financial distress prediction. They found the cross-validation was dependable to prevent overfitting for their median-sized problem with 250 observations in final sample. They adopted the ten-fold cross-validation and grid-search to find optimal hyperparameters for four different kernel functions.

*Yeh, Chi, & Hsu* [37] developed a hybrid SVM model for business failure prediction. The dataset included 38 failed and 76 healthy firms. They adopted RBF kernel for SVM. They performed a grid search and five-fold cross-validation on training data to identify the optimal parameters.

*Cao, Wan, & Wang* [5] developed a hybrid SVM model for financial distress prediction. The dataset comprised of 103 financial distressed and 103 normal companies listed on Chinese stock exchanges. They adopted RBF kernel for SVM. They applied a grid-search to find its optimal parameters by making the five-fold cross-validation accuracy high.

*Telmoudi, Ghourabi, & Limam* [32] developed a hybrid SVM model for bankruptcy prediction. The dataset contained financial ratios of 225 bankrupt and 225 non-bankrupt firms. They adopted RBF kernel for SVM. They performed a grid search with five-fold cross-validation for parameters optimization.

*Huang et al.* [17] developed a hybrid SVM model for bankruptcy prediction. They adopted 2-degree polynomial as kernel for SVM. The datasets pooled together 50 failed and 100 non-failed firms in different ways. The datasets were randomly divided into ten parts and applied ten-fold cross validation for evaluating models.

## 5 Methodology

Develop the proposed ML models for stock selection with BSE 500 and improve them using comparative studies for decision support systems [31, 34]. Beginning of each financial year, retrain the models with more relevant data. Avoid the overfitting of models using ten-fold cross validation. Apply the models to predict 25 top-performing companies listed on BSE 500 for stock selection. These stocks are included in an equally-weighted portfolio. The goal is to win the market index BSE 500 in the subsequent year. To evaluate the models, their portfolios performances were assessed on long-term returns. The models are validated, optimized, and compared for improving prediction results. Figure 2 presents flow chart of the proposed methodology.

### 5.1 Experimental Design

To find an optimal length of the training period, we experimented using observations spanning different historical periods in training sets. Each time, we appended training samples from four years historical records to the training sets of their predecessor models, starting with the recent one year records. As the result, we developed the following four pairs of Logit and ANN models to select stocks for a study period of ten consecutive years:

- ML-I: Logit and ANN models trained using 200 examples from one year historical records and denoted by LOG-I and ANN-I, respectively.
- ML-II: Logit and ANN models trained using 1000 examples from five years historical records and denoted by LOG-II and ANN-II, respectively.
- ML-III: Logit and ANN models trained using 1800 examples from nine years historical records and denoted by LOG-III and ANN-III, respectively.
- ML-IV: Logit and ANN models trained using 2600 examples from thirteen years historical records and denoted by LOG-IV and ANN-IV, respectively.

To avoid conflicting references with our other similar studies, we use the following notations, specific to this study:

- LOG to denote LOG-I, LOG-II, LOG-III, and LOG-IV models, in general.
- ANN to denote ANN-I, ANN-II, ANN-III, and ANN-IV models, in general.
- ML to denote ML-I, ML-II, ML-III, and ML-IV models, in general.

### Training Sets

*Moving Window System* We constructed a series of ten training sets for each of ML-I and ML-II models using the moving window system. Moving window system



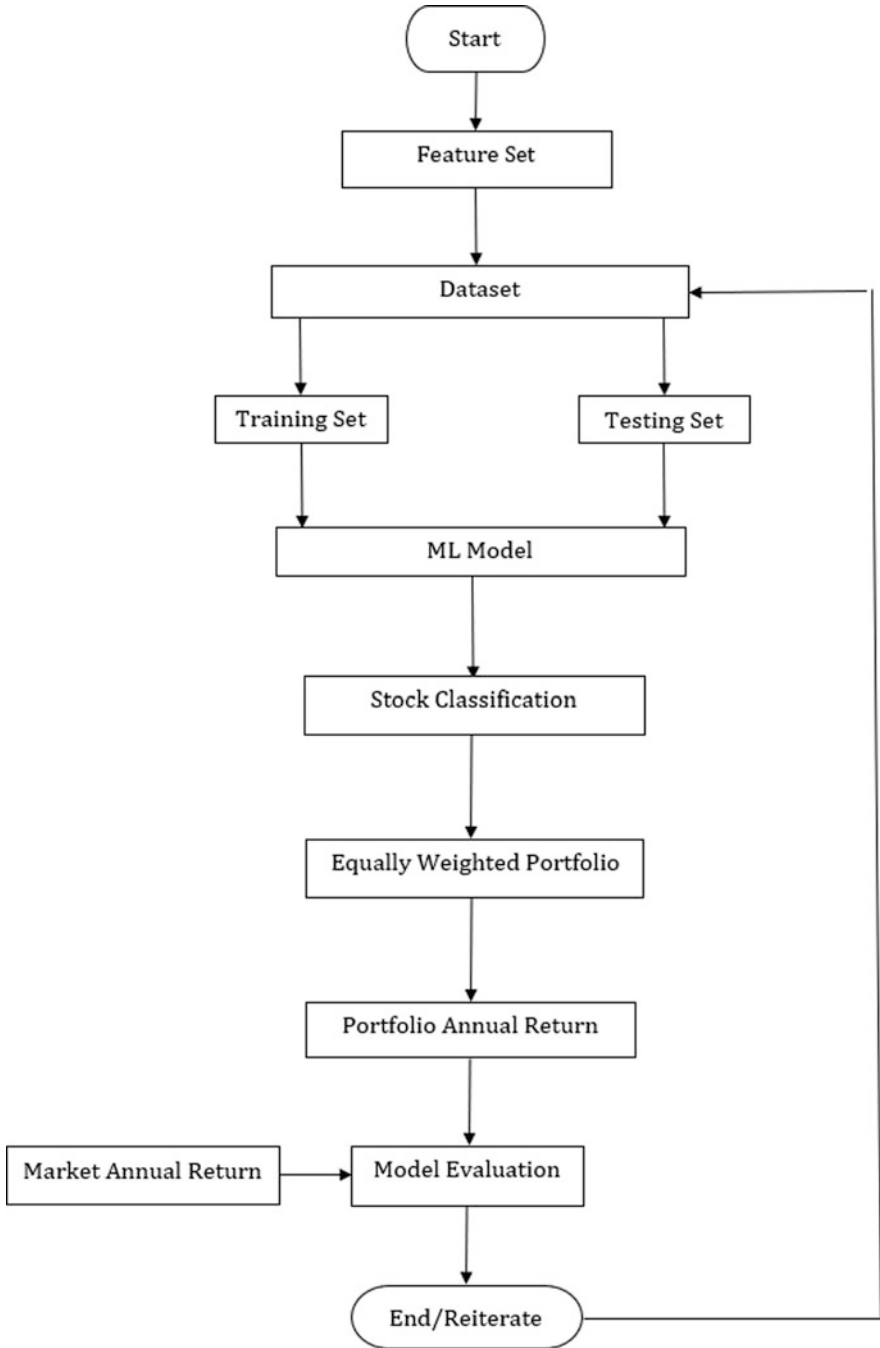


Fig. 2 Flow chart of the proposed methodology

Case/Year	2000-2001	2001-2002	2002-2003	2003-2004	2004-2005	2005-2006	2006-2007	2007-2008	2008-2009	2009-2010	2010-2011	2011-2012	2012-2013	2013-2014	2014-2015
1	Black	Black	Black	Black	Black	Red									
2		Black	Black	Black	Black	Black	Red								
3			Black	Black	Black	Black	Black	Red							
4				Black	Black	Black	Black	Black	Red						
5					Black	Black	Black	Black	Black	Red					
6						Black	Black	Black	Black	Black	Red				
7							Black	Black	Black	Black	Black	Red			
8								Black	Black	Black	Black	Black	Red		
9									Black	Black	Black	Black	Black	Red	
10										Black	Black	Black	Black	Black	Red

Fig. 3 Training set (black) and test set (red) pairs for ML-II models

help in retraining the models using new data drawn from recent environments for fundamental analysis. It can be used to construct and rebalance the portfolios for long-term stock investments [10, 16, 20, 25, 29]. The training sets comprised the recent past one year and five years records in case of ML-I and ML-II models, respectively and the data of its subsequent year formed the corresponding test set. Figure 3 shows the training set and test set pairs for the ML-II model.

*Nested Cross-Validation* We constructed further ten pairs of training sets for ML-III and ML-IV models similar to the outer 10 folds of the nested 10\*10-fold cross-validation used for evaluating trained models [6]. The inner ten-fold cross-validation for searching optimal hyperparameters is explained in Sect. 6. To obtain the training sets for ML-III and ML-IV models, we appended records of eight adjacent years to the training sets of ML-I and ML-II, respectively (Figs. 4 and 5). In both the cases, some samples were borrowed from period occurred later than the target year. However, in cross-sectional analysis, we are interested in the relative strengths of stocks, independent of market performance, and their period of occurrences in the history. Rather, including longer historical data may rather smooth out the predictions from the impact of market scenarios on fundamentals.

**Test Sets** It is not necessary to apply downsampling on test sets [28]. Thus, we included the entire dataset of the corresponding year in the test set of a target year.

Case/Year	2004-2005	2005-2006	2006-2007	2007-2008	2008-2009	2009-2010	2010-2011	2011-2012	2012-2013	2013-2014	2014-2015
1	Black	Red	Black	Black	Black	Black	Black	Black	Black	Black	Black
2	Black	Black	Red	Black	Black	Black	Black	Black	Black	Black	Black
3	Black	Black	Black	Red	Black	Black	Black	Black	Black	Black	Black
4	Black	Black	Black	Black	Red	Black	Black	Black	Black	Black	Black
5	Black	Black	Black	Black	Black	Red	Black	Black	Black	Black	Black
6	Black	Black	Black	Black	Black	Black	Red	Black	Black	Black	Black
7	Black	Black	Black	Black	Black	Black	Black	Red	Black	Black	Black
8	Black	Black	Black	Black	Black	Black	Black	Black	Red	Black	Black
9	Black	Black	Black	Black	Black	Black	Black	Black	Black	Red	Black
10	White	Black	Black	Black	Black	Black	Black	Black	Black	Black	Red

Fig. 4 Training set (black) and test set (red) pairs for ML-III models

Case/Year	2000-2001	2001-2002	2002-2003	2003-2004	2004-2005	2005-2006	2006-2007	2007-2008	2008-2009	2009-2010	2010-2011	2011-2012	2012-2013	2013-2014	2014-2015
1	Black	Black	Black	Black	Black	Red	Black	Black	Black	Black	Black	Black	Black	Black	Black
2	Black	Black	Black	Black	Black	Black	Red	Black	Black	Black	Black	Black	Black	Black	Black
3	Black	Black	Black	Black	Black	Black	Black	Red	Black	Black	Black	Black	Black	Black	Black
4	Black	Black	Black	Black	Black	Black	Black	Black	Red	Black	Black	Black	Black	Black	Black
5	Black	Black	Black	Black	Black	Black	Black	Black	Black	Red	Black	Black	Black	Black	Black
6	Black	Black	Black	Black	Black	Black	Black	Black	Black	Black	Red	Black	Black	Black	Black
7	Black	Black	Black	Black	Black	Black	Black	Black	Black	Black	Black	Red	Black	Black	Black
8	Black	Black	Black	Black	Black	Black	Black	Black	Black	Black	Black	Black	Red	Black	Black
9	Black	Black	Black	Black	Black	Black	Black	Black	Black	Black	Black	Black	Black	Red	Black
10	White	Black	Black	Black	Black	Black	Black	Black	Black	Black	Black	Black	Black	Black	Red

Fig. 5 Training set (black) and test set (red) pairs for ML-IV models

However, we excluded the CGY column in test set as it was used in deriving the response variable. Thus, we obtained a series of ten test sets for the ten target years 2005–2014. We denote them by  $S_5, S_6, \dots, S_{14}$ . Where,  $S_k$  is the set of all test samples of BSE 500 for the financial year  $[k, k + 1]$ . For example,  $S_{10}$  represents the test set of BSE 500 for the financial year 2010–2011. In our semi-supervised learning model, the trained classifier will display its classification outputs in an output column.

**Data Processing** We used the Data Widgets in Orange in our workflows (see Figs. 6 and 14) for the data and file handling tasks (see Appendix).

## 6 Model Development

### 6.1 Model Configurations

Figure 6 shows a workflow for the model development in Orange. We used 40 such separate Orange-workflows for obtaining the 40 pairs of LOG and ANN model stock selection test cases. We used the Logistic Regression and the Neural Network widgets for developing the LOG and ANN models, respectively. We opted for ten-fold cross validation with stratified samples in the Test and Score widget. This inner cross-validation used for finding optimal hyperparameters before training models and the outer cross-validation in Sect. 6.4 is for evaluating the trained models. We simultaneously improved performances of both the classifiers using evaluation results of the Test and Score widget (see Fig. 7). After several iterations with various combinations of hyperparameters, we arrived at the optimal configurations shown in Figs. 8 and 9, which we used for developing all the LOG and ANN models.

### 6.2 Model Estimations

For LOG models, we used the ridge regularization with default cost strength  $C = 1$ . In Orange, the Neural Network widget is built-in with the most common MLP with back-propagation learning algorithm. For ANN models, we used a single hidden layer of 10 neurons with the logistic sigmoid activation functions. This is also the most common ANN architecture in financial forecasting [33]. The sigmoid function takes values between 0 and 1, thus meeting our requirements of probability values. Further settings like Adam solver for weight optimization,  $\alpha$  value for L2 penalty parameter, and maximum number of iterations were made using repeated trial-and-error experiments. These settings were used throughout the experiments to train the forty pairs of classifiers using their respective training sets.

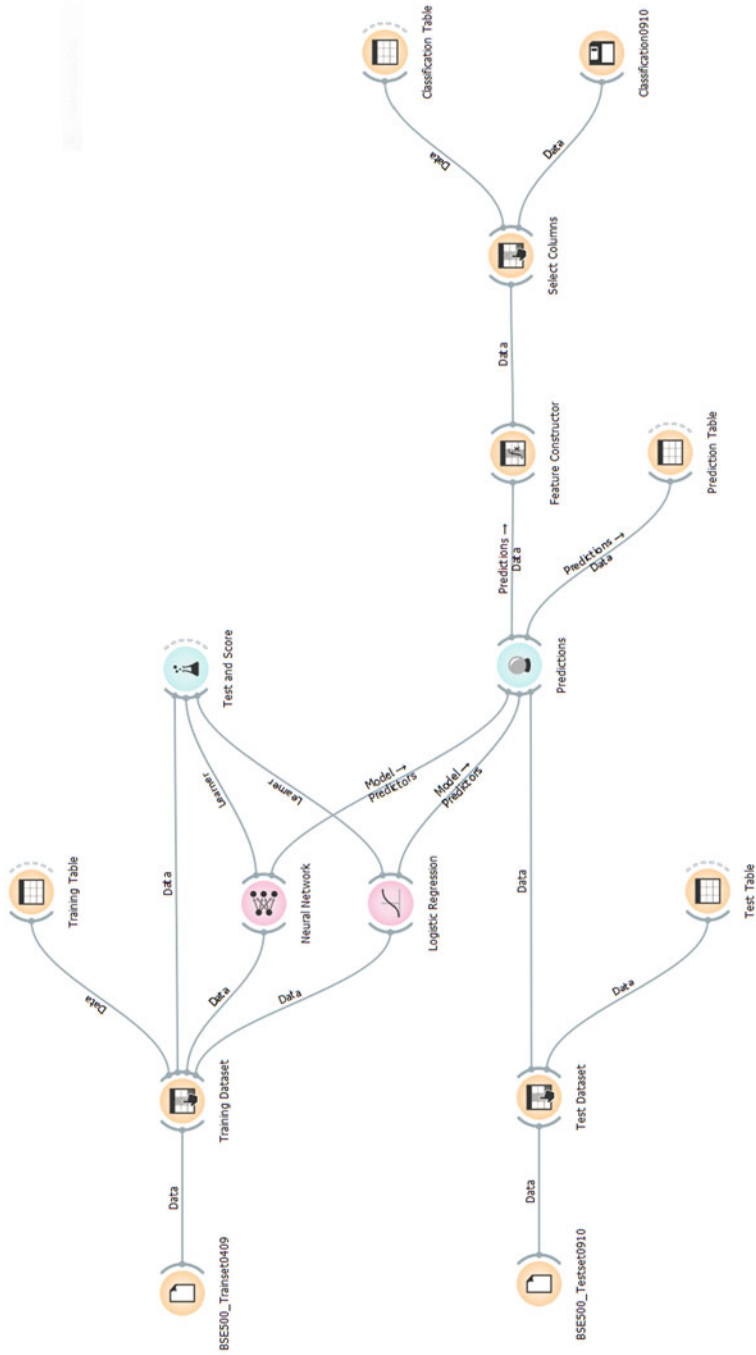


Fig. 6 Workflow for model development

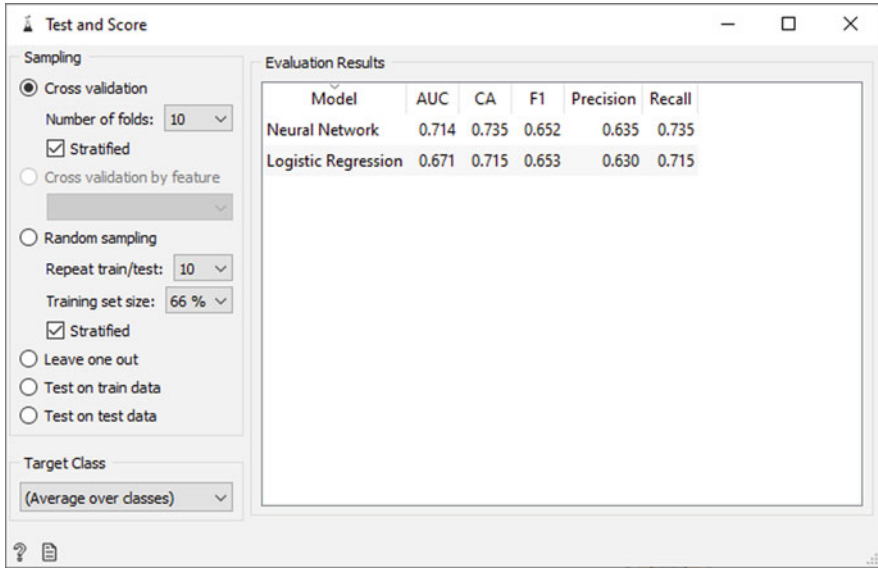
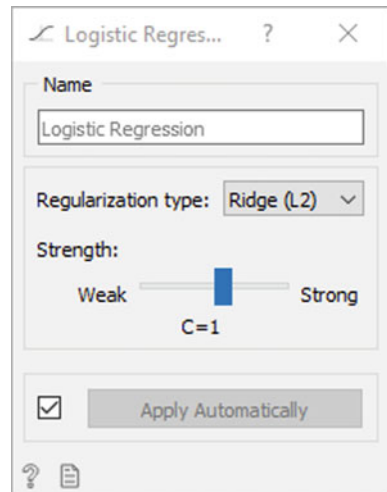


Fig. 7 Cross validation using test and score widget

Fig. 8 Configured logistic regression widget



### 6.3 Model Applications

The trained LOG and ANN classifiers were sent to the Predictions widget to make predictions on their corresponding unseen test sets. In our semi-supervised learning experiments, the trained classifiers displayed the classification results in the output columns for LOG and ANN, respectively. Using show probabilities option, the widget displayed two additional outputs representing the probabilities of being

**Fig. 9** Configured neural network widget

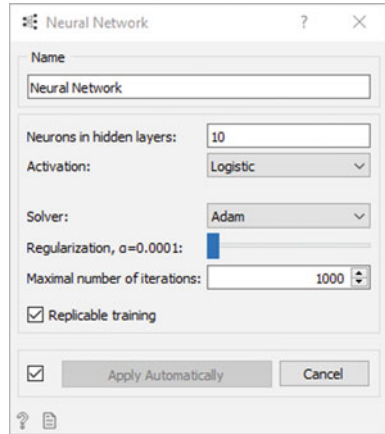


Table with 12 columns: CID, CName, PE, PB, ROA, ROE, DE, Current, and two unlabeled columns for predicted probabilities. The table lists 25 stocks and their corresponding predicted probabilities for Class-A and Class-B.

CID	CName	PE	PB	ROA	ROE	DE	Current	Class-A Prob	Class-B Prob
C1415001	JM India Ltd.	103.72	5.69	3.07	6.34	0.17	0.89	0.49	0.51
C1415002	BK Miles Software Services Ltd.	1180.09	4.73	0.27	0.41	0.12	0.88	0.88	0.12
C1415003	A B India Ltd.	95.02	6.60	1.87	6.71	0.23	0.91	0.65	0.35
C1415004	A C Ltd.	33.22	3.18	8.82	14.37	0.00	0.73	0.40	0.60
C1415005	A I A Engineering Ltd.	16.68	4.00	18.25	23.80	0.00	2.24	1.38	0.62
C1415006	Aarti Drugs Ltd.	5.47	1.27	7.86	26.82	1.51	0.99	0.70	0.30
C1415007	Aarti Industries Ltd.	7.40	1.39	6.41	20.43	1.34	0.83	0.38	0.62
C1415008	Alkan Offshore Ltd.	13.08	1.25	4.42	9.64	0.76	0.71	0.56	0.44
C1415009	Abbott India Ltd.	25.47	4.73	11.45	27.86	0.00	0.88	0.54	0.46
C1415010	Adani Ports & Special Economic Zone Ltd.	35.42	4.16	10.34	25.42	0.96	1.12	1.07	0.93
C1415011	Adventia Ltd.	146.48	3.24	0.31	1.41	1.58	1.09	0.94	0.06
C1415012	Aeris Logistics Ltd.	27.62	1.76	3.38	6.03	0.62	0.79	0.60	0.40
C1415013	Ajanta Pharma Ltd.	15.86	6.56	28.39	49.49	0.24	1.83	1.11	0.89
C1415014	Akzo Nobel India Ltd.	27.86	4.95	6.78	15.41	0.00	0.65	0.37	0.63
C1415015	Alkem Laboratories Ltd.	22.72	8.30	19.47	43.74	0.17	1.00	0.48	0.52
C1415016	Alkermid Bank	4.36	0.45	0.35	11.16	1.12	3.26	3.25	0.01
C1415017	Allergo Logistics Ltd.	45.81	1.57	2.64	4.70	0.36	0.75	0.64	0.36
C1415018	Alicia Industries Ltd.	2.87	0.17	1.19	6.44	2.87	1.27	0.64	0.36
C1415019	Alkermid India Ltd.	24.71	3.00	6.58	26.41	0.00	0.69	0.59	0.41
C1415020	Alkermid T & D India Ltd.	75.02	5.06	2.34	10.88	0.40	1.00	0.82	0.18
C1415021	Amara Raja Batteries Ltd.	18.50	4.89	19.51	36.34	0.06	2.11	1.49	0.61
C1415022	Amrigo Concrete Ltd.	31.36	3.11	10.14	14.16	0.00	1.40	1.08	0.32
C1415023	Amul Auto Ltd.	13.11	0.70	2.48	6.51	1.52	1.38	0.71	0.29
C1415024	Amul Rags Ltd.	19.27	0.43	1.80	2.28	0.33	4.33	1.93	2.40
PL141014	Amul Rags	8.49	0.43	0.19	1.16	1.15	7.13	7.06	0.07

**Fig. 10** Partial view of outputs for 2014–2015 using predictions widget

Class-A and Class-B stocks as predicted by classifier. Figure 10 shows a partial view of predictions by LOG-I and ANN-I for year 2014–2015, displayed using Data Table widget. The Feature Constructor widget allowed to append two modified class columns to the outputs. These are discrete attributes derived from the predicted class probabilities to relabel exactly the top 25 stocks as Class-A. Figure 11 displays logical expressions in Python for defining the new thresholds that separated the top 25 stocks with higher probabilities of being Class-A from the others. Figure 12 shows the predicted 25 Class-A stocks of LOG-I and ANN-I models for target year 2014–2015. The classification outputs were finally exported to Excel worksheets for all further analyses.

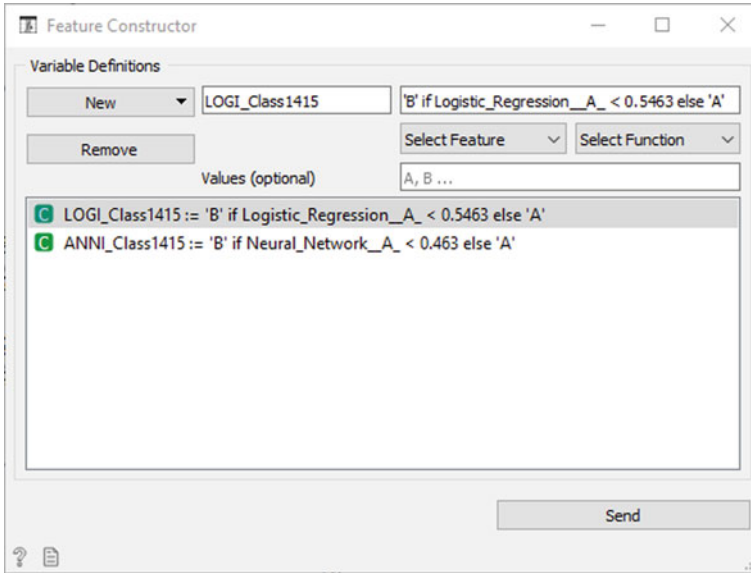


Fig. 11 Redefining class boundaries using feature constructor widget

The image shows two side-by-side screenshots of 'Prediction Table' windows. Both windows display a table with columns: CID, CName, LOGI\_Class1415, and ANNI\_Class1415. The left window shows the first 25 rows of data, with the first row (CID 13) highlighted in blue. The right window shows the next 25 rows of data, with the first row (CID 67) highlighted in blue. Both windows include a sidebar with options like 'Show variable labels (if present)', 'Visualize numeric values', 'Color by instance classes', and 'Select full rows'. A 'Send Automatically' button is visible at the bottom of each window.

Fig. 12 Predicted 25 Class-A stocks of LOG-I and ANN-I models for 2014–15

### 6.4 Model Performance Measures

We used the Microsoft Excel for summarizing and analyzing the experimental results. Since profit is the main motive of stock investments, profitability is the better measure of prediction models than any statistical measures. As a result, stock selection models based on classification or ranking are evaluated primarily using their portfolio returns. Validation of such models are conducted on portfolio basis



for winning the market as benchmark. For comparison, they are further examined for whether one model portfolios outperform the other model portfolios. Since the models aim for long-term investments, their profits are annually assessed for their consistency in wealth accumulation over a long period of ten to fifteen years [2, 10, 19, 20, 25, 28, 29, 38].

**Portfolio Performance Measures** We use the following profitability measures in assessing the model portfolios:

*Annual stock returns* for the sake of simplicity, we used the annual CGY of constituent stocks as their annual returns, ignoring other costs and benefits like brokerage, dividends, etc.

*Annual portfolio returns* the average annual returns of the constituent Class-1 stocks selected using models for the financial year.

*Summary statistics on annual return* of annual portfolio returns like mean, maximum, minimum, standard deviation to summarize overall performances of models over the ten years.

*Cumulative return of portfolios* to compare the accumulated returns of the annually rebalanced buy-and-hold portfolios of models and market at the end of ten years.

*Line chart of cumulative returns* to compare steadiness in the cumulative returns of models at the end of financial years using buy-and-hold with annual rebalancing. The line chart shows the accumulation of wealth for an initial investment of Rs. 100 by the models and market portfolios, during, and at the end of the study period. Ideally investors would like a model that makes money quickly and steadily. However, in reality we may use the model that makes returns with the least variance, making it suitable for long term investments, to avoid losing money (Farmer, 2020).

## 6.5 Model Evaluations

Following similar studies cited above, it is clear that the model performance and generalization ability can be assessed in two levels: the portfolio returns format and the percentage of top performers within the portfolio. Therefore, we evaluate our models primarily on portfolio level and secondarily on stock level using hit rate given by [27]:

$$\text{Hit rate} = \text{TP/P} \quad (2)$$

TP = number of true positive cases

P = number of real positive cases

**Portfolio Level Evaluation** At portfolio level, P, the number of model portfolios constructed for winning the market and a true positive occurs when a model wins the market. A model wins when its portfolio outperforms the benchmark BSE 500

index during that fiscal year. Thus, at portfolio level, the hit rate gives the percentage of winning portfolios among constructed model portfolios. In our study, we evaluate the models on portfolio basis using market as the benchmark.

**Stock Level Evaluation** At stock level, a true positive occurs when a constituent stock of the model portfolio outperforms the benchmark. That is, the annual return of the true positive stock exceeds BSE 500 index return during that fiscal year. In other words, a constituent stock generates true positive if it contributed to its portfolio in winning the market. Thus, at stock level, the hit rate gives the percentage of winning stocks among selected stocks using model. This can be used secondarily to further compare models on directional accuracy.

## 7 Experimental Results

### 7.1 Model Validation

Using stocks selected by LOG and ANN models, we constructed 40 pairs of equally-weighted 25-stocks portfolios for the ten consecutive financial years. Tables 1 and 2 display the LOG-I and ANN-I model portfolios, annual returns of their constituent stocks, and annual portfolio returns for year 2014–2015, respectively. Tables 3 and 4 display annual returns and average annual returns of the model portfolios with BSE 500 index annual returns for the 10 test cases. Each of our model portfolio included exactly 25 stocks selected using a model. Tables 5 and 6 display the number of true positive stocks in the LOG and ANN model portfolios, respectively. The hit rates at portfolio level (see Table 7) validate our proposed stock selection models. Thus, we can extend our evaluation to stock level for the model optimization.

### 7.2 Model Optimization

In this section, we optimize the training period for the ML models. For this purpose, we summarize and compare the ML-I, ML-II, ML-III, and ML-IV models on hit rates at portfolio (see Tables 3 and 4) and stock level (see Tables 5 and 6), respectively (see Tables 8 and 9)

We summarize the experimental results as follows:

- The ML model portfolios outperformed the BSE 500 index in 72 out of the 80 test cases. The overall average annual return of the ML model portfolios was 61.08%, whereas that of BSE 500 was 20.90%.
- The ML-I showed optimal performance among the four ML models:

**Table 1** Portfolio using LOG-I models for 2014–2015

CName	CID	Return
Ajanta Pharma Ltd.	C1415013	205.93
Avanti Feeds Ltd.	C1415036	195.66
Bajaj Auto Ltd.	C1415040	−2.93
Bajaj Corp Ltd.	C1415041	111.49
Castrol India Ltd.	C1415078	52.71
Coal India Ltd.	C1415088	25.88
Colgate-Palmolive (India) Ltd.	C1415089	46.60
Crisil Ltd.	C1415095	64.09
Eclerx Services Ltd.	C1415113	50.14
Eicher Motors Ltd.	C1415115	166.82
H C L Technologies Ltd.	C1415164	40.91
Hero Motocorp Ltd.	C1415170	16.27
Hindustan Unilever Ltd.	C1415177	44.60
I T C Ltd.	C1415191	−7.79
Kaveri Seed Co. Ltd.	C1415239	54.20
Lupin Ltd.	C1415250	114.45
Monsanto India Ltd.	C1415268	90.94
Nestle India Ltd.	C1415283	38.61
Page Industries Ltd.	C1415301	111.83
Strides Shasun Ltd.	C1415360	204.33
Sunteck Realty Ltd.	C1415366	−7.81
Suven Life Sciences Ltd.	C1415368	300.97
Symphony Ltd.	C1415370	252.95
Tata Consultancy Services Ltd.	C1415381	19.68
V S T Industries Ltd.	C1415416	−2.55
<b>Average</b>		<b>87.52</b>

- ML-I models had the highest hit rates with 95% and 59% at portfolio and stock level, respectively.
- Moreover, ML-I had a high average annual return of 63.68% for the ten target years.
- There was no further improvement, rather some deterioration, in the hit rates by adding more training samples spanning longer historical periods.
- The 200 training samples from the recent past year were included in the training sets of ML-II, ML-III, and ML-IV models as well. Therefore, the 200 companies with extreme performances during the recent past year provide the optimal training samples.

**Table 2** Portfolio using ANN-I models for 2014–2015

CName	CID	Return
Ajanta Pharma Ltd.	C1415013	205.93
Alembic Pharmaceuticals Ltd.	C1415015	59.68
Asian Paints Ltd.	C1415031	47.99
Britannia Industries Ltd.	C1415067	155.94
Caplin Point Laboratories Ltd.	C1415075	430.31
Castrol India Ltd.	C1415078	52.71
Coal India Ltd.	C1415088	25.88
Colgate-Palmolive (India) Ltd.	C1415089	46.60
Crisil Ltd.	C1415095	64.09
Dabur India Ltd.	C1415102	47.84
Eclerx Services Ltd.	C1415113	50.14
Emami Ltd.	C1415117	131.08
Hatsun Agro Products Ltd.	C1415168	15.48
Hexaware Technologies Ltd.	C1415171	107.64
Hindustan Unilever Ltd.	C1415177	44.60
I T C Ltd.	C1415191	-7.79
Indo Count Inds. Ltd.	C1415198	861.42
Kaveri Seed Co. Ltd.	C1415239	54.20
Marksans Pharma Ltd.	C1415264	157.41
Page Industries Ltd.	C1415301	111.83
Strides Shasun Ltd.	C1415360	204.33
Sunteck Realty Ltd.	C1415366	-7.81
Suven Life Sciences Ltd.	C1415368	300.97
Symphony Ltd.	C1415370	252.95
Triveni Turbine Ltd.	C1415403	77.60
<b>Average</b>		<b>139.64</b>

### 7.3 Model Comparison

In this section, we compare the optimally trained ML-I models using hit rates, annual returns, and cumulative returns. For this, we compare the ten pairs of portfolios constructed using the LOG-I and ANN-I models. At portfolio level, LOG-I won the market in all cases and ANN-I won 9 cases out of the 10 test cases (see Tables 3, 4, and 10), achieving portfolio level hit rates of 100% and 90%, respectively. Moreover, at stock level, LOG-I had a higher hit rate than ANN-I (see Tables 5, 6, and 11). Similarly, LOG-I portfolios yielded higher average annual returns than ANN-I portfolios. However, ANN-I models had lower standard deviation in annual returns than LOG-I models (see Table 12). Figure 13 shows the accumulation of wealth for an initial investment of Rs. 100 in model and market portfolios during study period. The LOG-I and ANN-I models consistently outperformed the market with cumulative returns of 9322% and 3768%, respectively, far exceeding the market with a mere 416%, at end of the ten years. It

**Table 3** LOG models compared on portfolio returns using BSE 500

Test Case	Financial Year	Annual Returns of Logit Model Portfolios (%)				Annual Return BSE500 Index (%)
		LOG-I	LOG-II	LOG-III	LOG-IV	
1	2005-2006	125.92	52.05	76.85	96.46	65.17
2	2006-2007	39.60	87.31	4.29	18.42	10.23
3	2007-2008	68.65	70.46	59.21	51.84	24.25
4	2008-2009	-22.61	-42.59	-49.03	-45.86	-42.77
5	2009-2010	239.50	250.95	209.59	250.95	96.38
6	2010-2011	42.28	26.53	26.53	32.41	7.48
7	2011-2012	19.24	.59	17.82	14.97	-9.11
8	2012-2013	20.64	21.23	13.34	89.88	7.07
9	2013-2014	73.80	42.84	34.28	22.46	17.08
10	2014-2015	87.52	136.13	163.55	179.88	33.19
<b>Mean Return</b>		<b>69.45</b>	<b>64.55</b>	<b>55.64</b>	<b>71.14</b>	<b>20.90</b>

**Table 4** ANN models compared on portfolio return using BSE 500

Test Case	Financial Year	Annual Returns of ANN Model Portfolios (%)				Annual Return BSE500 Index (%)
		ANN-I	ANN-II	ANN-III	ANN-IV	
1	2005-2006	96.20	49.06	73.37	63.63	65.17
2	2006-2007	54.88	19.64	9.55	21.73	10.23
3	2007-2008	78.85	62.78	24.68	37.78	24.25
4	2008-2009	-45.15	-41.67	-37.51	-41.77	-42.77
5	2009-2010	187.87	257.85	215.52	257.85	96.38
6	2010-2011	20.58	24.40	32.75	40.51	7.48
7	2011-2012	3.63	2.14	17.83	3.35	-9.11
8	2012-2013	15.53	17.97	15.39	22.46	7.07
9	2013-2014	26.93	42.53	30.71	34.20	17.08
10	2014-2015	139.64	143.24	139.39	160.53	33.19
<b>Mean Return</b>		<b>57.90</b>	<b>57.79</b>	<b>52.17</b>	<b>60.03</b>	<b>20.90</b>

Cell Colour	Result	Description
	Win	Model wins the market
	Loss	Market wins the model

**Table 5** Comparison of LOG models on stock level

Test case	Financial year	Number of true positives in 25 stocks model portfolios			
		LOG-I	LOG-II	LOG-III	LOG-IV
1	2005–2006	12	8	12	13
2	2006–2007	12	10	9	9
3	2007–2008	14	18	12	12
4	2008–2009	14	14	10	12
5	2009–2010	17	18	17	18
6	2010–2011	17	16	15	19
7	2011–2012	18	12	18	15
8	2012–2013	15	16	13	8
9	2013–2014	15	13	10	9
10	2014–2015	18	20	20	22
<b>TP</b>		<b>152</b>	<b>145</b>	<b>136</b>	<b>137</b>

**Table 6** Comparison of ANN models on stock level

Test case	Financial year	Number of true positives in 25 stocks model portfolios			
		ANN-I	ANN-II	ANN-III	ANN-IV
1	2005–2006	13	7	13	11
2	2006–2007	14	7	10	11
3	2007–2008	15	16	8	11
4	2008–2009	13	14	13	13
5	2009–2010	15	19	17	19
6	2010–2011	14	16	17	18
7	2011–2012	14	13	18	12
8	2012–2013	13	14	14	15
9	2013–2014	11	15	10	10
10	2014–2015	21	21	21	20
<b>TP</b>		<b>143</b>	<b>142</b>	<b>141</b>	<b>140</b>

**Table 7** Portfolio level hit rates of LOG and ANN models

Performance measure at portfolio level	LOG models	ANN models
Number of test portfolios (P)	40	40
Number of winning portfolios (TP)	36	36
<b>Hit rate (%)</b>	<b>90</b>	<b>90</b>

**Table 8** Portfolio level comparison of models using four training periods

Performance measure at portfolio level	ML-I	ML-II	ML-III	ML-IV
Number of model portfolios constructed (P)	20	20	20	20
Number of winning portfolios (TP)	19	18	17	18
<b>Hit rate (%)</b>	<b>95</b>	<b>90</b>	<b>85</b>	<b>90</b>

**Table 9** Stock level comparison of models using four training periods

Performance measure at stock level	ML-I	ML-II	ML-III	ML-IV
Number of stocks selected using models (P)	500	500	500	500
Number of winning stocks (TP)	295	287	277	277
<b>Hit rate (%)</b>	<b>59.00</b>	<b>57.40</b>	<b>55.40</b>	<b>55.40</b>

**Table 10** ML-I models compared on portfolio level hit rates

Performance measure at portfolio level	LOG-I model	ANN-I model
Number of test portfolios (P)	10	10
Number of winning portfolios (TP)	10	9
<b>Hit rate (%)</b>	<b>100</b>	<b>90</b>

**Table 11** ML-I models compared on stock level hit rates

Performance measure at stock selection	LOG-I models	ANN-I models
Number of stocks selected using models (P)	250	250
Number of winning stocks (TP)	152	143
<b>Hit rate (%)</b>	<b>60.80</b>	<b>57.20</b>

**Table 12** ML-I models compared on annual returns

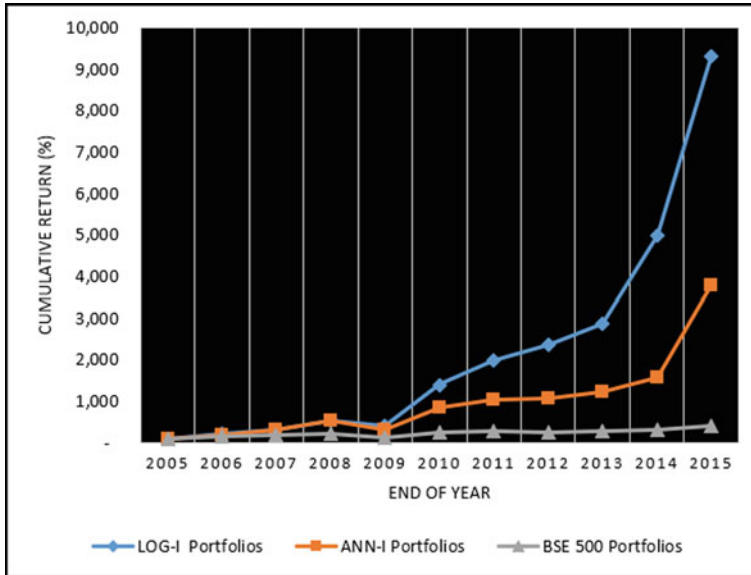
Summary Statistics on Model Portfolio Returns	LOG-I models	ANN-I models	BSE 500
Mean annual return (%)	69.45	57.90	20.90
Maximum annual return (%)	239.50	187.87	96.38
Minimum annual return (%)	-22.61	-45.15	-42.77
Standard deviation of annual return (%)	72.62	69.28	38.40

is further observed that the variances of cumulative returns for ANN-I was less than that of LOG-I models.

## 8 Conclusions

In this study, we developed four pairs of ML models by including one, five, nine, and fourteen years historical records in their training sets, respectively. We validated them based on their model portfolio performance during ten consecutive years. We optimized the models by finding optimal length of training period. Finally, we compared the optimally trained ML-I models.

- The proposed ML models enhanced equities-screening with BSE 500 for improving long-term returns. The models consistently outperformed the market in cumulative returns during the ten years study period.
- The recent past one year records of BSE 500 companies provided the optimal training samples for modeling ML for stock selection.
- The optimally trained ML-I models displayed different strengths:



**Fig. 13** Cumulative returns of ML-I models using buy-and-hold strategy

- LOG-I model had higher portfolio level and stock level hit rates.
- LOG-I model yielded higher average annual returns and accumulated higher cumulative return for the 10 years.
- ANN-I portfolios had a lower standard deviation in annual returns and cumulative returns, implicating some advantageous steadier returns.

### 1. Limitations and Challenges.

The prediction performance of ANN-I was not higher than LOG-I as expected. There are many challenges to this kind of comparative studies. The following reasons are more relevant to this study:

- The changing local and global economy, government policies, and other non-stationary macro indicators may dynamically alter the relationships among fundamental variables in long run.
- Small size training sets constrained through moving window system.
- Possible combination of linear and non-linear patterns in financial data.

### 2. Scope for Further Studies.

This study provides models for the implementation of a full-fledged decision support systems to enhance stock selection. It is intended for long-term investments with Indian stocks, for individual and institutional investors. Furthermore, it demonstrated a methodology for improving stock selection models using comparative studies. We anticipate this methodology unleash a great scope for further studies



in ML for long-term investment decision support systems. We list the following few possibilities in this area of research:

- We included nine financial ratios that appeared frequently in the ratio analysis literature. However, similar experiments may be repeated incorporating many other financial ratios as predictor variables and observe for the model improvements.
- While including many more financial ratios to the models, it may require developing hybrid intelligent models to incorporate: (i) parameter optimization; (ii) feature engineering. Consequently, many comparative studies on hybrid intelligent models will become necessary.
- Apply reinforcement learning to check for the feasibility of this new advanced techniques in stock selection.

## Appendix

**Data Widgets in Orange** We used the following widgets in our workflows (see Figs. 6 and 14) for the data and file handling tasks:

*File* widget to read data from external files into workflows.

*Select Columns* widget to select input attributes, class attribute, and meta attributes from the available columns.

*Concatenate* widget to vertically merge instances from multiple files for ML-III and ML-IV.

*Select Rows* widget to exclude samples belong to the target year after the concatenation of multiple training sets.

*Data Table* widget to view the composed training and test sets in spreadsheet form after selecting columns and rows.

*Save Data* widget to export the end results viewed in Data Table from Orange workflows to external files.

In our case, the external files were in available Excel worksheets or in tab separated text files. Figure 14 shows the workflows for constructing the training sets in Orange. Figure 15, 16, 17, 18 illustrate the widgets reading, composing, and presenting the training, and test sets of ML-I models for the target year 2010–2011.

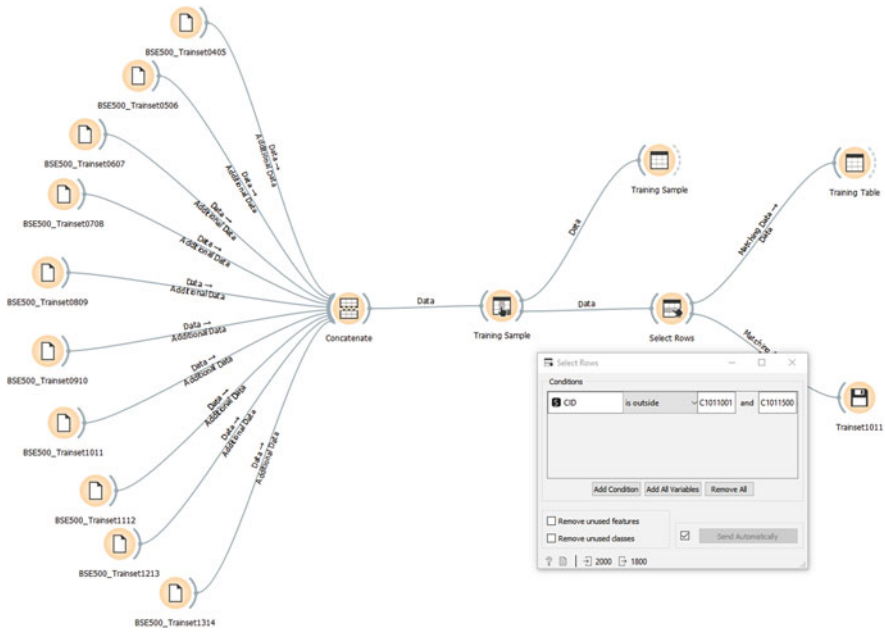


Fig. 14 Workflow for constructing training sets of ML-III

Name	Type	Role	Values
1 PE	numeric	feature	
2 PB	numeric	feature	
3 ROA	numeric	feature	
4 ROE	numeric	feature	
5 DE	numeric	feature	
6 Current	numeric	feature	
7 Quick	numeric	feature	
8 BVS	numeric	feature	
9 EPS	numeric	feature	
10 Returns	numeric	skip	
11 Class	categorical	target	A, B
12 CName	text	meta	
13 CID	text	meta	

Fig. 15 Reading and composing training set

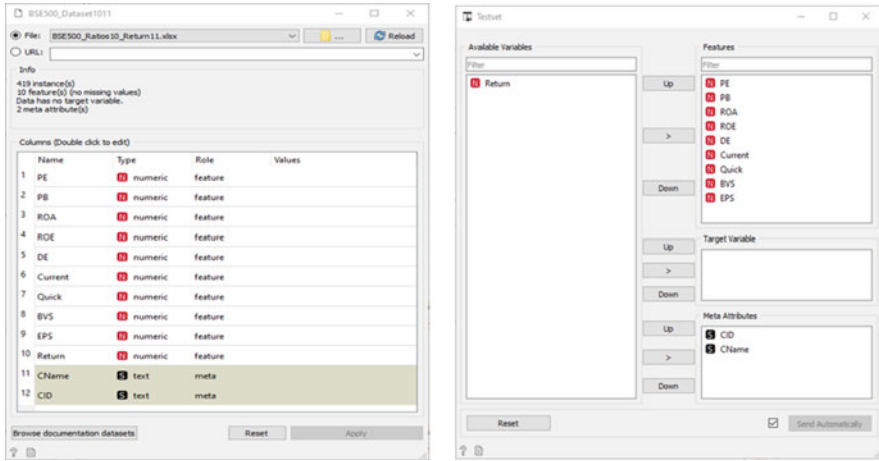


Fig. 16 Reading and composing test set

The image shows a 'Training Table' window in Orange3. It displays a table with 21 rows and 13 columns. The columns are Class, CID, PE, PB, ROA, RDE, DE, Current, Quick, BVS, and EPS. The data represents various financial ratios for different stocks, categorized into Class A and B.

Class	CID	PE	PB	ROA	RDE	DE	Current	Quick	BVS	EPS
B	C0910002	17.99	4.15	10.63	29.46	0.00	1.34	1.12	103.02	23.76
B	C0910003	9.42	2.03	15.39	26.71	0.10	0.96	0.57	284.11	61.24
B	C0910006	2.47	0.61	8.55	26.78	1.42	1.06	0.61	46.68	11.50
B	C0910011	101.71	1.77	1.38	1.96	0.23	0.74	0.30	254.56	4.42
B	C0910014	16.34	1.73	23.29	34.04	0.00	0.56	0.27	254.68	26.95
B	C0910016	18.31	3.62	13.95	21.01	0.42	1.01	0.97	218.72	43.23
B	C0910017	4.06	0.37	2.21	11.82	3.76	1.63	0.99	33.29	3.07
B	C0910019	18.27	6.27	9.75	35.60	0.65	0.92	0.72	32.42	11.12
A	C0910020	4.07	0.78	9.16	21.79	0.70	2.10	1.35	47.49	9.08
B	C0910021	7.90	1.81	19.45	27.17	0.05	1.47	0.82	39.45	9.03
B	C0910024	20.11	1.80	3.74	9.05	0.33	0.95	0.67	226.30	20.50
B	C0910029	17.07	1.97	6.48	12.15	0.31	1.79	1.23	22.84	2.64
A	C0910030	3.56	0.45	8.28	16.68	0.43	1.46	0.68	81.44	10.27
B	C0910031	20.23	11.42	20.89	55.61	0.54	1.40	1.21	49.37	27.86
B	C0910032	3.48	0.37	3.79	11.40	1.06	1.53	0.88	116.66	12.36
A	C0910033	8.22	0.77	3.34	10.12	1.60	1.30	0.82	245.67	23.09
B	C0910035	7.55	1.31	11.39	18.94	0.00	1.47	0.67	136.65	23.69
A	C0910038	2.96	1.08	9.57	44.53	0.92	1.14	0.90	136.18	49.56
A	C0910039	8.88	0.23	0.99	3.15	1.48	0.96	0.04	297.50	7.72
B	C0910040	58.79	2.02	2.60	3.47	0.26	0.64	0.65	83.32	2.86
B	C0910041	16.14	0.64	1.48	4.19	0.00	0.00	0.00	116.96	18.30

Fig. 17 Partial view of a training set

The image shows a 'Test Table' window in Orange3. It displays a table with 21 rows and 13 columns. The columns are CID, CName, PE, PB, ROA, RDE, DE, Current, Quick, BVS, and EPS. The data represents various financial ratios for different stocks, categorized into Class A and B.

CID	CName	PE	PB	ROA	RDE	DE	Current	Quick	BVS	EPS
C1011001	3M India Ltd.	37.55	5.69	16.82	23.55	0.00	2.24	1.52	391.31	
C1011002	A B B India Ltd.	63.15	7.28	6.24	15.71	0.00	1.47	1.19	114.03	
C1011003	A C C Ltd.	11.56	2.78	17.01	29.36	0.09	0.66	0.36	342.01	
C1011004	A I A Engineering Ltd.	30.71	5.06	11.86	17.60	0.00	1.21	0.82	78.86	
C1011005	Aarti Drugs Ltd.	4.68	0.88	8.56	19.95	1.26	1.39	0.94	116.52	
C1011006	Aarti Industries Ltd.	5.21	0.91	6.36	18.94	1.10	0.96	0.49	51.96	
C1011007	Alban Offshore Ltd.	18.07	2.74	5.41	20.29	1.88	0.30	0.39	424.55	
C1011008	Alkerm India Ltd.	16.97	4.27	15.51	31.46	0.30	0.68	0.46	207.20	
C1011009	Adani Enterprises Ltd.	117.72	12.66	3.45	14.09	2.26	0.77	0.71	37.06	
C1011010	Adani Ports & Special Economic Zone Ltd.	46.29	9.08	9.60	21.81	0.82	0.32	0.30	86.99	
C1011011	Adani Power Ltd.	147.99	4.36	1.39	4.23	1.68	0.16	0.16	26.40	
C1011012	Aegis Logistics Ltd.	11.20	2.22	12.74	21.58	0.41	1.17	1.00	99.07	
C1011013	Ajanta Pharma Ltd.	7.48	1.21	6.18	17.40	1.16	2.13	1.00	150.16	
C1011014	Alkzo Nobel India Ltd.	14.17	2.24	11.58	16.25	0.00	0.54	0.27	268.94	
C1011015	Alkhabad Bank	5.40	1.08	1.11	22.21	0.82	3.18	3.17	131.73	
C1011016	Allcargo Logistics Ltd.	21.39	2.83	10.48	19.21	0.14	0.87	0.82	65.79	
C1011017	Alksh Industries Ltd.	5.37	0.64	2.32	11.06	3.13	1.90	1.14	34.48	
C1011018	Alksum India Ltd.	26.14	0.41	5.00	37.29	0.03	0.83	0.75	73.79	
C1011019	Alksum T & D India Ltd.	51.81	0.39	5.43	24.17	0.89	0.82	0.89	36.34	
C1011020	Amara Raja Batteries Ltd.	8.42	2.57	17.68	35.19	0.17	1.65	0.97	63.65	
C1011021	Ambuja Cements Ltd.	14.29	2.63	14.55	20.08	0.03	1.02	0.64	45.50	

Fig. 18 Partial view of a test set

## References

1. Ahn, B. S., Cho, S. S., & Kim, C. Y. (2000). The integrated methodology of rough set theory and artificial neural network for business failure prediction. *Expert Systems with Applications*, 18(2), 65–74.
2. Becker, Y., Fei, P., & Lester, A. (2007). Stock selection: An innovative application of genetic programming methodology. *Genetic Programming Theory and Practice IV*, (617), 315–334.
3. Beynon, M. J., Clatworthy, M. a., & Jones, M. J. (2004). The prediction of profitability using accounting narratives: a variable-precision rough set approach. *Intelligent Systems in Accounting, Finance & Management*, 12(4), 227–242.
4. Cao, Y., Chen, X., Wu, D. D., & Mo, M. (2011). Early warning of enterprise decline in a life cycle using neural networks and rough set theory. *Expert Systems with Applications*, 38(6), 6424–6429.
5. Cao, Y., Wan, G., & Wang, F. (2011). Predicting Financial Distress of Chinese Listed Companies Using Rough Set Theory and Support Vector Machine. *Asia-Pacific Journal of Operational Research*, 28(01), 95.
6. Cawley, G. C., & Talbot, N. L. C. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11, 2079–2107.
7. Chen, S.-S., Huang, C.-F., & Hong, T.-P. (2013). A multi-objective genetic model for stock selection. *Kaigi.Org*, 2–6.
8. Demsar, J., Curk, T., Erjavec, A., Gorup, C., Hocevar, T., Milutinovic, M., . . . Zupan, B. (2013). Orange: Data Mining Toolbox in Python. *Journal of Machine Learning Research*, 14, 2349–2353.
9. Ding, Y., Song, X., & Zen, Y. (2008). Forecasting financial condition of Chinese listed companies based on support vector machine. *Expert Systems with Applications*, 34(4), 3081–3089.
10. Fan, A., & Palaniswami, M. (2001). Stock selection using support vector machines. *IJCNN'01. International Joint Conference on Neural Networks. Proceedings*, 3, 1793–1798.
11. Feng, X., & Kong-lin, K. (2008). Five-Category Evaluation of Commercial Bank's Loan by the Integration of Rough Sets and Neural Network. *Systems Engineering - Theory & Practice*, 28(1), 40–45.
12. Geng, R., Bose, I., & Chen, X. (2015). Prediction of financial distress: An empirical study of listed Chinese companies using data mining. *European Journal of Operational Research*, 241(1), 236–247.
13. Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann (3rd Editio).
14. Härdle, W., Lee, Y., Schäfer, D., & Yeh, Y. (2009). Variable selection and oversampling in the use of smooth support vector machines for predicting the default risk of companies. *Journal of Forecasting*, 28, 512–534.
15. Hargreaves, C., & Hao, Y. (2013). Prediction of stock performance using analytical techniques. *Journal of Emerging Technologies in Web Intelligence*, 5(2), 136–142.
16. Hassan, G., & Clack, C. (2009). Robustness of multiple objective GP stock-picking in unstable financial markets. In *GECCO'09 Proceedings of the 11th Annual conference on Genetic and evolutionary computation* (pp. 1513–1520).
17. Huang, C.-F., Chang, B. R., Cheng, D.-W., & Chang, C.-H. (2012). Feature selection and parameter optimization of a fuzzy-based stock selection model using genetic algorithms. *International Journal of Fuzzy Systems*, 14(1), 65–75.
18. Huang, C.-F., Hsieh, T., Chang, B. R., & Chang, C. (2011). A comparative study of stock scoring using regression and genetic-based linear models. *2011 IEEE International Conference on Granular Computing*, 268–273.

19. Huang, S.-C., Tang, Y.-C., Lee, C.-W., & Chang, M.-J. (2012). Kernel local Fisher discriminant analysis based manifold-regularized SVM model for financial distress predictions. *Expert Systems with Applications*, 39(3), 3855–3861.
20. Krishna Kumar, M. S., Subramanian, S., & Rao, U. S. (2010). Enhancing stock selection in Indian stock market using value investment criteria: An application of artificial neural networks. *The IUP Journal of Accounting Research and Audit Practices*, 9(4), 54–67.
21. Lai, K., Yu, L., Wang, S., & Zhou, C. (2006). A double-stage genetic optimization algorithm for portfolio selection. *13th International Conference on Neural Information Processing*, 928–937.
22. Min, J. H., & Lee, Y. C. (2005). Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. *Expert Systems with Applications*, 28(4), 603–614.
23. Min, S. H., Lee, J., & Han, I. (2006). Hybrid genetic algorithms and support vector machines for bankruptcy prediction. *Expert Systems with Applications*, 31(3), 652–660.
24. Mironiuc, M., & Robu, M.-A. (2013). Obtaining a Practical Model for Estimating Stock Performance on an Emerging Market Using Logistic Regression Analysis. *Procedia – Social and Behavioral Sciences*, 81, 422–427.
25. Olson, D., & Mossman, C. (2003). Neural network forecasts of Canadian stock returns using accounting ratios. *International Journal of Forecasting*, 19(3), 453–465.
26. Pao, H. (2008). A comparison of neural network and multiple regression analysis in modeling capital structure. *Expert Systems with Applications*, 35(3), 720–727.
27. Powers, D. M. W. (2007). Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies*, 2, 37–63.
28. Quah, T.-S. (2008). DJIA stock selection assisted by neural network. *Expert Systems with Applications*, 35(1–2), 50–58.
29. Quah, T.-S., & Srinivasan, B. (1999). Improving returns on stock investment through neural network selection. *Expert Systems with Applications*, 17(4), 295–301.
30. Šarlija, N., Bilandžić, A., & Stanić, M. (2017). Logistic regression modelling: procedures and pitfalls in developing and interpreting prediction models. *Croatian Operational Research Review*, 8(2), 631–652.
31. Shmueli, G., Patel, N., & Bruce, P. (2010). *Data Mining for Business Intelligence in XLMiner*. Wiley.
32. Telmoudi, F., Ghourabi, M., & Limam, M. (2011). RST-GCBER-CLUSTERING-BASED RGA-SVM model for corporate failure prediction. *Intelligent Systems in Accounting, Finance & Management*, 18(June 2011), 105–120.
33. Thenmozhi, M. (2006). Forecasting Stock Index Returns Using Neural Networks. *Delhi Business Review*, 7(2), 59–69.
34. Turban, E., Sharda, R., & Delen, D. (2011). *Decision Support and Business Intelligence Systems* (9th Editio). Prentice Hall.
35. Vanstone, B., Finnie, G., & Tan, C. (2004). Enhancing security selection in the Australian stockmarket using fundamental analysis and neural networks. *Bond University EPublications@bond*.
36. Witten, I., Frank, E., & Hall, M. (2011). *Data Mining:: Practical Machine Learning Tools and Techniques* (Third). The Morgan Kaufmann Series in Data Management Systems.
37. Yeh, C.-C., Chi, D.-J., & Hsu, M.-F. (2010). A hybrid approach of DEA, rough set and support vector machines for business failure prediction. *Expert Systems with Applications*, 37(2), 1535–1541.
38. Yildiz, B. (1999). Fundamental analysis with neuro-fuzzy technology: An experiment in Istanbul stock exchange. *Sbd.Ogu.Edu.Tr*, 8(2), 25–42.
39. Yildiz, B., & Yezegel, A. (2010). Fundamental analysis with artificial neural network. *The International Journal of Business and Finance Research*, 4(1), 149–159.
40. Zekić-Sušac, M., Šarlija, N., Has, A., & Bilandžić, A. (2016). Predicting company growth using logistic regression and neural networks. *Croatian Operational Research Review*, 7(2), 229–248.

# Landslide Detection with Ensemble-of-Deep Learning Classifiers Trained with Optimal Features



Abhijit Kumar , Rajiv Misra, T. N. Singh, and Vinay Singh

## 1 Introduction

Due to the force of gravity [1–4], soil, debris, and rocks slide down a slope, causing landslides. Multiple fatalities and large monetary losses are very common in the highlands and hills. There are also long-term consequences to its secondary risks. Landslide formation is influenced by a variety of environmental factors, including topography, lithology, land cover, and hydrology, according to some studies [5–8]. In addition to gravity, landslides may be triggered by rain, earthquakes, and human activity. With that being the case, it's vital to detect and anticipate landfalls' locations in order to avoid or minimize potential losses [9–12].

Quantitative predictive techniques for regional landslide spatial prediction have recently been published. Detailed geological field surveys may offer very accurate results when employing slope stability and landslide models. The issue is that effective physical modelling requires a wide variety of components, which is

---

A. Kumar (✉)

Department of Computer Science and Engineering, Indian Institute of Technology Patna,  
School of Computer Science, University of Petroleum and Energy Studies Dehradun, Dehradun,  
IndiaPatna, India

R. Misra

Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna,  
India

T. N. Singh

Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna,  
India

Department of Earth Science, Indian Institute of Technology Bombay, Bombay, India

V. Singh

Department of Earth Science, Indian Institute of Technology Bombay, Bombay, India

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023

R. Misra et al. (eds.), *Advances in Data Science and Artificial Intelligence*,

Springer Proceedings in Mathematics & Statistics 403,

[https://doi.org/10.1007/978-3-031-16178-0\\_21](https://doi.org/10.1007/978-3-031-16178-0_21)

difficult to accomplish. This series has [13–16]. A new study has employed landslide susceptibility mapping (LSM), GIS, and remote sensing (RS) to evaluate the possible links between landslide conditions and the likelihood of occurrence. LSM can identify landslide-prone areas and build disaster mitigation and prevention strategies [17–19].

To assist LSM, many quantitative methods have been developed. Machine learning (ML) is increasingly being used in LSM because of its capacity to efficiently capture the correlations between landslides and environmental factors, such as logistic regression (LGR), support vector machines (SVMs), and decision trees (DTs), and random forests (RFs). However, classification by most methodologies does not reveal many characteristics of the causes of landslides. Deep learning methods such as DBN, RNN, and CNN, as well as convolutional neural networks, have become increasingly popular because of their ability to extract features (CNNs). In LSM's research, CNNs performed better in extracting spatial features than the others [24]. Twenty-five and a half. Landslide sampling, on the other hand, restricts the use of certain designs for feature extraction because of its data expression. There has been a lot of interest in deep learning approaches, which are capable of extracting features and capturing deep representations from large datasets. But, only a few hybrid models were used for the thorough usage of characteristics [26–29]. However, since most techniques only consider one feature dimension, they have poor generalizability in more complex situations. Hybrid methods are now in use that makes use of each approach's advantages for optimal feature utilization because of the complex nonlinear relationship of components and over-fitting.

## 2 Literature Review

To quantify landslide risk in Zichang City, China, Chen et al. [1] developed the bivariate statistical kernel logistic regression models PLKLR, PUKLR, and RBFKLR in 2021. It is now possible to create landslide susceptibility maps by comparing three landslide susceptibility maps and examining geographical trends. To begin, a 263 site historical landslide inventory was established. 263 landslide sites were used to train and test model assumptions and hypotheses. Second, 14 landslide conditioning variables were derived from the geographic data. Then, using frequency ratios, we investigated the relationship between the conditioning elements and landslide incidence. Then maps of landslide susceptibility were created using the normalized frequency ratios of the three models. Using correlation statistics, we looked at multi-collinearity. Researchers employed AUC comparison and validation to assess a model's predictive ability. As a result, quantitative comparisons of susceptibility maps are required to prevent over or underestimating factors (distance to the river and slope). The PUKLR model has AUC values of 0.884 and 0.766 for the training and validation datasets, respectively. The datasets were trained using RBFKLR and PLKLR models with AUCs of 0.879 and 0.797. These models were

used to verify and train datasets (AUC values of 0.758 and 0.752, respectively). The landslide susceptibility map may assist Zichang's decision-makers avoid future natural disasters.

In 2021, Zhang et al. [2] developed a deep learning system using spatial response characteristics and machine learning classifiers (SR-ML). There are three stages to the process. DSC collects geographical features to avoid confusing multi-factor data. Second, spatial pyramid pooling is used to obtain response characteristics of varied sizes (SPP). 3. High-level features are integrated with ML classifiers to enhance feature categorization. Examples of meaningful feature categorization using machine learning classifiers are provided in this framework. The Yarlung Zangbo Grand Canyon area gathered data on 203 landslides and 11 conditioning variables. The AUC for the suggested SR and SR-ML was 0.920 and 0.910, greater than the random forest (RF, with the largest AUC in ML group). Bigger landslide samples exhibited the lowest mean error (0.01), suggesting that LSM might benefit from utilizing larger landslide samples.

There will be an increased danger of earthquakes in the Zagros Mountains in Iran by 2021, according to Paryani et al. [3]. They used a combination of machine learning and metaheuristic algorithms, including the adaptive neuro-fuzzy inference system and the Harris hawks optimization (BA). Landslide data was divided in half using a 70/30 ratio for training and testing purposes. There were 14 landslide-related variables examined, and the stepwise weight assessment ratio (SWARA) was used to find the relationship between landslides and components. It was then used to create landslide susceptibility maps (LSMs) based on the hybrid models of ANFIS, HHO, SVR, SVR-HHO, and SVR-BA (LSMs). Lastly, two indices, namely MSE and AUROC, were utilized to compare and validate the models employed in the study. The AUROC values for the ANFIS-HHO, ANFIS-BA, SVR-HHO, and SVR-BA were 0.849, 0.82, 0.895, and 0.865, respectively, according to the validation results. With an AUROC of 0.895 and an MSE of 0.147, SVR-HHO was the most accurate while ANFIS-BA was the least accurate, both based on an AUROC value of 0.82. Based on the data, the SVR model is superior than the ANFIS model, and the HHO algorithm has beaten the bat approach in terms of performance. Property use planners may utilize the map created in this study to better manage their land.

Using computer-based sophisticated machine learning methodologies in the year 2021, Mandal et al. [4] built LSMs and compared the models' performance. A total of twenty factors, including both starting and contributing components, were examined in order to properly appreciate the landslide's spatial connection. One of the most popular machine learning techniques, convolutional neural network (CNN), was utilized to develop LSMs. Random forest (RF), artificial neural network model (ANN), and the bagging model were all used in conjunction with it. Landslide and non-landslide locations were randomly selected for training and validation datasets. The training and validation locations were selected in a ratio of 70:30. Multi-collinearity was assessed using the tolerance and variance inflation factor, while the information gain ratio was utilized to assess the significance of certain conditioning factors. Studies have shown a low degree of multi-collinearity when it comes to landslide conditioning factors, with rainfall being the most significant



contributor. Using each model's final prediction results, LSM was then split into five distinct categories, such as "very low," low, medium, high, and very high susceptibility. Based on the landslide susceptibility class distribution, more than 90% of the landslide area is vulnerable to landslides. According to the area under the receiver operating characteristics curve (ROC) curve and statistical methodologies such as RMSE and MAE were used to evaluate the models' accuracy (MAE). The CNN model achieved the highest AUC values in both datasets (training and validation): 0.903 and 0.939, respectively. As can be seen by the lower RMSE and MAE values, the CNN model performs better than other models. As a result, all models have performed well, but the CNN model has outperformed the other models in terms of accuracy.

With the use of sentinel-1D InSAR and MODIS data, Al-Najjar [5] hopes to demonstrate a method for predicting and mapping long-term and seasonal land surface deformation (subsidence/uplifting) and permafrost active layer thickness in Alaska's Donnelly Training Area (DTA) by the year 2020 (ALT). SAR images were compared for coherence (or resemblance) to see whether they could be used together. They were tested for their overall quality by conducting sensitivity analyses and an accuracy review. Seasonal subsidence in June and July was forecast to be in the range of 0–0.43%, whereas the predicted uplifting from September to May of the next year was predicted to be in the range of 0.34 m. DTA's southern and northern areas were expected to experience the majority of the long-term subsidence and rising (from 2015 to 2018). In the east river, west, and south, ALT estimates were greater, while those in the north were lower as a result of spatially variable time delays. For example, coherence estimates were considerably different from zero, and the average residual from ALT estimations was statistically indistinguishable from zero when compared to the referenced forecasts from a commonly used yearly prediction model. For seasonal surface deformation estimates, regional distributions of the uncertainties in model coefficient estimates, phase change estimates, modeling error estimates, and image pairs were similar. With the use of the InSAR pictures and MODIS data, this approach has been able to map and monitor permafrost changes in regions like DTA, where collecting field observations is difficult and costly. We also discovered that the DTA's permafrost deformation is very variable in terms of location and time, which enabled us to develop a near-real-time monitoring system for the permafrost environment.

### 3 Problem Statement

LSM depends on aerial picture interpretation and field verification, but gathering aerial photos is difficult [7, 8]. Researchers have gradually applied LDM to environmental monitoring using remote sensing (RS) [9–13]. Initially, high-resolution laser images were used to locate large-scale landslides [14, 15]. In modest shallow landslides, laser scans cannot detect A landslide's texture, color, and other features were compared to the surrounding ground objects in optical photographs.

WorldView [17], QuickBird [18], GaoFen-2 [19], IKONOS [20], and InSAR technologies are increasingly employed to LDM [21]. Both pixel-based and object-oriented techniques are used [22]. In this method, high-resolution photographs are used to identify landslides. These and other stages of landslide detection generate error accumulation and fluctuation in accuracy (ACC) [5, 23]. However, pixel-based LDM may solve these difficulties by simply identifying single-pixel [24, 25]. In addition, support vector machines (SVM), random forests (RF), decision trees (DT), and artificial neural networks (ANN) are used (ANN). Landslides may also be retrieved from optical images [30]. Others exploited the study area's attributes for LDM such as hydrology or evening light. However, most of the literature using RS images for LDM disregards landslide debris' morphological, geological, and other features. So we are worried about landslides. Convolutional neural networks (CNNs) can train a large number of parameters efficiently by utilizing weight sharing and other qualities. LeNet, AlexNet, GoogleNet, VGGNet, ResNet, and DenseNet are all deep neural network models based on the basic CNN model. These models excel in image classification. The CNN model is often utilized in landslide research. In landslide detection, its upgraded approach residual neural networks (ResNets) has demonstrated promising results. Due to the large amount of RS data employed, the preceding strategies need long model training durations and a large number of training parameters. DenseNets are narrow networks with few parameters, reducing model training time. DenseNet excels in medical image segmentation [16]. In landslide detection, DenseNet has limited applicability. An ensemble of deep learning classifiers is proposed.

## 4 Methodology

In this research work, a novel landslide detection model for GIS images will be introduced by following their major phases: (i) pre-processing, (ii) feature extraction (iii) feature selection, and (iv) classification. The captured images will be pre-processed via Gabor filtering. Subsequently, the features like GLCM based texture features, temperature-vegetative index-based characteristics, Brightness Index (BR), Normalized Difference Vegetation Index (NDVI) and Green Normalized Difference Vegetation Index (GNDVI), Red-over-Green difference (RGD), Vegetation index difference (VID), Brightness difference (BRD), NDI based features and coloration index features are extracted from the pre-processed data. The extracted features will be fused together, and among those features, the optimal ones will be selected via a new hybrid optimization model. The new hybrid optimization model will be the conceptual blend of the standard Teamwork Optimization Algorithm (TOA) [31] and Poor and rich optimization algorithm (PRO) [32]. Finally, the selected optimal features are subjected to the newly constructed ensemble-of-classifiers model. The ensemble-of-classifiers model is constructed with Recurrent Neural Network (RNN), Bi-LSTM, and Bi-GRU. All these classifiers are trained with the selected optimal features acquired with the new hybrid optimization model. The

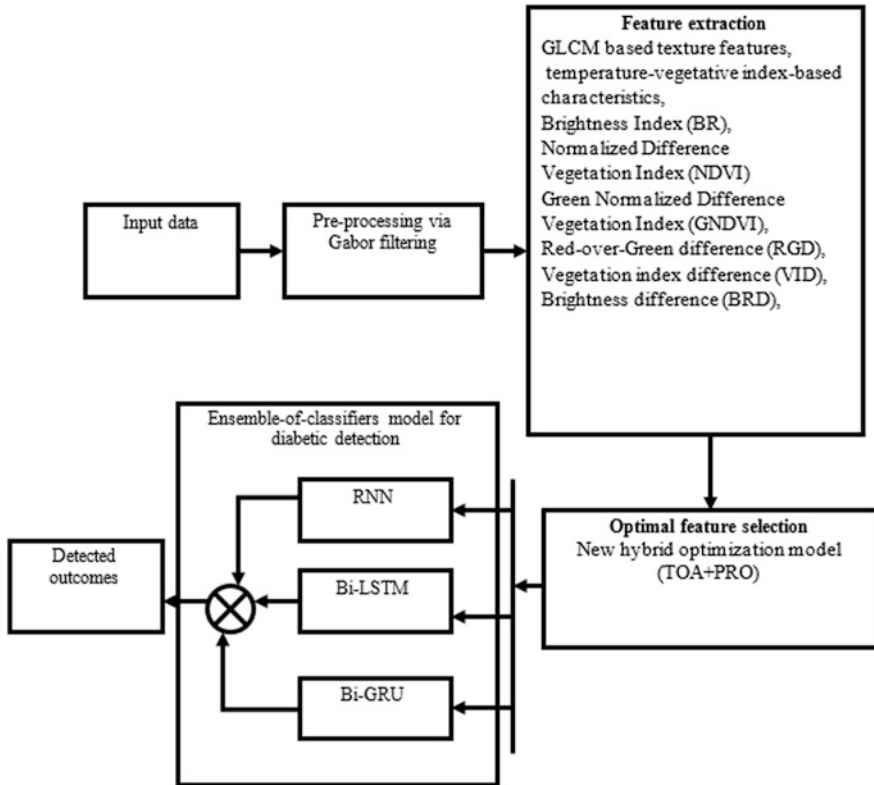


Fig. 1 The architecture of the projected model

ultimate outcome regarding the landslide forecasting will be acquired by fusing the outcomes acquired from RNN, Bi-LSTM and Bi-GRU as depicted in Fig. 1.

### 4.1 RNN

RNN is a type of neural network that has cyclic connections between its own nodes and is designed to mimic problems that a fully connected network cannot. Most neural networks, such as ANN and CNN, only build weight connections between layers; nodes between layers are disconnected, and nodes must be self-contained. In real life, however, many data are related to one another, necessitating a network model that can incorporate both prior and subsequent information, as well as handle data of any length at the same time. RNN was created to solve difficulties like these.

## 4.2 Bi-LSTM

The Bi-LSTM architecture consists of a forward LSTM and a reverse LSTM. The forward and backward layers both perform the usual LSTM function. The forward layer will compute a positive input sequence, whereas the backward layer will compute a reverse time sequence. The output of Bi-LSTMs can be described as a summation function of two hidden layer function outputs.

## 4.3 Bi-GRU

LSTM is proposed to tackle the “long dependencies” difficulties of regular RNNs, but it also fails to deal with very long-term and many dependencies. As a result, Bi-GRU is better suited to processing very lengthy dependencies since it may use both prior and subsequent data.

## 5 Results & Discussion

The proposed model has been tested in Python. The proposed model’s performance has been compared to other existing models using Type I and Type II metrics. Negative predictive value (NPV), F1-Score, and Mathews correlation coefficient (MCC) are positive measurements whereas false-positive rate (FPR), false-negative rate (FNR), and false discovery rate (FDR) are negative measures.

The ultimate outcome regarding the landslide forecasting will be acquired by fusing the outcomes acquired from RNN, Bi-LSTM and Bi-GRU. The outcome acquired from RNN is  $out^{RNN}$ . The outcome acquired from Bi-LSTM is  $out^{Bi-LSTM}$ . The outcome acquired from Bi-GRU is  $out^{Bi-GRU}$  the final outcome is

$$out = \frac{out^{RNN} + out^{Bi-LSTM} + out^{Bi-GRU}}{3}.$$

This outcome tells about the presence/absence of landslide in input GIS images.

An indication for evaluating the proposed prediction model’s performance is introduced to compare the the mean square error (MSE), root mean square error (RMSE), standard deviation (SD) and mean error (ME) of the simulation results. Equations (1)–(4) illustrate RMSE, MSE, SD, and ME:

$$RMSE = \sqrt{\frac{\sum_{k=1}^n (y_k - \hat{y}_k)^2}{N}} \quad (1)$$

$$MSE = \frac{\sum_{k=1}^n (y_k - \hat{y}_k)^2}{N} \quad (2)$$

$$SD = \sqrt{\frac{1}{N} \sum_{k=1}^n (y_k - \mu)^2} \quad (3)$$

$$ME = \frac{1}{N} \sum_{k=1}^n (y_k - \hat{y}_k)^2 \quad (4)$$

Where  $n$  implies a number of sample data,  $\mu$  implies the observed data's arithmetic mean,  $y_k$  represents  $\hat{y}_k$  denotes the observation and prediction values.

## 6 Conclusion

We compared the ensemble-of-classifiers model to existing landslide detection methods using the same training and simulation parameters. The effectiveness of the ensemble of classifiers (RNN + Bi-LSTM + Bi-GRU) landslide detection system is compared with vanilla RNN + Bi-LSTM and RNN + Bi-GRU. The experiments proved that the proposed method has outperformed the RNN + Bi-LSTM and RNN + Bi-GRU both in terms of performance measures and robustness. With the hybrid optimization model, our network performs better, which achieves 87% Training Accuracy. The main problem that we faced is training images, which is a very time-consuming task. Our future work will focus on reducing the computational complexities of this landslide detection system.

## References

1. Xi Chen, Wei Chena, "GIS-based landslide susceptibility assessment using optimized hybrid machine learning methods", CATENA, 2021
2. HuijuanZhang, YingxuSong, YueWang, "Combining a class-weighted algorithm and machine learning models in landslide susceptibility mapping: A case study of Wanzhou section of the Three Gorges Reservoir, China", Computers & Geosciences, 2021

3. Sina Paryani, Aminreza Neshat, Biswajeet Pradhan, “Improvement of landslide spatial modeling using machine learning methods and two Harris hawks and bat algorithms”, *The Egyptian Journal of Remote Sensing and Space Science*, 2021
4. Kanu Mandal, Sunil Saha, Sujit Mandal, “Applying deep learning and benchmark machine learning algorithms for landslide susceptibility modelling in Rorachu river basin of Sikkim Himalaya, India” *Geoscience Frontiers*, 2021
5. Husam A. H, Al-Najjar, Biswajeet Pradhan, “Spatial landslide susceptibility assessment using machine learning techniques assisted by additional data created with generative adversarial networks”, *Geoscience Frontiers*, 2020.
6. S. Chen, Z. Miao, L. Wu and Y. He, “Application of an Incomplete Landslide Inventory and One Class Classifier to Earthquake-Induced Landslide Susceptibility Mapping,” in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 1649–1660, 2020. <https://doi.org/10.1109/JSTARS.2020.2985088>.
7. H. Cai, T. Chen, R. Niu and A. Plaza, “Landslide Detection Using Densely Connected Convolutional Networks and Environmental Conditions,” in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol.14, pp. 5235–5247, 2021. <https://doi.org/10.1109/JSTARS.2021.3079196>.
8. N. Shen *et al.*, “Short-Term Landslide Displacement Detection Based on GNSS Real-Time Kinematic Positioning,” in *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–14, 2021, Art no. 1004714. <https://doi.org/10.1109/TIM.2021.3055278>
9. B. Pradhan, H. A. H. Al-Najjar, M. I. Sameen, M. R. Mezaal and A. M. Alamri, “Landslide Detection Using a Saliency Feature Enhancement Technique From LiDAR-Derived DEM and Orthophotos,” in *IEEE Access*, vol. 8, pp. 121942–121954, 2020. <https://doi.org/10.1109/ACCESS.2020.3006914>
10. T. Liu, T. Chen, R. Niu and A. Plaza, “Landslide Detection Mapping Employing CNN, ResNet, and DenseNet in the Three Gorges Reservoir, China,” in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol.14, pp.11417–11428, 2021. <https://doi.org/10.1109/JSTARS.2021.3117975>
11. Z. Y. Lv, W. Shi, X. Zhang and J. A. Benediktsson, “Landslide Inventory Mapping From Bitemporal High-Resolution Remote Sensing Images Using Change Detection and Multi-scale Segmentation,” in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 5, pp. 1520–1532, May 2018. <https://doi.org/10.1109/JSTARS.2018.2803784>
12. Y. Yi and W. Zhang, “A New Deep-Learning-Based Approach for Earthquake-Triggered Landslide Detection From Single-Temporal RapidEye Satellite Imagery,” in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 6166–6176, 2020. <https://doi.org/10.1109/JSTARS.2020.3028855>
13. W. Shi and P. Lu, “Intelligent Perception of Coseismic Landslide Migration Areas Along Sichuan–Tibet Railway,” in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 8876–8883, 2021. <https://doi.org/10.1109/JSTARS.2021.3105671>
14. W. Shi, M. Zhang, H. Ke, X. Fang, Z. Zhan and S. Chen, “Landslide Recognition by Deep Convolutional Neural Network and Change Detection,” in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 6, pp. 4654–4672, June 2021. <https://doi.org/10.1109/TGRS.2020.3015826>
15. B. Fang, G. Chen, L. Pan, R. Kou and L. Wang, “GAN-Based Siamese Framework for Landslide Inventory Mapping Using Bi-Temporal Optical Remote Sensing Images,” in *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 3, pp. 391–395, March 2021. <https://doi.org/10.1109/LGRS.2020.2979693>
16. M. I. Sameen and B. Pradhan, “Landslide Detection Using Residual Networks and the Fusion of Spectral and Topographic Information,” in *IEEE Access*, vol. 7, pp. 114363–114373, 2019. <https://doi.org/10.1109/ACCESS.2019.2935761>

17. S. L. Ullo *et al.*, “A New Mask R-CNN-Based Method for Improved Landslide Detection,” in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 3799–3810, 2021. <https://doi.org/10.1109/JSTARS.2021.3064981>
18. M. Zhang, W. Shi, S. Chen, Z. Zhan and Z. Shi, “Deep Multiple Instance Learning for Landslide Mapping,” in *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 10, pp. 1711–1715, Oct. 2021. <https://doi.org/10.1109/LGRS.2020.3007183>
19. M. Q. Pham, P. Lacroix and M. P. Doin, “Sparsity Optimization Method for Slow-Moving Landslides Detection in Satellite Image Time-Series,” in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 4, pp. 2133–2144, April 2019. <https://doi.org/10.1109/TGRS.2018.2871550>
20. Q. Huang, C. Wang, Y. Meng, J. Chen and A. Yue, “Landslide Monitoring Using Change Detection in Multitemporal Optical Imagery,” in *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 2, pp. 312–316, Feb. 2020. <https://doi.org/10.1109/LGRS.2019.2918254>
21. B. Zhang and Y. Wang, “An Improved Two-Step Multitemporal SAR Interferometry Method for Precursory Slope Deformation Detection Over Nanyu Landslide,” in *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 4, pp. 592–596, April 2021. <https://doi.org/10.1109/LGRS.2020.2981146>
22. G. Yao *et al.*, “An Empirical Study of the Convolution Neural Networks Based Detection on Object With Ambiguous Boundary in Remote Sensing Imagery—A Case of Potential Loess Landslide,” in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 323–338, 2022 <https://doi.org/10.1109/JSTARS.2021.3132416>
23. L. Nava, O. Monserrat and F. Catani, “Improving Landslide Detection on SAR Data Through Deep Learning,” in *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022, Art no. 4020405. <https://doi.org/10.1109/LGRS.2021.3127073>
24. F. K. Sufi and M. Alsulami, “Knowledge Discovery of Global Landslides Using Automated Machine Learning Algorithms,” in *IEEE Access*, vol. 9, pp. 131400–131419, 2021, <https://doi.org/10.1109/ACCESS.2021.3115043>
25. C. Ye *et al.*, “Landslide Detection of Hyperspectral Remote Sensing Data Based on Deep Learning With Constrains,” in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 12, pp. 5047–5060, Dec.2019. <https://doi.org/10.1109/JSTARS.2019.2951725>
26. Z. Lv, T. Liu, X. Kong, C. Shi and J. A. Benediktsson, “Landslide Inventory Mapping With Bitemporal Aerial Remote Sensing Images Based on the Dual-Path Fully Convolutional Network,” in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 4575–4584, 2020. <https://doi.org/10.1109/JSTARS.2020.2980895>
27. J. Liu, D. Chen, Y. Wu, R. Chen, P. Yang and H. Zhang, “Image Edge Recognition of Virtual Reality Scene Based on Multi-Operator Dynamic Weight Detection,” in *IEEE Access*, vol. 8, pp. 111289–111302, 2020. <https://doi.org/10.1109/ACCESS.2020.3001386>
28. T. Lei, Y. Zhang, Z. Lv, S. Li, S. Liu and A. K. Nandi, “Landslide Inventory Mapping From Bitemporal Images Using Deep Convolutional Neural Networks,” in *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 6, pp. 982–986, June2019. <https://doi.org/10.1109/LGRS.2018.2889307>
29. L. Zhiyong, T. Liu, R. Y. Wang, J. A. Benediktsson and S. Saha, “Automatic Landslide Inventory Mapping Approach Based on Change Detection Technique With Very-High-Resolution Images,” in *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022, Art no. 6000805. <https://doi.org/10.1109/LGRS.2020.3041409>
30. C. Ren, H. Shang, Z. Zha, F. Zhang and Y. Pu, “Color Balance Method of Dense Point Cloud in Landslides Area Based on UAV Images,” in *IEEE Sensors Journal*, vol. 22, no. 4, pp. 3516–3528, 15 Feb.15, 2022. <https://doi.org/10.1109/JSEN.2022.3141936>
31. Mohammad Dehghani and Pavel Trojovský, “Teamwork Optimization Algorithm: A New Optimization Approach for Function Minimization/Maximization”, *Sensors*, 2021
32. Seyyed Hamid, SamarehMoosavi, VahidKhatibi, Bardsiri, “Poor and rich optimization algorithm: A new human-based and multi populations algorithm”, *Engineering Applications of Artificial Intelligence*, Vol.86, PP.165–181, 2019.

# A Survey Paper on Text Analytics Methods for Classifying Tweets



Utkarsh Bansod, Dheetilekha Nath, Chanchal Agrawal, Srishti Yadav, Ashwini Dalvi, and Faruk Kazi

## 1 Introduction

Social media is a platform where people share their views, information, facts, and news about various matters or events on diverse domains. Over the past decade, social media has facilitated a platform for analyzing popular topics, sentiment/mood, defining users within a group, characterizing how group members interact, identifying influential users, and prediction of real-world events or characteristics. Out of all social media platforms, Twitter has come up to be one of the most widely used platforms for the majority of people worldwide.

Tweets serve as the source of a huge amount of unstructured textual data, which can be analyzed by application of various techniques such as sentiment analysis, opinion mining, argument mining, stance detection, topic modeling, and more, each serving a different purpose. This survey paper is focused on the various text analytics methods on tweets in order to gather some valuable insights from the textual data that represents feelings/opinions and mood of the public. Gathering the information regarding the opinion or general feeling/mood of the masses attracts organizations and researchers from many diverse fields and areas, including sociology, marketing, finance, and computer.

---

U. Bansod · D. Nath · C. Agrawal (✉) · S. Yadav  
Department of Electronics Engineering, Veermata Jijabai Technological Institute, Mumbai, India  
e-mail: [caagrawal\\_b18@el.vjti.ac.in](mailto:caagrawal_b18@el.vjti.ac.in)

A. Dalvi  
Department of Computer Engineering, Veermata Jijabai Technological Institute, Mumbai, India

F. Kazi  
Department of Electrical Engineering, Veermata Jijabai Technological Institute, Mumbai, India



To the best of our knowledge, this is the first survey paper to discuss and differentiate among the various text classification methods, specifically sentiment analysis, opinion mining, argument mining, and stance classification. Given the task of classifying a user's bias or general feeling toward a campaign, law or product leads one to explore these various text analysis techniques, especially on tweets that can serve as true market research. This survey paper aims to define and differentiate the aforementioned techniques, so that the reader can understand the nuanced difference in terms of their application and can refer to this survey as a guideline to choose the appropriate text classification method, based on one's problem statement.

The chapter is organized into various sections. We have used the systematic literature review (SLR) method [1] partly. In each section, we have defined the research questions and answered them accordingly.

Section 2 focuses on the first research question (RQ1), which lists the various possible text analytics methods available, in general. Section 3 provides the know-how on how to decide the optimum text classification method according to one's application (RQ2). Section 4 embodies RQ3 that gives deeper insights into each of sentiment analysis, opinion mining, argument mining, and stance detection, in succession. This section also covers the current research trends in each method. Finally, Sect. 5 contrasts the differences between each of the four aforementioned text analytics methods.

## 2 Text Analytics Methods

*RQ1: What Are the Various Text Analytics Methods?* Text mining and text analysis are expansive umbrella terms portraying a scope of advancements for examining and handling semi-organized as well as unstructured text data. The fundamental thought behind these terms is to apply strong calculations in texts by changing over them into numbers so significant examination can be performed on that information.

Text analysis and text mining are utilized interchangeably. The main distinction is that the previous option is utilized in all the more habitual business settings, while the last option is utilized in before applications, quite life-sciences exploration, and government insight.

As seen in Fig. 1, text analytics can be categorized into various methods, out of which sentiment analysis, link analysis, text mining, summarization, text categorization, and key-phrase identification are the commonly used methods. Additionally, according to our research, text mining can further be classified into various categories, out of which we are focusing on opinion mining and argument mining. Stance detection/classification forms a step of argument mining, which itself is a text classification method.

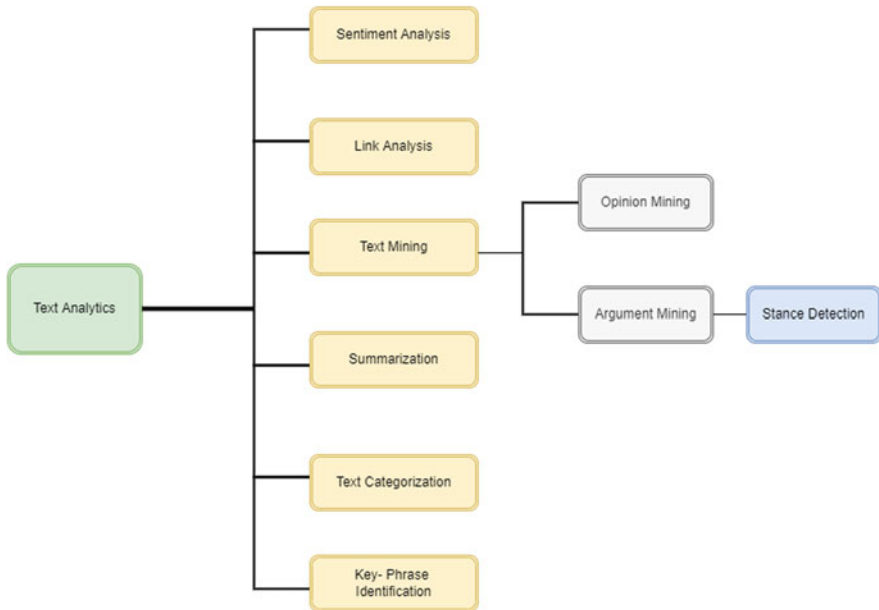


Fig. 1 Text analytics methods

### 3 Choosing Optimum Method

*RQ2: How to Choose the Best Text Classification Method for Your Use Case?* Text classification is an AI procedure that allocates a bunch of predefined classifications to open-ended text. Text classifiers can be utilized to put together, structure, and sort essentially any sort of message—from records, clinical examinations, and documents, and all around the Web. It is one of the major assignments in natural language processing with expansive applications. Thus, to conclude which interaction ought to be followed, a bunch of questions should be answered.

Some of the important questions are listed below:

*Are You Interested in Results About Individual Words or Sentences/Documents?* The focal point of this question is to track down the ideal granularity in your task. This is the greatest division between classes of text mining calculations. To address this question, you should take a gander at the result of your concern—Is it about portraying words or documents? By and large, this includes documents, sentences, “tweets” via virtual entertainment and so on.

Once you have decided the focus of your research, you can proceed with the next question. This chapter focuses on tweet classification so we will proceed with the documents/sentences category, and hence, the next question you need to answer is:

### *Are You Interested in Finding Specific Words or Characterizing the Entire Set?*

Further, there are two ways the text/documents/words can be studied. You can either perform search operations (where you perform search operation on a keyword called as information extraction) or you can perform sorting operations (where you classify your tweets into different categories and classes). The answer to this question further leads you to your desired text analytics process. This question basically answers the use case of your text analytics whether you want to perform search operations on particular keywords or classification on the entire set.

Once you have decided what operations you are going to perform i.e., search or sort, the next important step would be to analyze the kind of information available to you. If you want to perform classification operations (sort), then the kind of dataset available to you plays an important role. So, our next question would be:

*What Kind of Information Is Available?* This question regards the information available during the analysis. If the dataset you have is unlabeled, then you can use some unsupervised algorithms available to perform classification also known as clustering in this case. However, if you have a labeled dataset, then you can use supervised algorithms available that give better and more powerful results. Web content mining algorithms are the ones you can use to perform some specific tasks related to classification. This further consists of different methods to classify the tweets out of which 4 are discussed in detail—sentiment analysis, opinion mining, argument mining, and stance detection. Each of these has different results and can be used based on your use case. This process can be depicted using Fig. 2.

## **4 Exploring a Few Methods**

*RQ3: What Is Sentiment Analysis, Opinion Mining, Argument Mining, and Stance Detection?* This section talks about sentiment analysis, opinion mining, argument mining, and stance detection in detail, covering various research work done till date along with work on datasets, model implementations, and possible scope of more study in the respective fields. We have tried to cover the topics in depth for the reader to understand the current levels and future scope of each of the mentioned methods, thus providing a comprehensive study of each.

### **4.1 Sentiment Analysis**

Sentiment analysis is a famous task in natural language processing, to decide the extremity in a piece of text. A term alludes to the utilization of text analysis and computational semantics to achieve the mentality of a speaker or essayist toward a

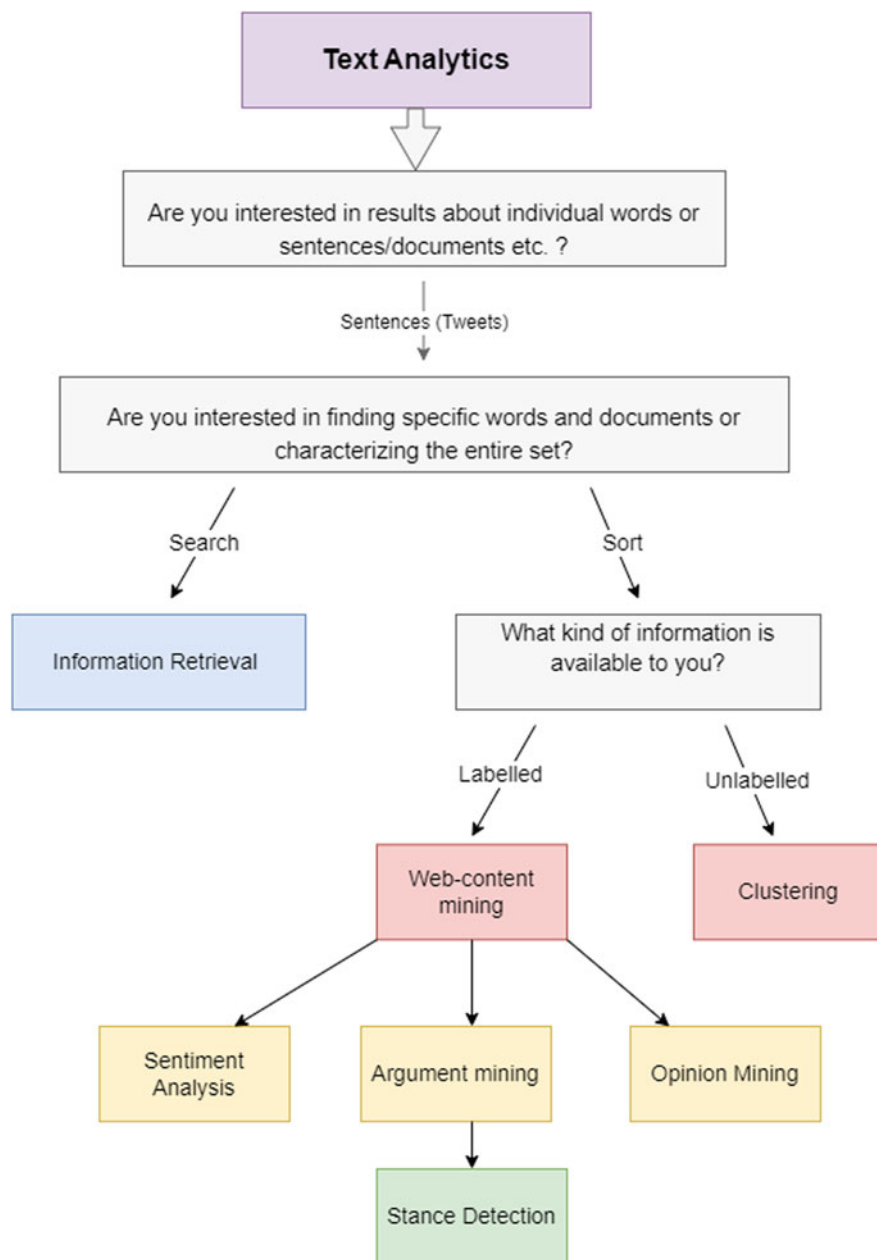


Fig. 2 Decision tree to find the right text classification method [2]

particular theme. Essentially, it decides if a text is communicating sentiments that are positive, negative, or neutral.

Sentiment analysis is generally used to mine abstract data from content on the Web. This is finished utilizing a wide range of procedures, including NLP, measurements, and AI methods. It is another examination field with wide potential for a different genuine application where found assessment data can be utilized to help individuals or associations to settle on better choices.

The sentiment analysis task is considered as a sentiment classification problem. The initial phase in this issue is to extricate and choose text characteristics. A portion of the ongoing work includes:

- Terms presence and recurrence: These elements are individual words or word n-grams and the count of their frequencies. It either gives the words in parallel or uses term recurrence loads to demonstrate the overall significance of these terms.
- Grammatical forms (POS): Tracking down descriptive words, as they are significant marks of suppositions.
- Opinion words and expressions: These are words regularly used to offer viewpoints. Then again, a few expressions offer viewpoints without utilizing assessment words.
- Negations: The presence of negative words might change the assessment direction such as bad is comparable to awful.

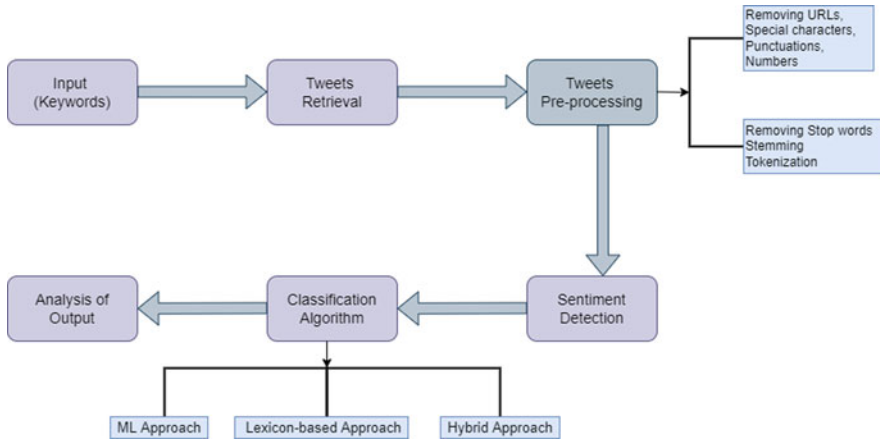
Sentiment analysis methods can be separated into ML approach, lexicon-based approach, and hybrid approach. The machine learning approach (ML) applies the ML calculations and utilization features. The lexicon-based approach depends on a sentiment dictionary and precompiled sentiment terms. It is separated into word-reference-based approaches and corpus-based approaches that utilize factual or semantic methods to track down sentiment extremity. The hybrid approach consolidates the two methodologies.

Few real-world applications are listed below [3]:

- Market and FOREX rate prediction—Concentrating on market expectation by enlarging Twitter sentiment to monetary information.
- Box office prediction—To decide the accommodation of reviews used to become familiar with the relationship between vector of features and the nature of surveys.
- Business analytics—Helps in grouping the items into messages in either positive or negative classes.
- Recommender system—Helps to get a rundown of designated clients at the ongoing stage, and appropriate ways for data dissemination. Based on that, a suggested rundown of designated clients for underwriting an ad can be created.

*Sentiment Analysis on Twitter Data* Figure 3 shows the workflow for sentiment analysis. The system consists of the four main modules: data collection module, data processing module, classification module, and analysis of output.

Table 1 shows the datasets available for sentiment analysis on Twitter [4].



**Fig. 3** The workflow of sentiment analysis

**Table 1** The workflow of sentiment analysis

Dataset name	Technique used
Customer Review Twitter Dataset	Naive Bayes
	Maximum entropy
	SVM
	Semantic analysis (WordNet)
Twitter posts about electronic products	Naive Bayes
	SVM
	Maximum entropy
11,875 manually annotated tweets	Unigram
	Senti-features Kernel

The workflow can be explained as:

1. Input: Choose a subject; then we will gather the tweets with that catchphrase and perform sentiment analysis on those tweets.
2. Tweets retrieval: Tweets can be an organized, semi-organized, and unstructured sort.
3. Tweets preprocessing: Data pre-handling means that we are separating the information to eliminate the deficient, noisy, and conflicting information. The tasks engaged with pre-handling are referenced in flowchart above.
4. Sentiment detection: The principal task in sentiment analysis is arranging the extremity of the given tweets. The three sorts of polarities are: positive, negative, and neutral.
5. Classification algorithm: Various ML algorithms, unsupervised learning, and lexicon-based algorithms can be applied to the tweets as seen in the flowchart.

Sentiment analysis is still an open field for research. Naive Bayes and support vector machines are the most often involved ML calculations for taking care of analysis issues. They are considered as a kind of perspective model where many proposed calculations are contrasted with.

## 4.2 *Opinion Mining*

Opinions influence most of the human behaviors. Hence before making any important decision, we tend to know people's opinion on the topic. Every organization and business want this information about their products and services. This feedback helps them grow better. Even users (individuals) prefer knowing the existing opinions on a particular product or service before purchasing it.

All types of organizations and businesses conduct surveys, feedback forms, and opinion polls about their product to get public opinion. This practice is always beneficial to the company in various aspects such as marketing, public relations, etc. In the Web 2.0 era and immense engagement over social media, organizations are trying their best to extract useful information that would eventually help in crucial decision-making. There are many useful product reviews available on various public forums for people to look up and make a decision before any purchase. Organizations benefit from such public opinions available in any social media platform. But studying this data and extracting useful information from it can be a difficult task because of the escalation of different sites. Each one of them consists of a huge number of opinionated texts that might be difficult to interpret. Any individual would find it difficult to identify relevant sites and conclusive opinions in them. Hence, automated opinion mining systems are needed.

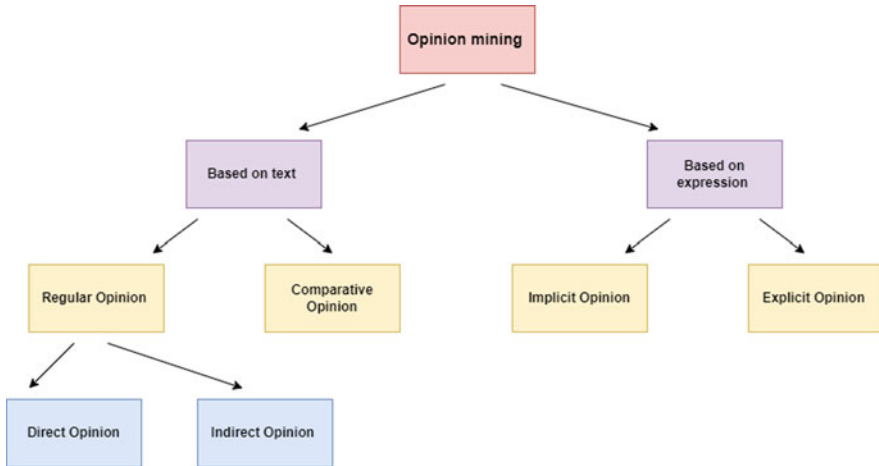
Opinion mining aims "to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document. The attitude may be his or her judgment or evaluation."

Opinion mining covers a scope of themes, for example, expectation of market size for specific administrations and items, buyer response, subjectivity or objectivity recognition, order of sentiment extremity at word level, sentence level or record level, as well as viewpoint opinion pair extraction [5].

Sentiment analysis and opinion mining are equivalent to one another. Sentiment analysis will be analysis of one's sentiments, assessments, perspectives, and feelings toward substances. There are likewise many names and marginally various errands, e.g., sentiment analysis, opinion mining, opinion extraction, sentiment mining, subjectivity analysis, influence analysis, feeling analysis, survey mining, and so on. Be that as it may, they are presently all under the umbrella of sentiment analysis or opinion mining [6].

### **Different Types of Opinions**

So far, we have discussed regular opinion [7]. There are many other types of opinion mining as shown in Fig. 4. Another type is called comparative opinion [8]. In fact,



**Fig. 4** Types of opinion mining

we can also classify opinions based on how they are expressed in text, explicit opinion, and implicit (or implied) opinion. Each type is discussed below.

#### *Regular and Comparative Opinions*

*Regular opinion:* A regular opinion is frequently referred to just as an opinion in the writing, and it has two primary sub-types:

1. Direct opinion: An opinion expressed directly on a substance or part of an element, e.g., “The food tastes delicious.”
2. Indirect opinion: An indirect opinion is an opinion that is communicated in a way on a substance or part of an element in view of its impacts on a few different elements. This sub-type frequently happens in the clinical space. For instance, the sentence “After taking the vaccine, my eyes felt puffy” portrays an unfortunate impact of the vaccine on “my eyes,” which by implication offers a negative perspective or sentiment to the vaccine. For the situation, the substance is the vaccine, and the perspective is the impact on the eyes.

A significant part of the flow research centers around direct opinions. They are easier to deal with. Indirect opinions are frequently more difficult to manage.

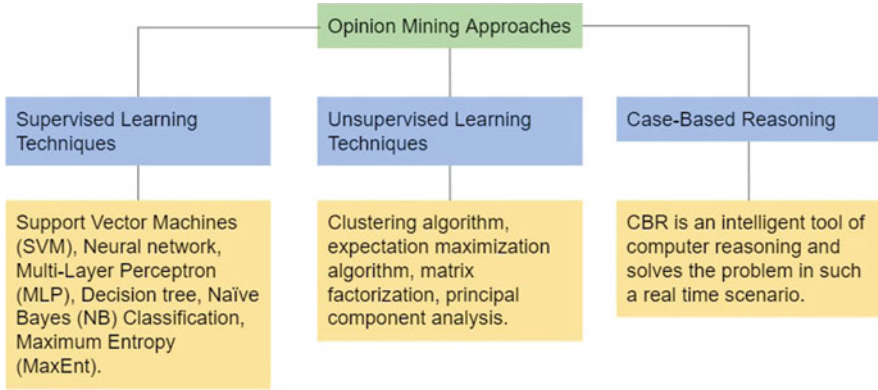
*Comparative opinion:* A comparative opinion communicates similarities or contrasts substances [8]. For example, the sentences “Green Lays are better than blue” and “Green Lays are the best” express two comparative opinions.

#### *Explicit and Implicit Opinions*

*Explicit opinion:* An explicit opinion is a subjective statement giving comparative opinion, e.g., “Green Lays are better than blue” and “Green Lays are the best.”

*Implicit (or implied) opinion:* An implicit opinion is an objective statement implying comparative opinion. Such an objective explanation generally communi-





**Fig. 5** Opinion mining approaches

icates an attractive or unwanted truth, e.g., “The camera quality of Iphones is better than Redmi.”

Explicit opinions are more straightforward to distinguish and to group than implicit opinions. Generally, less work has been done on implicit opinions [9].

### Opinion Mining Techniques

There are mainly three types namely supervised learning techniques, unsupervised learning techniques, and case-based reasoning. All the algorithms in the respective techniques are mentioned in Fig. 5.

The main applications of opinion mining can be hugely in the fields of:

- Opinion spam detection
- Purchasing product or service
- Quality improvement in product or service
- Marketing research

### 4.3 Argument Mining

Argument mining (AM) is one of the famous exploration subjects in natural language processing (NLP). It is the undertaking of distinguishing and removing the derivation structure and the thinking behind them from the regular language text or argumentative designs. By “Argument,” we mean the introduction of an explanation or reasons on the side of some activity or guarantee.

Argument mining helps us to decide positions held by individuals across different areas and the explanations for the equivalent. It is adequately proficient to find the data expected to recognize supports for normal conclusions and for refinement of normal assessment mining calculations over complex issues also. Social media is a huge stage for individuals to set up conclusions, realities, counterfeit news, and any remaining sorts of data according to their advantage. Utilization of argument

mining methods to such a huge broadened information is a difficult assignment. Additionally, consider different dialects used to compose posts and messages via Web-based entertainment stages. We likewise go over the circumstances where we want to recognize convictions and undeniable realities and identify sources coursing the data about these realities to permit confirmation of the base of current realities. Argument mining is additionally equipped for doing the two realities acknowledgment and source distinguishing proof. In particular, the terms guarantee and proof structure the two center parts of an argument: a case is the angle to a theme, while proof alludes to the steady or contradicting data concerning the case.

Recently referenced strategies: feeling examination, assessment mining, subject displaying, and other text investigation procedures become lacking to meet the illustrative necessities of conversational information. Certain difficulties exist that limit the utilization of argument mining in true applications. These are fundamentally exceptionally costly calculation prerequisites or human comment necessities, neither of which are adaptable to a constant investigation setting where enormous volumes of information alongside the absence of solid information sources and casual language typically win. At last, another open test manages the multilingualism of texts as without a doubt, not very many methodologies handled the issue of applying argument mining methods to texts in other normal dialects than English.

*Argument Mining on Tweets* These issues have hindered the development of the task, but the work done till date also proved its feasibility. Work on specifically tweet-based argument mining not only provides identification, extraction, and analysis of sub-groups of argumentative texts but also it has become a field of interest for the broader argument mining community as more and more innovative approaches are being put on the table for testing during Twitter-specific argument mining system development. Corpus annotation becomes a severe bottleneck for application of argument mining. Successful model training is highly dependent on well-annotated data. The foremost work for detecting arguments in tweets is identifying argument components on full tweet level. This is also the precondition for tasks such as argument graph building. No statistics exist on tweet-based argumentation.

A mega tweet corpus annotated for argumentation [10], called Dataset of Arguments and their Relations on Twitter (DART), was developed to aid to the extraction of argument components and their relations. The dataset contains 4000 English tweets on four topics related to politics (e.g., the Grexit and Brexit) and product releases.

We distinguish the accompanying reasonable AM tasks:

1. Argument detection: An extensive number of tweets are non-argumentative in nature, and fostering a framework that can isolate argumentative from non-argumentative tweets seems, by all accounts, to be a legitimate beginning stage. To be sure, most examinations focused on argument recognition somewhat. As most informational indexes depend on full tweet explanations, moving toward this undertaking through administered grouping is a typical decision, in spite

of the fact that distinctions exist concerning the really utilized characterization calculation.

Current ways to deal with general argument location are typically founded on administered characterization.

Different calculations are executed for this including SVM (support vector machine), LR (logistic relapse), CNN (convolutional brain organizations), RNN (intermittent brain organizations), Naive Bayes (NB), random forest, and decision tree models as well as current methodologies such as BERT (bidirectional encoder portrayals from transformers). LR put together methodology with respect to the DART corpus yielded a F1 score of 0.78. The examinations uncover that SVM, LR, and XGBoost models yielded best outcomes, individually. Generally utilized highlight sets incorporate lexical, etymological, or Twitter-related highlights. Later methodologies are likewise founded on BERT embeddings, which, as indicated by different investigations, also yield nice exactness. One such significant work is [11].

An overall argument identification is an underlying advance that addresses a significant channel for downstream errands in an AM pipeline. Notwithstanding, more fine-grained part identification is required in the event that more point-by-point argument structures are to be separated.

2. Claim detection: Claim identification is the assignment of distinguishing suppositions and stances concerning a subject. It might be applied very well to an informational index as of now pre-sifted by argument identification or as an underlying advance itself (contingent upon the utilization case). Also, it could possibly be drawn nearer alongside proof identification. As in argument recognition, the existing ways to deal with guarantee discovery are intensely founded on administered characterization and arrangement naming. A few past works in all actuality do contain comments for the classes guarantee, guarantee with proof, and endlessly counterclaim with proof. In this way, proof is just viewed as in mix with a case and cannot be identified autonomously.

Here too, algorithms such as NB, SVM, and DT have been widely used for claim detection. Pretrained BERT document embedding has also been utilized for classification tasks.

3. Evidence (type) detection: We characterize proof identification as the errand of recognizing proof proclamations given regarding a specific case. While proof does not solely incorporate genuine proclamations, past work likewise centered around truth acknowledgment. Proof discovery was distinguished as a center errand in tweet-based AM. Like argument and guarantee discovery, current methodologies depend on regulated grouping and arrangement marking. In particular, SVM, LR, XGBoost, and CRF models were utilized.
4. Relation detection: The majority of work focuses on the distinguishing proof of argument parts, and first work exists that looks at argument relations, for example support/assault. Past concentrate on DART dataset explored relation detection as the third step of their AM pipeline. Tests depended on the relation explanation layers of the DART corpus that rely upon recently made tweet matches. To start with, the creators moved toward the undertaking utilizing textual entailment,

given the reasonable closeness between help/entailment and assault/logical inconsistency, individually. They applied a brain order approach in view of a long short-term memory (LSTM)-driven encoder–decoder organization. In any case, the outcomes were uninspiring (F1 (support): 0.20; F1 (assault): 0.16), which were credited to the trouble of the errand. The creators called attention to the test of matching tweets reasonably and managing the intricacy of some tweet’s substance.

So far, tweet-based AM has been primarily focused on conventional ML techniques. Development of neural network architectures and deep learning approaches has begun in this field recently.

#### 4.4 *Stance Detection*

As discussed in the previous section, an argument is defined to be consisting of two elements, namely a target and a stance. So, in this survey paper, we ignore the target detection step and focus on stance classification, which enables us to use techniques from text mining and sentiment analysis [12]. “Stance is defined as the expression of the speaker’s standpoint and judgement towards a proposition” [13].

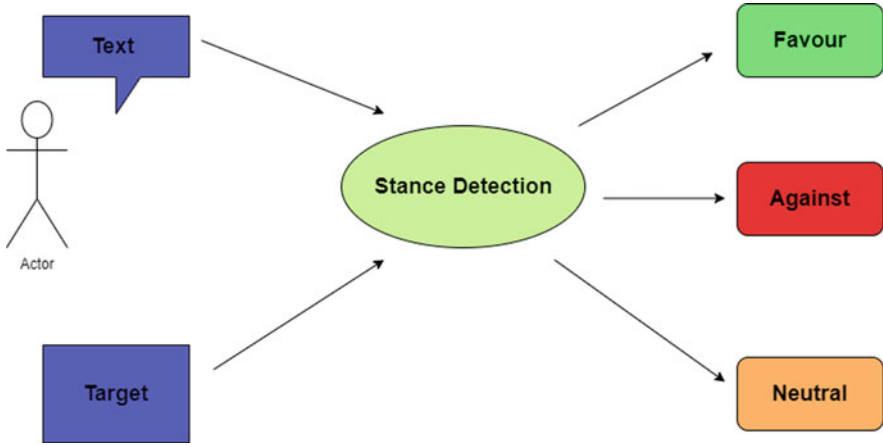
Many go against this definition of stance-taking, by contending that it is a subjective issue. As the process of taking a stance is complex, due to its various individual, social and cultural perspectives [14].

Stance detection has many names, also known as “Stance classification” [15], “stance identification” [16], “Debate side classification” [17], and “debate stance classification” [18].

As a formal definition according to [19], stance detection is defined as: “For an input in the form of a piece of text and a target pair, stance detection is a classification problem where the stance of the author of the text is sought in the form of a category label from this set: Favor, Against, Neither. Occasionally, the category label of Neutral is also added to the set of stance categories and the target may or may not be explicitly mentioned in the text” [20].

If we state this definition in other words, stance detection is predicting one’s stance given a target (what we are interested in), to be either in favor, against, or having a neutral stance toward the target. Figure 6 provides a schematic representation of the aforementioned definition. Prior work focused on analyzing debates, which now evolved to focus more on social media platforms, especially Twitter.

*Stance Detection Datasets* Despite stance detection being a recent research topic, considerable effort is devoted to the creation of stance-annotated datasets, most of which are made publicly available. The available stance detection datasets are categorized as classification and prediction datasets, wherein the classification datasets are further categorized as target-specific, multi-target, and claim-based stance dataset. Küçük and Can [19] give a table wherein all the stance detection



**Fig. 6** Schematic representation of the stance detection procedure

datasets are grouped, consisting of all the domains. Since our survey mainly focuses on the stance detection on tweets, the following table depicts the stance detection datasets where the domains are tweets in Table 2.

*Stance Detection Approaches* This section discusses the major machine learning algorithms used for stance detection, which can be classified into supervised learning, weakly supervised/transfer learning, and unsupervised stance detection.

Work in the supervised learning domain gained momentum with the SemEval stance dataset (Task A). In general, algorithms such as the ones mentioned in Fig. 7 are used frequently. For transfer learning, the knowledge that an algorithm has learned from one task is applied to a separate task, such that in the task of stance detection, the learning from one target can be applied to a different target. This helps to address the problem of scarce annotated dataset. For unsupervised stance detection models, clustering techniques are primarily used [20–23].

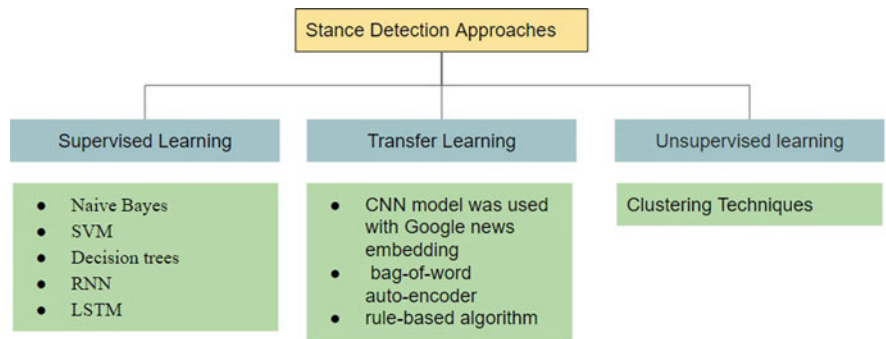
The best performing algorithms as observed over the years are the supervised algorithms, as they are more effective than transfer learning approaches. It is interesting to see that simpler machine learning algorithms are more effective than deep learning models.

The main applications of stance detection can be hugely in the fields of:

- Social sensing technique
- Analyze attitudes of a campaign or after disruptive events
- Solve algorithmic issues on social media
- Fake news detection
- Rumor detection

**Table 2** Stance detection tweet datasets

Authors	Target(s)	Size
Mohammad et al. [24]	Atheism, Climate change is a real concern, Feminist movement, Hillary Clinton, Legalization of abortion, Donald Trump	4870 tweets
Mohammad et al. [24]	Atheism, Climate change is a real concern, Feminist movement, Hillary Clinton, Legalization of abortion, Donald Trump	4870 tweets
Taulé et al. [25]	Independence of Catalonia	5400 tweets in Spanish and 5400 tweets in Catalan
Sobhani et al. [26]	{Clinton-Sanders}, {Clinton-Trump}, {Cruz-Trump}	4455 tweets
Küçük [27]	Galatasaray, Fenerbahçe	700 tweets
Küçük and Can [28]	Galatasaray, Fenerbahçe	1065 tweets
Derczynski et al. [29]	Rumorous tweets	5568 tweets (4519 + 1049)
Swami et al. [30]	Demonetization in India in 2016	3545 tweets
Lai et al. [31]	2016 referendum on reform of the Italian Constitution	993 triplets (2889 tweets)
Lozhnikov et al. [32]	Claims extracted from news and tweets	700 tweets and 200 news articles
Mutlu et al. [33]	Use of “Chloroquine” and “Hydroxychloroquine” for the treatment or prevention from the coronavirus.	14,374 tweets



**Fig. 7** Stance detection approaches

**Table 3** Examples of tweet/target pairs from the COVID-19-Stance dataset

Tweet	Target	Stance	Opinion	Sentiment
I don't know what to say, you're dumb if you refuse to put on a mask. This is not something to be political about. #MaskUp	Wearing a Face Mask	In Favor	Explicit	Negative
OMG the video. Trump will lose, but I can't recall the last time I watched such a cynical piece of propaganda. Keeping schools closed will be a bad idea for our most children	Keeping Schools Closed	Against	Explicit	Negative
I believe in SCIENCE, I wear a mask for YOUR Safety. #JoeBidenForPresident2020	Anthony S. Fauci, M.D.	In Favor	Implicit	Positive
"@realDonaldTrump23 No. of Deaths are down REAL BIG! United States of America is ready to rock"	Stay at home orders	Against	Implicit	Positive

**Table 4** Comparison table

Sentiment analysis	Opinion mining	Argument mining	Stance detection
Answers whether text is positive, negative, or neutral. It gives the net sentiment	Answers which part is opinion expressing; Who wrote the opinion; What is being said	Goal is to extract arguments and relations from a given text	Goal is to determine from text alone an author's stance (i.e., whether for/against/none)

## 5 Differentiating the Aforementioned Methods

Table 3 shows a comparison on the results of stance, opinion, and Sentiment analysis. The analysis is performed on various targets. On each target, a sample tweet is studied. The results are presented as: for stance, if the tweet is in favor of the target or against the target; for opinion, if the tweet is explicitly or implicitly talking about the target; for sentiment, if the tweet mentions a negative or positive sentiment irrespective of the target. Table 4 compares each of the four text analytics methods in layman language.

## 6 Conclusion

This chapter presents a complete, cutting edge audit on the exploration work done in different parts of text analytics techniques. This survey paper talked about various procedures of sentiment analysis, opinion mining, argument mining, stance detection, and the system to perform message investigation.

The investigation of Twitter information is being done in various perspectives to mine the assessment or feeling, and the position that is created from tweet

examination is generally being concentrated in the natural language processing area. While performing Twitter information investigation, it is important to be aware of extricating the tweets, its construction, and their importance. Subsequently, the fundamental data expected to do tweets analysis is all around talked about in this survey paper. The literature shows that the accuracy is further developed when the stance detection calculations are followed up by the AI methods, such as SVM, Naïve-Bayes, and LSTM.

By and large, these text analytics strategies have observed different promising applications. A ton of work has been done, and there is degree for more in this field as recent fads and arising applications in this space keep the region drawing in and inclined to improvement.

## References

1. Kitchenham, B. and Charters, S., 2007. Guidelines for performing systematic literature reviews in software engineering.
2. Miner, G., Elder IV, J., Fast, A., Hill, T., Nisbet, R. and Delen, D., 2012. Practical text mining and statistical analysis for non-structured text data applications. Academic Press.
3. Wagh, R. and Punde, P., 2018, March. Survey on sentiment analysis using Twitter dataset. In 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA) (pp. 208–211). IEEE.
4. Medhat, W., Hassan, A. and Korashy, H., 2014. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4), pp.1093–1113.
5. Yun, Y., Hooshyar, D., Jo, J. and Lim, H., 2018. Developing a hybrid collaborative filtering recommendation system with opinion mining on purchase review. *Journal of Information Science*, 44(3), pp.331–344.
6. Liu, B., 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1), pp.1–167.
7. Liu, B., 2011. *Web data mining: exploring hyperlinks, contents, and usage data* (Vol. 1). Berlin: Springer.
8. Jindal, N. and Liu, B., 2006, July. Mining comparative sentences and relations. In *AAAI* (Vol. 22, No. 13311336, p. 9).
9. Zhang, L. and Liu, B., 2011, June. Identifying noun product features that imply opinions. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (pp. 575–580).
10. Bosc, T., Cabrio, E. and Villata, S., 2016, May. DART: A dataset of arguments and their relations on Twitter. In *Proceedings of the 10th Edition of the Language Resources and Evaluation Conference* (pp. 1258–1263).
11. Schaefer, R. and Stede, M., 2020, December. Annotation and detection of arguments in tweets. In *Proceedings of the 7th Workshop on Argument Mining* (pp. 53–58).
12. Clos, J., Wiratunga, N., Massie, S. and Cabanac, G., 2016. Shallow techniques for argument mining.
13. Biber, D. and Finegan, E., 1988. Adverbial stance types in English. *Discourse processes*, 11(1), pp.1–34.
14. ALDayel, A. and Magdy, W., 2021. Stance detection on social media: State of the art and trends. *Information Processing & Management*, 58(4), p.102597.



15. Walker, M., Anand, P., Abbott, R. and Grant, R., 2012, June. Stance classification using dialogic properties of persuasion. In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 592–596).
16. Zhang, S., Qiu, L., Chen, F., Zhang, W., Yu, Y. and Elhadad, N., 2017, April. We make choices we think are going to save us: Debate and stance identification for online breast cancer CAM discussions. In Proceedings of the 26th International Conference on World Wide Web Companion (pp. 1073–1081).
17. Konjengbam, A., Ghosh, S., Kumar, N. and Singh, M., 2018, September. Debate stance classification using word embeddings. In International Conference on Big Data Analytics and Knowledge Discovery (pp. 382–395). Springer, Cham.
18. Hasan, K.S. and Ng, V., 2013, October. Stance classification of ideological debates: Data, models, features, and constraints. In Proceedings of the Sixth International Joint Conference on Natural Language Processing (pp. 1348–1356).
19. Küçük, D. and Can, F., 2020. Stance detection: A survey. *ACM Computing Surveys (CSUR)*, 53(1), pp.1–37.
20. Trabelsi, A. and Zaiane, O., 2018, June. Unsupervised model for topic viewpoint discovery in online debates leveraging author interactions. In Proceedings of the International AAAI Conference on Web and Social Media (Vol. 12, No. 1).
21. Darwish, K., Stefanov, P., Aupetit, M. and Nakov, P., 2020, May. Unsupervised user stance detection on Twitter. In Proceedings of the International AAAI Conference on Web and Social Media (Vol. 14, pp. 141–152).
22. A. Joshi, P. Bhattacharyya, M. Carman, Political issue extraction model: A novel hierarchical topic model that uses tweets by political and non-political authors, in: Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, 2016, pp. 82–90.
23. R. Ammar, K. Mucahid, D. Kareem, E. Tamer, B. Cansin, Embeddings based clustering for target specific stances: The case of a polarized Turkey, ICWSM2021 (2021).
24. <http://www.saifmohammad.com/WebPages/StanceDataset.htm>
25. <http://stel.ub.edu/Stance-IberEval2017/data.html>
26. [http://www.site.uottawa.ca/~diana/resources/stance\\_data/](http://www.site.uottawa.ca/~diana/resources/stance_data/)
27. <https://github.com/dkucuk/Stance-Detection-Turkish-V1>
28. <https://github.com/dkucuk/Stance-Detection-Turkish-V3>
29. <https://s3-eu-west-1.amazonaws.com/downloads.gate.ac.uk/pheme/semEval2017-task8-dataset.tar.bz2>
30. [https://github.com/sahilswami96/StanceDetection\\_CodeMixed](https://github.com/sahilswami96/StanceDetection_CodeMixed)
31. <https://github.com/mirkolai/Stance-Evolution-and-Twitter-Interactions>
32. <https://github.com/npenzin/rustance>
33. <https://github.com/eceveco/COVID-CQ/blob/master/COVID-CQ.csv>

# A Survey on Threat Intelligence Techniques for Constructing, Detecting, and Reacting to Advanced Intrusion Campaigns



Ashutosh Anand, Mudit Singhal, Swapnil Guduru, and B R Chandavarkar

## 1 Introduction

An intrusion is an adversary's action or threat against a particular individual, entity, or organization that can lead to security and privacy violations. An intrusion campaign [1] is a set of these intrusions by an adversary against a particular organization. The common intention for attempting an intrusion campaign is for stealing data, breaching privacy, or stealing money.

Traditionally, most intrusion attempts were detected by administrators and security analysts by continuously monitoring different interfaces such as network traffic, web requests, Operating System logs, etc. [2]. This made detecting intrusion attempts slower and a gruesome task. Later, these approaches were substituted with Intrusion Detection Systems (IDSs) [3], an automated model that detects known intrusion attempts based on static properties. Although the model was much quicker in detecting intrusions, it would fail in classifying innovative and new intrusion attempts. Hence, a lot of research has been involved in standardizing threat intelligence [4] techniques, which makes use of developing technologies and delivers a framework for organizations to implement and secure their services and products.

There are currently several definitions of threat intelligence, but it can be mainly defined as “a collection of steps taken to detect and respond against an intrusion attempt based on evidence collected from various data points” [4]. A threat intelligent model will help organizations choose the most important parameter required to defend against the most probable intrusions. It does this by:

---

A. Anand (✉) · M. Singhal · S. Guduru · B. R. Chandavarkar  
Department of Computer Science and Engineering, National Institute of Technology Karnataka,  
Surathkal, India

- Bringing awareness of potential adversaries and how they might attack the organization
- Implementing secure practices and using secure software to reduce risk of intrusion
- Using the right software to automatically detect intrusion attempts
- Practicing the correct and efficient response strategy to combat an intrusion/attack

This chapter talks about different techniques and methods required to construct threat intelligence Platform for an organization by including the various parameters to be overlooked. The chapter gives a detailed exploration of the problem and challenges faced. It then gives a detailed description of how to detect the intrusion campaigns and covers different kinds of detection systems, followed by a brief comparison of certain methods. At last, it shifts focus on how to react to the intrusion campaigns, if detected, touching upon different comparison features followed by a brief survey of recent research regarding the topic.

Section 2 of the chapter talks about various tools, methods, and resources to construct a threat intelligent platform, after which different ways and techniques to detect intrusions are shown in Sect. 3. Section 4 talks about the practices to be followed to respond and react to an intrusion once detected.

## **2 Constructing Threat Intelligent Platform**

One of the initial steps in building a threat intelligence technique is constructing a platform that has access to information about threats and attacks and being aware of its impacts on different levels. The exchange of evidence-based knowledge about actual or potential threats across companies and authorities is known as threat intelligence sharing. This section talks about the different approaches used in building such a threat intelligence platform. Thus, various parameters are looked at while constructing a threat intelligence platform.

### ***2.1 Reasons for Threat Intelligence***

The first step to constructing a threat intelligent model is questioning the need for securing the company and identifying key areas to be secured [5]. For instance, a banking company would like to secure a customer's digital transaction to protect the financial health of their users, while a governmental organization would have to secure military information to prevent leaking confidential information. Hence, the organization should understand the starting and ending (target) points of an attack, which would help organizations understand the possible intrusions and its severity.

## ***2.2 Identifying Threat Actors***

An organization's threat intelligence platform should be aware of the type of adversaries. Adversaries present themselves as groups (commonly called Hacker Groups) or individuals or can even be the members of the organization's team (insider threats). These potential adversaries are the threat actors [6] of the organization. The purpose of identifying these threat actors is for the organization to stay vigilant and a step ahead of them.

Famous hacker groups make use of social media and online chat forums to promote their ideas. To make use of this, Huang et al. [7] proposed a dynamic vulnerability threat assessment model using Machine Learning and Text Mining approach to predict attacks from social media content analysis.

## ***2.3 Tactics, Techniques, and Procedures***

TTPs (tactics, techniques, and procedures) are actions and assets that an adversary uses to execute attacks successfully. The MITRE ATT&CK [8] is a popular open and freely available framework serving knowledge related to popular TTPs which help in constructing threat intelligence models and techniques.

V. Legoy et al. [9] proposed that different TTPs can be extracted from the ATT&CK framework and can be displayed in valid data representation techniques from textual cyber reports using text multi-label classification. The author further developed an interactive rcATT tool for automated analysis of cyber threat reports from their findings to help experts find ATT&CK TTPs from content of the text.

## ***2.4 Indicators of Compromise***

Indicators of Compromise (IoCs) are forensic pieces of evidence and reports that help in identifying malicious activities. These are static data of attacks and can contain information about the type of attack executed. Organizations have to keep collecting and storing IoCs from different resources. Zeng et al. [10] proposed a Dark Web crawler to collect Topics of Interest (ToI) from hacker websites and communities; this can be further used to update the IoC detection database. Zhi Liu et al. [11] make use of Query Committee Inconsistency (QCI) to detect IoC from internal logs for organizations that have a bigger database for IoCs.

## 2.5 *Threat Intelligent Sharing*

Cross-organizational threat intelligence sharing for collecting and analyzing threats is known as Threat Intelligence Sharing (TIS) [12]. TIS always enhances greater participation of organizations to share threat intelligence resources like IoCs, threat actor details, etc. Organizations participating in TIS thus benefit in more efficiently identifying new threats and helping construct their threat intelligence techniques. Since sharing information across independent entities can be confusing, Structured Threat Information eXpression (STIX) [13] standard, which is a widely known formatting and language-driven tool used for structurally representing different types of IoCs, can be used to bridge such gaps.

## 2.6 *Current Limitations and Challenges*

One of the main purposes of constructing a threat intelligent system is to be able to generate resources and information required for steps like detecting and reacting to intrusion systems. Unfortunately, most of these resources like TTPs, Threat Actors, IoCs, etc. change and update very frequently, rendering the previously owned resources useless over time.

Most organizations do not participate in threat intelligence sharing protocols since they may have to share private and confidential data to other organizations while sharing threat details. Among the threat actors, Advanced Persistent Threats (APTs) [14] are considered to be one of the more dangerous threat actors who usually remain stealthy in the systems for a long period.

While having an efficient threat intelligence model is important for an organization's safety and cyber-defenses, it can be observed that implementation of such models generally requires high investments, especially for security teams and resources, which many small-scale organizations may not have.

## 3 **Detection of Intrusion Attempts**

Intrusion Detection System (IDS) [3] is a system or a software that captures event and security logs for the purpose of detecting an intrusion attempt in a system or network. An intrusion campaign once detected by an IDS is reported to the organization, where actionable procedures are followed to prevent the damages that can be caused by the intrusion. The tools and resources provided during the construction of the threat intelligence model are used here. An IDS is generally divided into two groups based on their different detection methods and positioning within the system.

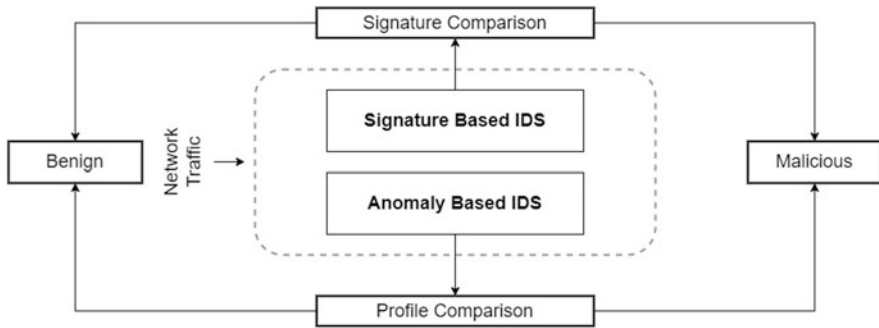


Fig. 1 Workflow of signature-based and anomaly-based IDSs [15]

### 3.1 IDS Technology Types Based on Their Detection Method

Intrusion detection techniques can be classified into two major categories based on detection methods. They are signature-based methods or anomaly-based methods. The flow of both methods is depicted in Fig. 1

#### 3.1.1 Signature-Based Intrusion Detection System (SIDS)

Signature intrusion detection systems (SIDSs) [16] are based on pattern matching techniques to find a known attack. In other words, whenever an intrusion signature resembles a signature from a prior intrusion that already prevails in the signature database, an alert signal is generated. For previously known intrusions, SIDS usually provides excellent detection accuracy. However, on the other hand, it has a hard time identifying zero-day attacks since no matching signature exists in the database until the new attack's signature is extracted and stored.

#### 3.1.2 Anomaly-Based Intrusion Detection System (AIDS)

Anomaly-Based Intrusion Detection System (AIDS) [17] was introduced to detect unknown malware attacks as new pieces of malware are developed rapidly. Using machine learning, statistical-based, or knowledge-based methodologies, a standard model of a computer system's behavior is built-in AIDS. A substantial discrepancy between observed behavior and the model, which might be interpreted as an invasion, is characterized as an anomaly. The main limitation of AIDS is that the systems' performance depends largely on the rules that are to be defined on each interface of the system, which can further increase the complexity in implementation.

## **3.2 *IDS Technology Types Based on Their Positioning Within the Computer System***

### **3.2.1 Network Intrusion Detection Systems (NIDS)**

Network-based intrusion detection systems (NIDSs) [18] are IDS that capture traffic within a network (in a passive manner) and analyze them to find occurrences of possible intrusion campaigns. NIDSs frequently feature two network interfaces, one being used for promiscuous network listening and the other for control and reporting. An NIDS can degrade efficiency within the network since it has to capture, store, and analyze the traffic, hence creating “choke-points.”

### **3.2.2 Host Intrusion Detection Systems (HIDSs)**

A host-based intrusion detection system (HIDS) [19] is a system that monitors the computer system on which it is deployed for signs of intrusion and/or misuse, subsequently logs the behavior, and notifies the appropriate authorities. A HIDS can be regarded as an agent that monitors and analyzes if anything or anyone, internal or external, has compromised the security policy of the system. Since the HIDS is implemented on a singular system, it may have limitations in passing intrusion-based information to other devices in the network, if the system were ever attacked.

### **3.2.3 Protocol-Based Intrusion Detection Systems (PIDSs)**

A protocol-based intrusion detection system (PIDS) [20] is an IDS that monitors traffic at the protocol layer (usually HTTP requests), that the computer system seems to be utilizing. A PIDS oversees the protocol’s dynamic behavior and state. It is often composed of a system or an agent that resides on the server’s front end that monitors and analyzes communication between connected devices and the system they are safeguarding. Implementing a PIDS over multiple layers of the network can increase complexity and be expensive compared to other IDSs.

### **3.2.4 Hybrid Intrusion Detection Systems**

When two or more IDS technologies are combined, a hybrid intrusion detection system [21] is created. It is more suitable than conventional intrusion detection systems. A hybrid IDS can combine the pros and cons of various IDSs together, and this is hence the suggested IDS model.

### 3.3 Comparison of Model Performances on Public Datasets

Table 1 shows the comparison of different IDS models based on the detection accuracy they generated from the year-wise literature survey. These models were trained on popular Intrusion Detection Evaluation Datasets. Observations of each model are also shown in the table.

**Table 1** Comparison of IDS on various datasets based on the literature survey

Dataset	Result	Observations	Reference
DARPA 98	On using ANN Analysis system calls, the model achieved a detection rate of 96%	An Artificial Neural Network classifier was deployed to prepare and test the network.	McHugh et al. [22] [2000]
	On using Snort's detection, around 69% of total generated alerts were false.	Just SIDS was applied. The performance would have been lot better if AIDS was deployed with it.	Hu et al. [23] [2009]
NSL - KDD	On using Naive Bayes, the model achieved a detection rate of 89%	As the focus is on classifying for the intrusion instances and not calculating exact probabilities, Bayesian classifiers gave a moderate accuracy.	Adebowale et al. [24] [2013]
	On applying kNN algorithm, the model achieved an improved detection rate of 94%	kNNs generally use a search strategy that is based on similarity levels that helped here to improve the accuracy of the model by finding a locally optimal hypothesis function.	Adebowale et al. [24] [2013]
	On using a C4.5 approach, the accuracy improved further to 99%	The feature of the data that has the most efficiency in dividing its sample into subsets is selected by C4.5, hence improving the performance.	Tahseen et al. [25] [2013]
Bot-IoT	SVM model gave the best detection rate of 98%	As SVMs work better when there is a clear margin of separation between classes, it was a wise choice of applying it in the Bot-IoT dataset.	Koroniotis et al. [26] [2018]



## 4 Reacting to Intrusion Attempts

Once the intrusion is detected by the IDS [3], an appropriate action must be taken to mitigate the damage caused by the intrusion. The systems that are developed to react and respond to intrusions are known as Intrusion Response Systems (IRSs) [27]. The main goal of an IRS is to halt, isolate, and remove the presence of intrusions. An IRS also initiates recovery of the system.

### 4.1 Comparison Features

Upon analyzing various strategies and systems in the literature, the authors have identified the following features as benchmarks for further analysis. These features are as follows:

#### 4.1.1 Administrator's Involvement

The feature takes into consideration the requirement of the administrator's input during system building. Certain environments such as VMs might limit the fine-tuning administrators can implement. Based on the literature, reinforcement learning [28] can be used to improve results.

#### 4.1.2 Reaction Type

This feature takes into consideration whether the IRS proactively detects system vulnerabilities and acts against security incidents or acts actively and follows the ongoing attack to respond to it. Based on this classification, two types of IRS can exist and they are as follows:

- Proactive: These generally identify system characteristics, vulnerabilities, and potential threat sources. This can be used by testing tools to fix vulnerabilities.
- Active: These are concerned with systems' ability to respond to ongoing attacks. While it is more flexible than a proactive reaction [29], the active reaction requires more computational power.

While most of the IRSs provide both active and proactive reactions, for this survey, we consider only large networks and will label IRS as proactive or active reactions on large networks.

#### 4.1.3 Scalability

Scalability is the ability of a system to cope under expansion. It also measures the ability of the system to use its resources to their maximum potential. It is one of the most desirable attributes of a network or a system [30].

#### **4.1.4 Time Complexity**

This feature takes into consideration the complexity of the proposed system in terms of time, i.e., time complexity. Often, the complexity of the system can determine its scale and use cases.

#### **4.1.5 Response Policy**

When IRS chose between options and rank options, they consider various metric policies and choose an appropriate response [31]. The metrics vary from system to system and cover a wide range of options from cost effectiveness to security impacts, and often systems have multiple metrics. For this purpose, systems select certain features which are more desirable outcomes and take decisions accordingly. Based on the response policies selected, IRS can be more suitable for certain cases over others.

### ***4.2 Survey of IRS in Literature***

In this subsection, the authors briefly discuss a few of the major IRS in recent years in the literature. Each system is discussed briefly and an analysis is done. The papers are arranged in the order they were published with the papers published earlier appearing before the ones that were published on a later date.

#### **4.2.1 NICE: Network Intrusion Detection and Countermeasure Selection in Virtual Network Systems**

The work by Chung et al. [32] [2013] was one of the first attempts to deploy architecture to react in virtual machines in a cloud environment. The authors of the paper considered the fact that administrators often cannot fine-tune systems on VM's due to reasons such as Service Level Arguments. Due to the nature of this survey, the authors focus on the intrusion response part of the system. Once a vulnerability is discovered, a virtual networking-based pool is built which takes various countermeasures such as intrusiveness of attack and cost and is solved as a Multiple Objective Optimization Problem (MOOP) [33] using a Return of Investment Index (ROI). VM security index or VSI is monitored to apply mitigation strategies on VMs. NICE was centralized with a single point of failure. Its scalability was also limited, and it was not well suited for large-scale distributed systems.

#### **4.2.2 IDAR: An Intrusion Detection and Adaptive Response Mechanism for MANETs**

Nadeem et al. [34] [2014] developed IDAR for mobile ad hoc networks. The intrusion model was developed to facilitate a flexible approach to intrusion response based on confidence levels of the attacks. Previous work by the same authors isolated nodes in a predetermined way for all intrusion attempts [35].

This system takes into consideration the severity of the attack, degradation in network performance, and the impact the response action is expected to have on network performance. The confidence level of the attack is also calculated and taken into account. All network nodes operate in one of the three roles of manager node (MN), cluster heads (CH), and cluster nodes (CNs), and security measures [36] are used to enable communication between them. The administrator is required to select response actions in advance in the form of decision tables. IDAR depends heavily on MN, which makes it a single point of failure. IDAR also suffers from large information gathering and overhead due to regular updates of network.

#### **4.2.3 DOCS: Dynamic Optimal Countermeasure Selection for Intrusion Response System**

Kholidy et al. [37] [2014] proposed DOCS, which is an intrusion response system aimed at selecting optimal countermeasures for a dynamic reaction to threats. A Multiple Objective Optimization Problem is taken which tries to maximize security performance while minimizing the negative impact on services and cost. Attack–defense trees [38] and Service Dependency Graphs [39] are used to model possible attacks to improve results.

The Pareto optimal set is developed to identify the best possible solutions unlike other works and SAW [40] is used to extract optimal solutions following MCDM methods. The countermeasure deployed is evaluated based on a time window of 5 s and effectiveness is stored in databases. The proposed algorithm makes use of only a single attack–defense tree (ADT). However, in real-life scenarios, a forest of ADT's would be required which increases complexity. The generation and maintenance of Service Dependency Graph (SDG) has not been automated completely.

#### **4.2.4 ACIRS: A Risk Mitigation Approach for Autonomous Cloud Intrusion Response System**

Shameli-Sendi et al. [41] [2016] proposed ACIRS which is a defense strategy aimed at cloud network architecture. The authors of this paper took into consideration the distributed system issues of NICE [32] such as single point of failure and scalability in addition to the issues solved by NICE.

Initially, the administrator is required to set asset values for various resources during configuration. This helps the system in prioritizing certain resources. The

system follows the Multiple Objective Optimization Problem approach where it takes into consideration security impact, system risk status, operational cost, and benefit. VM security index or VSI [32] is monitored to apply mitigation strategies on VMs. The proposed model does not take into consideration the goals of the attacker which could improve response taken. A strong assumption has been made by the authors that the hypervisor is secure, but there are various sources in the literature [42] that note attacks on them.

#### **4.2.5 Deep Q-Learning: A Hybrid Model-Free Approach for the Near-Optimal Intrusion Response Control of Non-Stationary Systems**

Cardellini et al. [43] [2020] propose the use of Deep Q-Learning [44], which is a model-free technique leveraging deep reinforcement learning algorithm, to obtain near-optimal control of intrusion response on non-stationary systems.

The concept of transfer learning [45] was applied to run software simulating system models and attaching them to the DRL model which was then connected to real-life systems where it could learn possible gaps and adapt to the evolution of the system. This approach not only reduced the time required for training but also reduced the chances of the proposed system executing wrong or dangerous actions.

The system is based on the Markov Decision Process [46]. Actions are taken based on post-conditions and pre-conditions on states. Thus, the model is trained to adapt to the addition of actions, removal of actions, and change of post-conditions. Administrators are required to fine-tune a considerably larger number of hyperparameters to exploit the full potential of Deep Q-Learning. The proposed model does not have optimized pipelines to deal with multiple attackers simultaneously. Also, the time and the cost of actions are predefined and not dynamically decided based on impact on deployment in the environment.

### ***4.3 Comparison of Selected IRS from Literature***

Table 2 shows a comparison of the major IRS in recent years based on features like administrator's involvement, reaction type, scalability, complexity and its response policy.

Based on the literature survey, the use of standards should be encouraged to streamline the process of quantitative analysis of adopted reaction strategy and response evaluation models [47] can be used to improve results. Intrusion Risk Assessment (IRA) [48] can also be considered to enhance Intrusion Response Systems.

**Table 2** Comparison of IRS present in Literature

Name	Administrator involvement	Reaction type	Scalability	Complexity	Response policy	Reference
NICE	N/A	Proactive	Low	Low	Connectivity, cost, intensiveness	[32] [2013]
IDAR	Formulate decision table for response decision process	Active	Medium	Medium	Network Performance Degradation	[34] [2014]
DOCS	The relative importance of incompatibility cost ( $\gamma$ ) and administration cost ( $\delta$ ) need to be set	Active	High	Medium	SAW based on security benefit, security impact, security cost	[37] [2014]
ACIRS	Asset value is to be set which represent resource value	Active	High	Medium	Operational cost, security impact, security risk status	[41] [2016]
Deep Q-Learning	Hyper parameters of Deep Q-Learning i.e., number of hidden layers of $\gamma, \delta, \zeta$ . Action set is created with associated time and cost	Active	High	High	Reward function is characterized by a penalty on action dependent on normalized time and cost values	[43] [2020]

## 5 Conclusion

The chapter discussed the critical parameters required to construct a threat intelligence technique and the survey of different approaches by researchers to gather resources. For Intrusion detection systems, different datasets were referenced with their various respective implementations, which were then compared based on their performance, with key observations noted. For Intrusion Response Systems, features were identified to compare various methods, and important IRS were picked from the literature. The models were explained briefly and then compared on various features which help to choose the appropriate IRS based on requirements.

For future work, it could be worthwhile to survey various Intrusion Prevention Systems (IPSs) [49] in the literature, which can be used in combination with IRS. IPS can prevent detected alerts before occurring, whereas IRS responds to detected anomalies by using reactive response. The synergy between IPS and IRS can be exploited to provide better security.

## References

1. Intrusion campaigns, <https://stixproject.github.io/data-model/1.2/campaign/CampaignType/> (2022).
2. Challenges in cyber security, <https://www.rasmussen.edu/degrees/technology/blog/cyber-security-problems/> (2022).
3. R. A. Bridges, T. R. Glass-Vanderlan, M. D. Iannacone, M. S. Vincent, Q. G. Chen, *A survey of intrusion detection systems leveraging host data*, *ACM Comput. Surv.* 52 (6). <https://doi.org/10.1145/3344382>
4. What is threat intelligence? <https://www.recordedfuture.com/threat-intelligence/> (2022).
5. Why threat intelligence, <https://www.threatintelligence.com/blog/threat-intelligence> (2022).
6. V. Mavroeidis, R. Hohimer, T. Casey, A. Jesang, Threat actor type inference and characterization within cyber threat intelligence, in: 2021 13th International Conference on Cyber Conflict (CyCon), 2021, pp. 327–352. <https://doi.org/10.23919/CyCon51939.2021.9468305>.
7. S.-Y. Huang, T. Ban, Monitoring social media for vulnerability-threat prediction and topic analysis, in: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2020, pp. 1771–1776. <https://doi.org/10.1109/TrustCom50675.2020.00243>.
8. Mitre att&ck, <https://attack.mitre.org/> (2022).
9. V. Legoy, M. Caselli, C. Seifert, A. Peter, *Automated retrieval of att&ck tactics and techniques for cyber threat reports*, CoRR abs/2004.14322. [arXiv:2004.14322](https://arxiv.org/abs/2004.14322). <https://arxiv.org/abs/2004.14322>
10. M. Al-Ramahi, I. Alsmadi, J. Davenport, Exploring hackers assets: topics of interest as indicators of compromise, in: Proceedings of the 7th Symposium on Hot Topics in the Science of Security, 2020, pp. 1–4.
11. W. Zeng, Z. Liu, Y. Yang, G. Yang, Q. Luo, QBC inconsistency-based threat intelligence IoC recognition, *IEEE Access* 9 (2021) 153102–153107. <https://doi.org/10.1109/ACCESS.2021.3128070>.
12. S. Chandel, M. Yan, S. Chen, H. Jiang, T.-Y. Ni, Threat intelligence sharing community: A countermeasure against advanced persistent threat, in: 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), 2019, pp. 353–359. <https://doi.org/10.1109/MIPR.2019.00070>.
13. Stix, <https://oasis-open.github.io/cti-documentation/stix/intro> (2022).
14. Apt, <https://www.imperva.com/learn/application-security/apt-advanced-persistent-threat/> (2022).
15. A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, Survey of intrusion detection systems: techniques, datasets and challenges, *Cybersecurity* 2. <https://doi.org/10.1186/s42400-019-0038-7>.
16. A. H. Almutairi, N. T. Abdelmajeed, Innovative signature based intrusion detection system: Parallel processing and minimized database, in: 2017 International Conference on the Frontiers and Advances in Data Science (FADS), 2017, pp. 114–119. <https://doi.org/10.1109/FADS.2017.8253208>.
17. V. Jyothsna, K. M. Prasad, *Anomaly-based intrusion detection system*, in: J. Sen (Ed.), *Computer and Network Security*, IntechOpen, Rijeka, 2020, Ch. 3, pp. 1–16. <https://doi.org/10.5772/intechopen.82287>.
18. B. R. Raghunath, S. N. Mahadeo, Network intrusion detection system (NIDS), in: 2008 First International Conference on Emerging Trends in Engineering and Technology, 2008, pp. 1272–1277. <https://doi.org/10.1109/ICETET.2008.252>.
19. Y.-j. Ou, Y. Lin, Y. Zhang, Y.-j. Ou, The design and implementation of host-based intrusion detection system, in: 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, 2010, pp. 595–598. <https://doi.org/10.1109/IITSI.2010.127>.
20. K.-M. Yu, M.-F. Wu, W.-T. Wong, Protocol-based classification for intrusion detection, in: Proceedings of the 7th WSEAS International Conference on Applied Computer and Applied

- Computational Science, ACACOS'08, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 2008, p. 29–34.
21. A. Kumar, H. C. Maurya, R. Misra, A research paper on hybrid intrusion detection system, *International Journal of Engineering and Advanced Technology (IJEAT)* Vol 2.
  22. J. McHugh, Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln laboratory, *ACM Trans. Inf. Syst. Secur.* 3 (4) (2000) 262–294. <https://doi.org/10.1145/382912.382923>.
  23. J. Hu, X. Yu, D. Qiu, H.-H. Chen, A simple and efficient hidden Markov model scheme for host-based anomaly intrusion detection, *IEEE Network* 23 (1) (2009) 42–47. <https://doi.org/10.1109/MNET.2009.4804323>.
  24. A. Adebawale, S. Idowu, A. Amarachi, Comparative study of selected data mining algorithms used for intrusion detection, *International Journal of Soft Computing and Engineering (IJSCE)* 3 (3) (2013) 237–241.
  25. S. Thaseen, C. A. Kumar, An analysis of supervised tree based classifiers for intrusion detection system, in: 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering, 2013, pp. 294–299. <https://doi.org/10.1109/ICPRIME.2013.6496489>.
  26. N. Koroniotis, N. Moustafa, E. Sitnikova, B. P. Turnbull, **Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset**, CoRR abs/1811.00701. [arXiv:1811.00701](https://arxiv.org/abs/1811.00701). <http://arxiv.org/abs/1811.00701>
  27. C. A. Carver, U. W. Pooch, An intrusion response taxonomy and its role in automatic intrusion response, in: Proceedings of the 2000 IEEE Workshop on Information Assurance and Security, IEEE Computer Society Press West Point, NY, USA, 2000, pp. 129–135.
  28. H. S. Jomaa, J. Grabocka, L. Schmidt-Thieme, Hyp-rl: Hyperparameter optimization by reinforcement learning, *arXiv preprint arXiv:1906.11527*.
  29. D. J. Ragsdale, C. Carver, J. W. Humphries, U. W. Pooch, Adaptation techniques for intrusion detection and intrusion response systems, in: SMC 2000 conference proceedings. 2000 IEEE international conference on systems, man and cybernetics. 'cybernetics evolving to systems, humans, organizations, and their complex interactions' (cat. no. 0, Vol. 4, IEEE, 2000, pp. 2344–2349.
  30. F. Ullah, M. A. Babar, On the scalability of big data cyber security analytics systems, *Journal of Network and Computer Applications* 198 (2022) 103294.
  31. N. B. Anuar, S. Furnell, M. Papadaki, N. Clarke, Response mechanisms for intrusion response systems (IRSS), University of Plymouth: Plymouth, UK.
  32. C.-J. Chung, P. Khatkar, T. Xing, J. Lee, D. Huang, NICE: Network intrusion detection and countermeasure selection in virtual network systems, *IEEE transactions on dependable and secure computing* 10 (4) (2013) 198–211.
  33. S. Bandyopadhyay, S. Saha, Some single- and multiobjective optimization techniques, in: *Unsupervised classification*, Springer, 2013, p. 17–58.
  34. A. Nadeem, M. P. Howarth, An intrusion detection & adaptive response mechanism for MANETs, *Ad Hoc Networks* 13 (2014) 368–380.
  35. A. Nadeem, M. Howarth, Protection of MANETs from a range of attacks using an intrusion detection and prevention system, *Telecommunication Systems* 52 (4) (2013) 2047–2058.
  36. Y. Ping, Z. Futai, J. Xinghao, L. Jianhua, Multi-agent cooperative intrusion response in mobile ad hoc networks, *Journal of Systems Engineering and Electronics* 18 (4) (2007) 785–794.
  37. A. Shameli-Sendi, H. Louafi, W. He, M. Cheriet, Dynamic optimal countermeasure selection for intrusion response system, *IEEE Transactions on Dependable and Secure Computing* 15 (5) (2016) 755–770.
  38. B. Kordy, P. Kordy, S. Mauw, P. Schweitzer, ADTool: security analysis with attack–defense trees, in: *International conference on quantitative evaluation of systems*, Springer, 2013, pp. 173–176.
  39. N. Kheir, N. Cuppens-Boulahia, F. Cuppens, H. Debar, A service dependency model for cost-sensitive intrusion response, in: *European Symposium on Research in Computer Security*, Springer, 2010, pp. 626–642.

40. C. Hwang, K. Yoon, Methods for multiple attribute decision making. in multiple attribute decision making 1981 (pp. 58–191) (1981).
41. H. A. Kholidy, A. Erradi, S. Abdelwahed, F. Baiardi, A risk mitigation approach for autonomous cloud intrusion response system, *Computing* 98 (11) (2016) 1111–1135.
42. D. Perez-Botero, J. Szefer, R. B. Lee, Characterizing hypervisor vulnerabilities in cloud computing servers, in: *Proceedings of the 2013 international workshop on Security in cloud computing*, 2013, pp. 3–10.
43. S. Iannucci, V. Cardellini, O. D. Barba, I. Banicescu, A hybrid model-free approach for the near-optimal intrusion response control of non-stationary systems, *Future Generation Computer Systems* 109 (2020) 111–124.
44. V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, *nature* 518 (7540) (2015) 529–533.
45. J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, G. Zhang, Transfer learning using computational intelligence: A survey, *Knowledge-Based Systems* 80 (2015) 14–23.
46. M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*, John Wiley & Sons, 2014.
47. J. Zhu, K. Zou, X. Liu, K. Gao, Establishment of response evaluation model and empirical study of risk in enterprise threat intelligence, in: *2020 2nd International Conference on Economic Management and Model Engineering (ICEMME)*, IEEE, 2020, pp. 735–738.
48. F. Li, F. Xiong, C. Li, L. Yin, G. Shi, B. Tian, SRAM: A state-aware risk assessment model for intrusion response, in: *2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*, 2017, pp. 232–237. <https://doi.org/10.1109/DSC.2017.9>.
49. X. Zhang, C. Li, W. Zheng, Intrusion prevention system design, in: *The Fourth International Conference on Computer and Information Technology*, 2004. CIT '04., 2004, pp. 386–390. <https://doi.org/10.1109/CIT.2004.1357226>.



# Generalizing a Secure Framework for Domain Transfer Network for Face Anti-spoofing



B R Chandavarkar, Ayushman Rana, Mihir M. Ketkar, and Priyanka G. Pai

## 1 Introduction

With smartphone usage increasing worldwide, preserving privacy over matters relating to one's phone takes top priority. In recent years, a wide range of security tools have emerged to combine the principles of easy access to users and secure encrypted data, which is difficult for a malicious user to access. One of the newer technologies is face recognition, which at face value may seem like the perfect security tool; the face of a user is, for the most part, functionally unique and will not change for extended periods. This quality makes facial recognition an essential tool for looking into the future of information security. Scanning an associate's face alone will ensure their identity, which might mean that a person cannot have multiple driver's licences or fraudulent IDs or might be known among enforcement information. Face recognition is critical for future intelligent vehicle applications, like deciding whether or not someone is allowed or licenced to control a vehicle. The challenge for several security technology corporations these days is to create a face recognition security application that is quick, accurate, and ready to notice and verify a driver's identity while not constraining a car's driving surroundings. However, this skips away some flaws with the technology, i.e., the ability to "spoof" a person's face by using attacks like a replay attack or a print attack. As a designer of security tools, one must want only the user's face to unlock their phone, but what

---

B. R. Chandavarkar · A. Rana (✉) · M. M. Ketkar · P. G. Pai  
Department of Computer Science and Engineering, National Institute of Technology Karnataka,  
Mangalore, India

would happen if an attacker could bring in a high-definition image of the user's face on their iPad? Hackers would exploit certain flaws to steal valuable information from the users.

The biggest problem with face recognition technology is identifying and classifying a “real” face from a “spoofed” face. While there are many ways to go around spoofing faces, certain features also distinguish them from the real ones. The most common spoofing attacks are print attacks, replay attacks, and 3D masks [2–4]. In a print attack, the unauthorized person tries to get by as the authorized person by presenting a 2D image of the authorized person, either by printing it out on paper or by having a copy on their phone and holding it up to the camera. Such attacks are possibly the easiest to prevent as it shows up clearly during the feature extraction steps. A replay attack is when the unauthorized person replays a video of the authorized person to fool the system and get past its security. These attacks are more challenging to catch as the moving video makes the spoofed face appear livelier and more natural. The most challenging attack to prevent is likely the 3D mask attacks. In these attacks, the impersonator shows up with a three-dimensional facial mask of the authorised person [3].

With the growing requirement of biometric authentication, face detection has become the front runner due to its ease of use. However, as replication of faces is possible due to technological advancement, as discussed in the previous paragraph, we need methods to verify if the face detected is authentic or not. This chapter discusses several such anti-spoofing methods based primarily on convolutional neural networks.

The chapter proposes improvements to the existing anti-spoofing methods by introducing image re-colourization and security via image encryption. It explores image re-colourization based on Zhang et al.'s image colourization technique [5], later passing it through a spoof detector like GAN [1]. The image would then be stored securely after being encrypted.

The organization of the chapter is in the following manner. Section 1 presents the current methods used to tackle anti-spoofing primarily based on CNN. Section 3 describes the shortcomings faced by these methods. Section 4 introduces and discusses our suggestions to improve anti-spoofing methods and their security. Section 5 compares the method proposed with current methods. Section 6 of the chapter concludes it and discusses future works.

## 2 Related Works

Various works related to the framework presented by the chapter are listed down below.

### 2.1 Anti-spoofing Methods

#### 2.1.1 From RGB to Depth: Domain Transfer Network for Face Anti-spoofing [1]

CNN methods have provided excellent pattern recognition results, including face anti-spoofing detection, and CNN is generally used as a binary classifier that classifies live and spoofing faces. Nevertheless, there are issues with training deep neural networks for manual binary supervision and hence the absence of vigorous spoofing patterns. Therefore, there is a need for more auxiliary information that aids discrimination instead of solely relying on binary supervision. Generative adversarial networks (GANs) would provide us with the auxiliary information to generate superior quality images via adversarial training. GAN would take the RGB image and convert it to a depth image, providing supplementary information for the future classifier. This method consists of two networks that are domain transfer and classification. The image converts to depth images with domain transfer, and the GAN model itself has a generator and a discriminator. The generator creates the depth image, and the Discriminator trains the model to generate depth images. After creating the depth image, the classifier converts the feature map to Boolean values for binary classification. Compared to the RGB method, this model can learn to classify supervised features instead of memorizing them (Fig. 1).

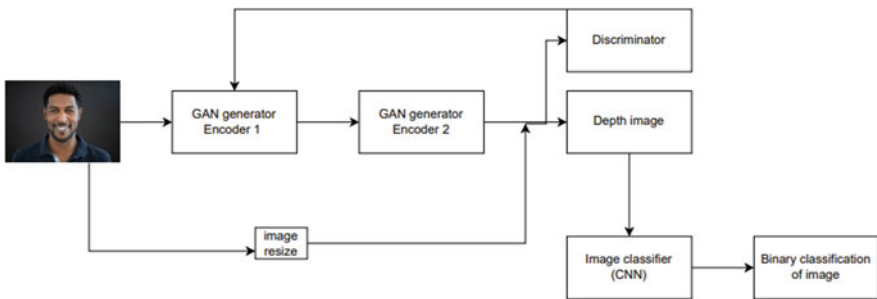


Fig. 1 Simple diagram illustrating domain transfer network

### **2.1.2 Detection of Spoofing Medium Contours for Face Anti-spoofing [6]**

As an extension to the CNN model, this method uses the number of regions of interest (RoIs) by selective search, and therefore it is called Region-based CNN (R-CNN). CNN independently classifies each RoI, and for predicting the object's mask, Mask R-CNN is used. This method proposes Contour Enhanced Mask R-CNN (CEM-RCNN) and is called so because this incorporates Mask R-CNN with contour objectness measure. After entering the image, the CNN model transforms the image to its feature map; the region proposal network (RPN) extracts the region of interest based on the CNN result. The following result is cropped to a fixed figure based on the region of interest using the RoIAlign layer. Mask R-CNN provides mask prediction on each trimmed feature patch. There are two different layers in classification: the box classification layer (cls) and the box regression layer (reg). The CI layer predicts whether the local block belongs to the background or foreground and detects spoofing medium contours (SMCs) by extracting two types of RoIs containing SMCs and one with a general object. If one of them is classified as a spoofing medium, the image is detected to be spoofed.

### **2.1.3 Face Anti-spoofing via Stereo Matching [7]**

Depth information extracted from spoofed data can be affected by the quality of the image or even the posture. Inaccurate depth information can lead to spoofed images appearing real. Stereo matching comes into the picture to prevent this. It accurately calculates the depth with just two rectified images that line up horizontally. The two images first undergo feature extraction and depth analysis. The results then go through regularization with the help of an encoder, spatial pyramid pooling, and decoder. The output then undergoes disparity regression. The disparity map generated is independent of the quality of the input image. Thus, this method is pretty robust in the detection of spoofed faces.

### **2.1.4 Attention-Based Two-Stream Convolutional Networks for Face Spoofing Detection [8]**

This method simultaneously analyses the image in two spaces: RGB space (original imaging space) and multi-scale retinex (MSR) space (illumination-invariant space). The method strives to take advantage of the complementary nature of the two spaces. While the RGB space detects facial features with great attention to detail, it is sensitive to illumination. On the other hand, the MSR space is unaffected by illumination but captures fewer details. The analyses give us two discriminative features to consider while differentiating real and spoofed faces. The results of these analyses are combined, and the decision is made based on this combined result. The results are combined carefully, using a deep learning-based fusion method. This

method is effective as it makes decisions based on a fusion of two discriminative features and not just one.

### **2.1.5 Attention-Based Two-Stream Convolutional Networks for Face Spoofing Detection [9]**

This method deals with the usage of unsupervised adaptation of domain for the purposes of anti-spoofing. Face detection methods have been limited to a loosely based on four kinds of detection methods, namely: motion based, distortion based, texture based, and convolutional neural network (CNN) based. One common assumption in computer vision and machine learning is that the training data and testing data are sampled from the same distribution. However, many practical application scenarios (e.g., face verification and spoofing detection) involve information originating from different distributions (facial appearance and pose, illumination conditions, camera devices, etc.). Hence, an overfitting issue may arise. This might cause a great degree of inaccuracies in the model. Domain adaptation is associated with transfer learning which aims at solving the learning problem in the target domain under a certain distribution.

## **2.2 Image Re-colourization**

Image re-colourization has made significant advancements with deep learning methods in recent years. For example, Zhang et al. [5] proposed convolutional neural networks to “hallucinate” an input black and white image to look like it would if the image was coloured. ImageNet dataset [10] trains the network by converting all the RGB images to the colour space. This method is generally used to modernize old pictures. Still, in anti-spoofing, the Image re-colourizer will use it to make the process more robust by correcting washed-out images or re-colouring black and white photos as RGB to Domain Map method heavily relies on the colour information of the picture.

## **2.3 Image Encryption**

As privacy is becoming more and more critical in today’s world, the framework must ensure the secrecy of the image to prevent it from being misused. This chapter proposes introducing an image encryptor to encrypt the final image after it goes through anti-spoofing detection. The RGB displacement method [11] is used for the same as it is light and straightforward with good encryption performance and low computational cost. It XORs the final image with a key image bit by bit.

### 3 Shortcomings of Existing Solutions

While the methods mentioned above are effective and popularly used, they are not free of fault and have room for improvement. The RGB to depth domain transfer network has a significant shortcoming in that it needs only coloured images. Should a black and white image be presented, it will not give a detailed and correct depth image as it heavily relies on the colours present in the pixels. The method of detection of SMCs has a shortcoming as well. It fails to detect a 3D face mask as a spoofed face. Anti-spoofing via stereo matching has drawbacks related to the CNN and a room for improvement. Attention-based two-stream convolutional networks, seemingly promising in the aspect of anti-spoofing, have an overfitting problem. This problem makes it unable to generalize well under unseen samples. Although advanced features were introduced in the current model, how to design and learn superior features that can better fit the scheme should be researched further. This domain adaptation scheme mandates sufficient information to transfer the data from the source to the target, and in the upcoming years domain adaptation with zero-shot learning will be studied. These drawbacks exist because of poor implementation of available data and limited research conclusions to only a few methods. They could be called gaps in information, while other gaps exist due to contradictory results across many papers that could be called gaps in reach.

### 4 The Proposed Framework

The proposed framework includes making the anti-spoofing method much more robust by introducing image re-colourization and security.

#### 4.1 Image Re-colourization [5]

There are images in which the pictures either lack enough colour saturation or are just black and white images (CCTV camera recording at night causing the lack of colours). The following method requires fully RGB images to create a proper depth map to detect spoofed images. So before the images are processed in the GAN generator, they are pre-processed in a re-colourizing algorithm to give the images the necessary details, which we propose.

Pre-processing will require OpenCV and is based on Zhang et al.'s "Colorful Image colourization technique." The method model is trained with the ImageNet dataset, and colour information is encoded based on lightness intensity, green–red and blue–yellow.

Intensity is denoted as  $L$ , whereas the colours are denoted as  $ab$ . The below algorithm presents the implementation of re-colourization (Fig. 2).

- 1:  $image \leftarrow original\_image$
- 2:  $L \leftarrow cv2.split(image)$

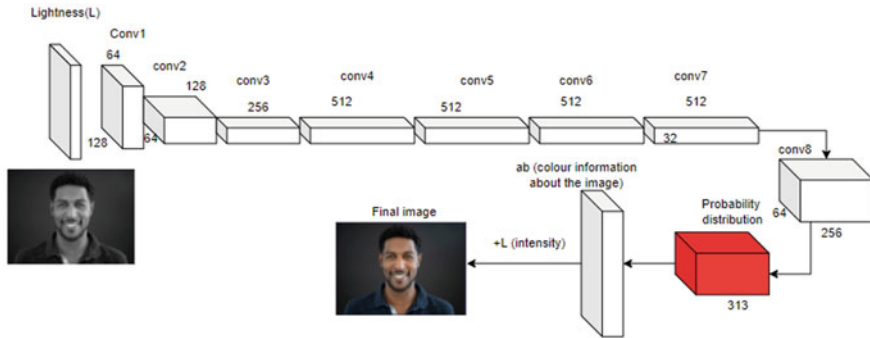


Fig. 2 Depiction of black and white colourization with deep learning

```

3: network ← cv2.BlobFromImage(L)
4: ab ← network.forward() {} will predict ab channel values
5: colorized ← np.concatenate(L, ab)
6: colorized ← cv2.cvtColor(colorized) {Color space will be converted to RGB}
7: output ← colorized
    
```

## 4.2 Anti-spoofing Detection [1]

The proposal involves the use of two deep networks. The first one will develop a domain transfer network to model RGB faces into depth maps. An adversarial learning paradigm is used to obtain high-quality depth maps with the generator and discriminator. Second, the framework at hand will use an in-built classifier to determine whether an input face image is real or an attempt at spoofing. Training of this network to a usable tool is done by using latent variables learnt in the previous (domain transfer) stage as the input of the classifier.

### 4.2.1 Domain Transfer

The first deep network (domain transfer) has been designed by using a GAN model for converting RGB to depth images. This GAN model has two primary functionalities, that of a generator and a discriminator. The generator consists of two parts, the first is a collection of convolutional layers and is primed for feature extraction. The second part is a map of the responses that is concatenated by adapting average pooling from the previous convolution blocks.

- 1: for K training iterations do
- 2: Sample a mini-batch with m image pairs (x(1), d(1)), ... (x(m), d(m));

- 3: Inference Enet1 and Enet2 to obtain  $z$ ,  $z$  and  $d$  as described in Eq. (1) and Eq. (2);
- 4: Inference Cls with  $z$  to calculate the output  $y$  as described in Eq. (6);
- 5: Update the discriminator parameters using the Adam optimiser;
- 6: Update the generator parameters using the Adam optimiser;
- 7: Update the classifier parameters using the Adam optimiser.
- 8: end for

The classifier aims to perform a final distinction between the input images (whether or not they are live or spoofing faces). The last layer of the aforementioned convolution layers is a fully connected layer that converts the feature map into 0 or 1. As mentioned earlier, this input originates from one part of the DTN branch (for classification). To be more specific, it maps to features presented in the first part of the Domain Transfer Network. The proposed tool can now be used for learning shared features. Optimization of the classifier is carried out by cross-entropy loss.

So we can have the equation:

$$L_c(Cls) = \frac{1}{N} \sum_i -[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

where  $y$  is the label, and  $y$  is the predictor.

As aforementioned, the input comes from one part of the proposed DTN branch. Specifically, it uses the feature maps generated by Enet1. Since the feature maps are generated during the domain transfer process, they can be regarded as features of the depth domain. As discussed earlier, compared with the information extracted from the RGB domain, the proposed method can gradually learn the shared features supervised by the depth domain transfer network. We optimize the classifier by minimizing the cross-entropy loss.

### 4.3 Image Encryption [11]

After the image goes through anti-spoofing detection, we need to make sure that the image is stored safely and protect the person's identity. It is done so by the RGB displacement method.

The processed image is XOR'ed by a randomly generated image as the key bit by bit (Fig. 3).

Key generation happens with the generation of a random image which gets split up to its RGB components. The process image is encrypted accordingly, and hence the framework for secure anti-spoofing detection is complete.



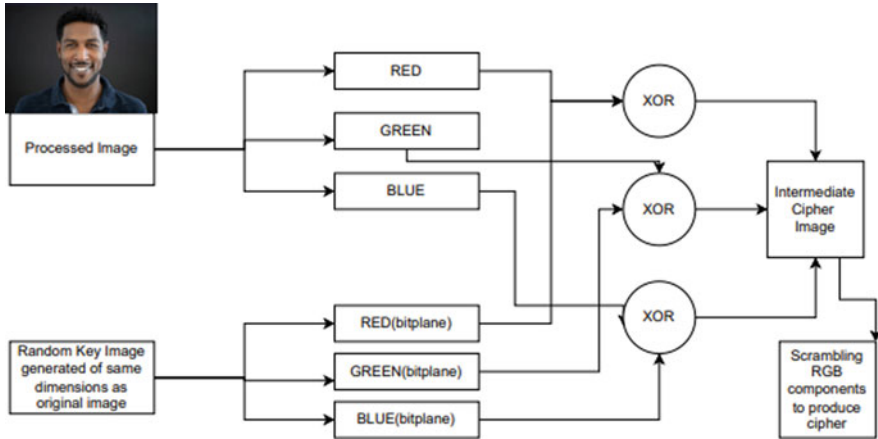


Fig. 3 Process of encrypting the image

### 5 Comparisons with Other Methods

This comparison will be between a generic anti-spoofing detection method and the method proposed by this chapter. The factors will be versatility, accuracy, time consumption, and security.

Comparison table		
Factors	Current methods	Method proposed
Versatility	Low	High
Accuracy	Medium	High
Time consumption	Low	Medium
Security	Low	High

On the basis of the comparison, it is evident that the proposed method solves a lot of the issues of the current method and eliminates major drawbacks.

### 6 Conclusions and Future Work

Currently, the methods of face anti-spoofing are advanced and robust but are not without faults. Understanding the working of these methods is crucial as they might contain minor yet glaring issues that stray away from the accurate processing of what is required. As we move into an exceedingly digital world, it is of utmost importance that we step up anti-spoofing technology to improve and cover the gaps in contemporary research.

We propose one of the improvements to re-colourize images using Zhang et al.'s "Colorful Image colourisation technique." This would ensure that the image is sufficiently saturated and contains enough information for easier feature extraction. Then, the image is passed through an anti-spoof detector. We also propose encrypting the images we store to ensure the person's identity in the image.

We will implement our ideas and test them out in real-world scenarios in our future works.

## References

1. Y. Wang, X. Song, T. Xu, Z. Feng, X.-J. Wu, From RGB to Depth: Domain Transfer Network for Face Anti-Spoofing, <https://ieeexplore.ieee.org/document/9507460>, [Accessed: 07-02-2022] (2021).
2. S. Zhang, X. Wang, A. Liu, C. Zhao, J. Wan, S. Escalera, H. Shi, Z. Wang, S. Z. Li, A Dataset and Benchmark for Large-scale Multi-modal Face Anti-spoofing, <https://arxiv.org/pdf/1812.00408v3.pdf>, [Accessed: 28-02-2022] (2019).
3. Z. Yu, Y. Qin, X. Li, C. Zhao, Z. Lei, G. Zhao, Deep Learning for Face Anti-Spoofing: A Survey, <https://arxiv.org/pdf/2106.14948v1.pdf>, [Accessed: 28-02-2022] (2021).
4. H. Feng, Z. Hong, H. Yue, Y. Chen, K. Wang, J. Han, J. Liu, E. Ding, Learning Generalized Spoof Cues for Face Anti-spoofing, <https://arxiv.org/pdf/2005.03922v1.pdf>, [Accessed: 28-02-2022] (2020).
5. R. Zhang, P. Isola, A. A. Efros, Colorful Image Colorization, <http://richzhang.github.io/colorization/>, [Accessed: 28-02-2022] (2016).
6. X. Zhu, S. Li, X. Zhang, H. Li, A. C. Kot, Detection of spoofing medium contours for face anti-spoofing, <https://ieeexplore.ieee.org/document/8884098>, [Accessed: 06-02-2022] (2019).
7. Z. Li, J. Yuan, B. Jia, L. X. Yifan He, An Effective Face Anti-Spoofing Method via Stereo Matching, <https://ieeexplore.ieee.org/document/9403897>, (2021).
8. H. Chen, G. Hu, Z. Lei, Y. Chen, N. M. Robertson, S. Z. Li, Attention-Based Two-Stream Convolutional Networks for Face Spoofing Detection, <https://ieeexplore.ieee.org/document/8737949>, [Accessed: 05-06-2021] (2019).
9. H. Li, W. Li, H. Cao, S. Wang, F. Huang, A. C. Kot, Unsupervised Domain Adaptation for Face Anti-Spoofing, <https://ieeexplore.ieee.org/document/8279564> (2018).
10. S. labs, ImageNet dataset, <https://www.image-net.org/>, [Accessed: 28-02-2022] (2016).
11. G. Zhu, W. Wang, X. Zhang, M. Wang, Digital image encryption algorithm based on pixels, <https://ieeexplore.ieee.org/document/5658790>, [Accessed: 05-02-2022] (2010).

# Survey on Game Theory-Based Security Framework for IoT



Pranav Joshi, Suresh Kamediya, Ritik Kumar, and B R Chandavarkar

## 1 Introduction

The Internet of Things (IoT) has emerged as a breakthrough in all internet-related technologies, particularly in intelligent appliances. It has the potential to improve the efficiency of physical object management and control through smart sensing and intelligent decision-making. The IoT market is estimated to grow to 75 billion by 2025. As a result, the Internet of Things will have a significant impact on our lives [1, 2]. IoT devices can be accessed from anywhere over a trusted network, but they are subject to a variety of cyber-attacks, compromising their security and reliability. Because IoT is used in so many different technologies, from nuclear weapons to self-driving cars to smart home appliances, the security and reliability of IoT are crucial. Attackers have the ability to take control of and harm essential infrastructure, and new attacks are discovered on a regular basis, with a trend towards becoming more intelligent and sophisticated. So it becomes a challenging work for the more secure and reliable framework [3].

This chapter includes a brief about different types of security threats, Requirements, and constraints, along with the need for Game Theory Models in IoT and a survey of Security frameworks for IoT based on Game Theory Models. The threats discussed can be classified into four categories: internal, external, clear and passive based on if the malicious nodes are part of IoT system and based on if the attacker is clearly attacking the nodes or just capturing the information without altering the nodes [4]. Some security requirements are also discussed, such as data confidentiality, integrity, authentication etc., which are necessary so that data do not get compromised between the nodes and outside of IoT system [5].

---

P. Joshi (✉) · S. Kamediya · R. Kumar · B. R. Chandavarkar  
National Institute of Technology Karnataka Surathkal, Computer Science & Engineering,  
Mangalore, Karnataka, India

A brief overview about game theory is also provided. Game theory comprises a strategic interaction between two groups where each group tries to optimise their output. It can be useful to develop different optimisation frameworks. Since in the most of the optimisation frameworks constraints and objectives are defined explicitly, consideration of game theory helps in building a framework with flexible constraints and objectives based on the opponent's move which may in turn into optimal output [6, 7].

The attackers are always rational beings who will do everything in their power to cause as much damage to the networks as possible. Therefore, the encounter entails an interaction between the defenders and attackers, which can be directly converted to a game among participants in which each participant seeks to advance his or her own interests [8]. The maximum payoffs for one side can only be obtained when the actions of the opposing side are taken into consideration. Inspired by this fact, concepts of game theory used in security frameworks are discussed in this chapter to achieve an enhanced layer of security [9].

In this chapter, a survey of game theory-based security frameworks for IoT is done. Because the actions of both attackers and defenders are reliant on each other's counteraction, game theory is the best fit for this security framework. The goal of this chapter is to offer a comparative study of game theory-based security framework for achieving the highest security and reliability across all systems in IoT.

In Sect. 2, some threats and security requirements have been discussed along with the constraints of the hardware used in IoT with respect to security frameworks. Section 3 discusses how game theory can be used in IoT Systems to enhance security and what are its uses and applications in the field of IoT. Section 4 discusses a survey of proposed game theory security frameworks targeting different spectrum of IoT in the area of security issues, followed by the conclusion of the work in Sect. 4.

## **2 Threats and Security Requirements in IoT**

IoT devices collect an enormous amount of data over the Internet which makes it vulnerable to cyber-attacks. It is assumed that the hacker knows about the security mechanisms of sensor nodes and can damage or capture the sensor node [10]. Hackers often exploit the weaknesses of IoT devices to capture sensible information for their own benefits [11, 12]. This section covers constraints of hardware, security requirements and possible threats in IoT.

### **2.1 Constraints of Hardware in IoT**

It is very hard to design security services for sensor nodes as there are many limitations on the hardware of sensor nodes [13]:

- **Energy:** In WSNs, communication is very expensive compared to computation because each bit transfer uses the same amount of power as performing 1000s of instructions. Furthermore, higher security levels in WSNs generally correspond to increased cryptographic function energy consumption [14].
- **Memory:** A sensor node's memory usually consists of flash memory and RAM. Application code is downloaded and stored in flash memory, while application programmes, sensor data and intermediate computations are saved in RAM. There is frequently insufficient space to run sophisticated algorithms after loading the OS and application code. As a result, the majority of current security techniques are ineffective.
- **Computation:** Sensor nodes' integrated CPUs are often less powerful than those in wired or ad hoc network nodes. As a result, WSNs cannot utilise complicated cryptographic algorithms.

## 2.2 *Evaluation of Security Schemes*

To determine whether a security method is appropriate for WSNs, it is recommended to utilise the following metrics [5]:

- **Security:** Security services in WSNs are designed to protect data and resources from assaults and misbehaviour. WSNs must meet the following security requirements [15]:
  - **Availability:** It ensures that network services remain available even when denial-of-service attacks are present.
  - **Authentication:** It ensures that communication between nodes is legitimate. For example, a malicious node cannot impersonate a trusted network node.
  - **Authorisation:** It ensures that only approved sensors are able to provide data to network services.
  - **Confidentiality:** It ensures that a message may only be understood by the intended recipients.
  - **Freshness:** It indicates that the data are current, ensuring that no opponent can replay previous messages.
  - **Integrity:** It ensures that messages transferred from one node to another are not tampered with by hostile intermediate nodes.
  - **Non-repudiation:** It means that a node cannot deny sending a message it has already sent.
  - **Forward secrecy:** After leaving the network, a sensor should never be able to read any future communications.
  - **Backward secrecy:** A new sensor that has just joined should not be able to read any previously broadcast messages.

- **Energy efficiency:** In order to maximise node and network lifespan, a security method must be energy efficient.
- **Flexibility:** Key management is required to accommodate various network deployment strategies, such as random node scattering and predetermined node placement.
- **Fault-tolerance:** Despite problems such as failed nodes, a security scheme should continue to deliver security services.
- **Resilience:** Even if few nodes are hacked, a security scheme should be able to withstand the attacks.
- **Scalability:** A security strategy must be scalable without compromising the security needs.
- **Assurance:** It is the capacity to transmit various types of information to end-users at various levels. A security method should provide options for desired reliability, latency and other factors.
- **Self-healing:** Sensors can fail due to a lack of energy. To maintain a certain level of security, the remaining sensors may need to be reorganised.

### ***2.3 Threats in IoT***

IoT devices suffer from different cyber-attacks, which generally targets basic layers of IoT such as (Physical, Data Link, Network, Transport and Application Layer). These attacks can be further classified as internal, external, clear and passive attacks [4, 16, 17]

1. **Internal and external attacks:** Internal attackers have access to internal protocols and so it can attempt various malicious activities such as intercepting and modifying messages between IoT devices. External attackers, on the other hand, may not have direct access to nodes, but they can still replace them with malicious node to disturb the normal performance of the network. For example, internal attacker could be a neighbour getting temporary access to home router, while an external attacker could be an ISP getting access to security gateway [1].
2. **Clear and passive attacks:** In clear attack, the attacker can manipulate traffic stream by transmitting, responding, modifying or deleting specific messages. Clear attacks cause some manipulations to WSNs and can prevent infected sensor nodes from communicating with others, so a great amount of information is needed for defence mechanism to mitigate these attacks. In passive attack, the attacker captures (eavesdrops) the transmitted traffic stream without modifying the network. For example, the attacker may target sensitive data such as encryption codes, the attacks where malicious nodes ignore packet forwarding to their neighbours to save their own power consumption [5].

### 3 Game Theory in IoT

Game theory is a very useful technique for intelligent optimisation. The game theory model comprises a game between two groups of players, which are the network devices on IoT in our case, which in turn choose to play cooperatively or non-cooperatively to increase their payoffs (favours to win the game). Here are some terminologies to introduce us with basic game theory model [5].

- **Game:** It can be defined as a situation like any normal game where two players as opponents try to make their move strategically to win the game.
- **Player:** A player is a decision-maker, who plays with a strategic point of view to increase one's payoff in the game.
- **Payoff:** A payoff can be mathematically treated as a value indicating the outcome of a strategy opted by the player playing the game. Higher the payoff means the strategy is more distinguished.
- **Strategy:** Strategy can be understood as a layout of all the actions you take or plan to take in the near future with respect to the game.
- **Nash Equilibrium:** Nash equilibrium (NE) can be treated as a condition when there is no choice for either of the players in the game to diverge from their chosen strategy from the beginning because that seems to be the most optimal one to them. This goes both ways, and hence both the players stick to their strategies. There can be more than one NE in a game [18].

IoT is a vast network that consists of a huge number of nodes in our physical world and serves various purposes in different fields. But it solely relies on the nodes and how the nodes communicate over the network. This communication can be facilitated and enhanced using game theory Models to improve various factors like maximising energy, detecting malicious activity, generating trustworthiness and many more. It proves to be a very useful tool in the field of IoT which can be used to deal with various problems and present optimised solutions to those problems. The only thing required to implement this is converting our physical world problem related to IoT in the form of respective standard game theory model [19].

There are many tools in game theory used, like payoffs, game trees, equilibrium, to denote the communication between two parties (nodes being these parties in case of IoT). The model analyses these nodes while communicating to keep a note of their behaviour and how that can be used as an indicator to formulate strategies related to the game theory model. It helps the players in a game to adjust and adapt to various strategies during the game. This leads to high versatility in studying the behaviour of communication in IoT [19].

To achieve maximum payoff in the form of minimising power consumption [19] and resource consumption and ensure a high quality while completing the assigned job, it requires communication to be light-weight. Also, in IoT, a huge amount of data is travelling through the network and data security for such a network should be at its top in case of any sensitive information is being sent or received. Therefore, we require a functionality to prevent such kinds of intrusions and maintain its security

consistently at all parameters. For all such kind of problems, game theory models can be used and applied in the field of IoT.

## 4 Game Theory-Based Security Models

### 4.1 Game Theory-Based Trust Model

Games in game theory can be classified into two types based on IoT-based security which are cooperative and non-cooperative games. Cooperative game comprises a collection of nodes (devices on IoT Network) working collectively to maximise the whole network security against various kinds of threats and attacks [20], whereas in non-cooperative game, each individual node tries to maximise its own payoff, which is used to model and deal with different security issues.

The trustworthiness mechanisms are of utmost importance for IoT network's security. The general flow of process for this trust model can be seen below (Fig. 1). This model is based on a previously discussed general trust model in [21]. In the first stage, it collects information from the traffic stream over the network, then comes the second stage, which uses a suitable trust model which analyses the data, and at the later stage, the IDS (Intrusion Detection System) checks the analysed data. The fourth and the final stage take charge, and it either punishes or rewards (like it happens in a game) depending upon whether the packets of data were infectious or noble, respectively. This method aims at achieving a network with efficient energy utilisation against the impact caused by the attacker, by using the basic principle of learning automata by monitoring the packets coming on the receiving end. The same principle is applied to improve IoT network's security using game theory [22].

This subsection focuses on a basic trust model based on game theory to deal with the threats and attacks pertaining on IoT networks which helps us to find the suspicious nodes (devices over network) and the ones that act selfishly. In this section, a security framework is discussed using a game theory model from [5].

There are 2 ways to proceed with a safeguarding strategy based on our game theory model-cooperative process and non-cooperative process. As we can see in Fig. 2, the cooperative process works as follows: the nodes that are the part of the cooperative game over the network begin with calculating 3 parameters, i.e. level of security, reputation and cooperation with respect to every other node that is a part of the process. Then, using these parameters, an information list is formed, and let us name it  $N(i)$ . Lastly, it is checked if each and every node is acting suspiciously or selfishly. That node is either punished or rewarded based on the findings of the previous step, which tells us if the node was malicious or not respectively. If the node is rewarded, it is also checked if there is NE (Nash equilibrium). For detecting the Nash equilibrium, a timer can be used which acts as a gateway, creating a threshold, which in turn can help in reducing the risk for such sensitive applications in real-time scenarios.



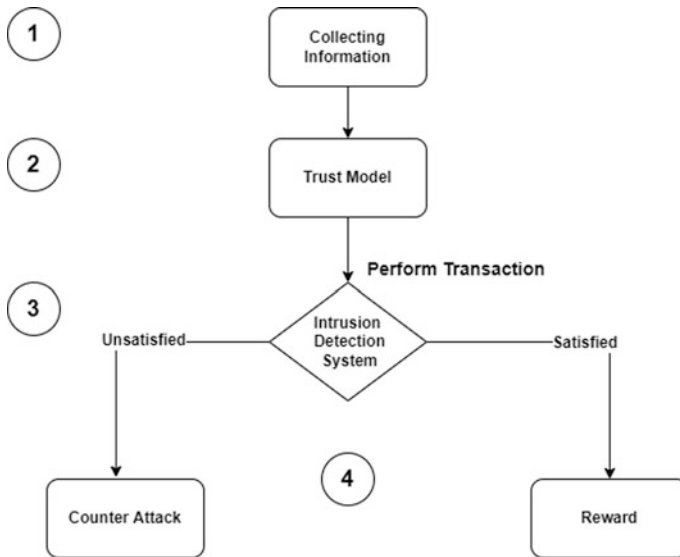


Fig. 1 General trust and reputation model mechanisms [5]

This strategy involves a non-cooperative process, so it begins with playing a non-cooperative game against the attacker. Then, the same series of steps are produced as in the cooperative process. But, instead here the algorithm starts from a different step. It is first checked whether the node is suspicious or noble, and also it is checked if it is selfish or not. Then, the next two steps are omitted after that, which includes calculating the parameters and collecting information to form the information list of neighbours  $N(i)$  as discussed in the cooperative process. Rest all the steps are the same.

Figure 1: Overview of General Trust Model: The model is based on a general trust model which is similar to a model presented in [23], which is divided into four phases. The first phase consists of collecting all the information that we get through analysis of the traffic stream over the network. After analysing all the information in the prior phase, the model chooses the respective game theory model for the data, and this marks the end of the second phase. After the second phase, all the analysed data are passed through an IDS (Intrusion Detection System), and based on the results of that, the fourth phase is responsible for either rewarding the packets or punishing them based on the study and results of the IDS. This approach can be used to improve security within networks like WSNs and IoT [5, 24].

#### 4.1.1 Analysis of General Trust-Based Model

The widely used game models based on cooperative games use the concept of centralised authentication. These models are used to keep track of noble nodes,

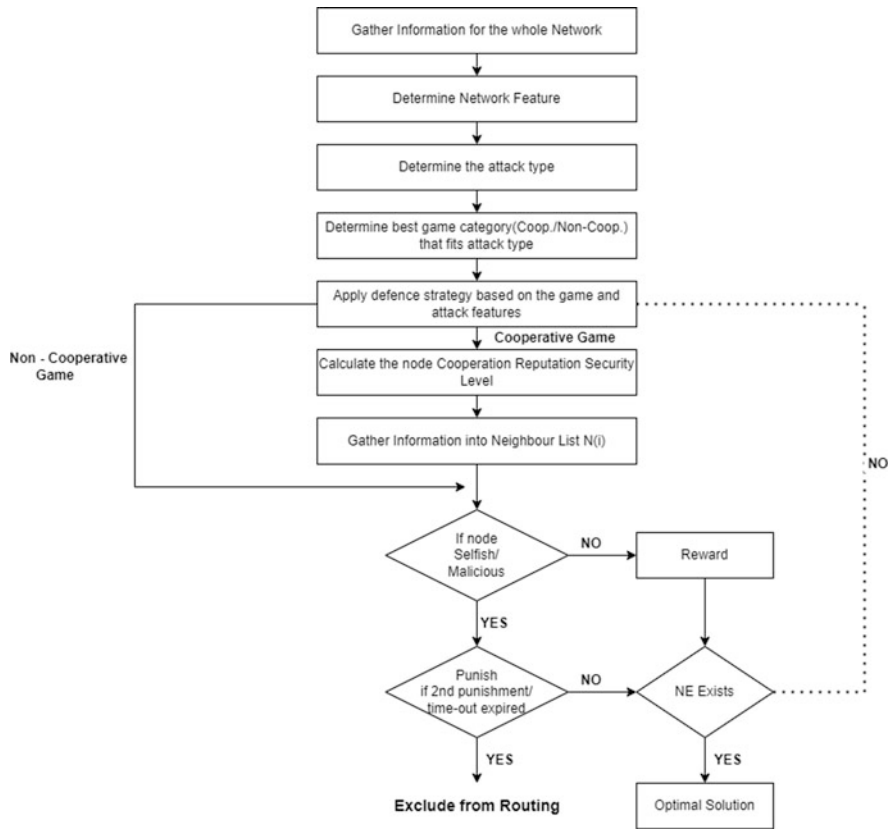


Fig. 2 Game theory trust model [5]

punishment of malevolent nodes, using cooperative ones. There are several benefits of centralised authentication, but there are also some drawbacks that can cause problems and compromise network security; for example, there can be a decay of resources for the central node for security. In research paper [25], one solution is introduced to tackle this problem, and it is called “group authentication,” which reduces such kinds of risks on the central node. It outsources the tasks related to security to other cooperative nodes.

Among the various game kinds, evolutionary games offer a valuable strategy in which participants logically change their behaviours in response to the game’s iterative development [26]. The evolutionary game is capable of dealing with population interactions among rational biological actors. Evolutionary games can also be divided into static and dynamic theories [27]. The evolutionary game employs evolutionarily stable strategies in the static hypothesis (ESS). The replicator dynamics have been used to model the adaptation of the players’ strategies in the dynamic hypothesis [27].The evolutionary game framework has a number

of advantages over the traditional non-cooperative game, including equilibrium selection, constrained rationality and dynamic player behaviour. As a result, we believe that evolutionary game models can be utilised to reduce sophisticated assaults in WSNs, resulting in more stable systems.

## 5 Conclusion

With the advancement in the field of Internet of Things, the topic of security has become of prime importance. In this chapter, different types of attacks against IoT are discussed along with the ways to protect them from the same. Many attacks are being launched against IoT and the ones that are discussed in this chapter are proven to be fatal and recurring over time [28]. Over this period of time, it made us to learn that to prevent any suspicious activity or any attack over an IoT network, it is important to protect its endpoints along with monitoring the traffic and securing transactions.

The security and reliability concerns for IoT are crucial as they can be accessed from anywhere through network. This chapter also discussed and analysed a new security framework based on game theory. To detect a threat, the basic trust model is considered. Based on the game theory two types of defence strategy cooperative and non-cooperative process are proposed. Every node is checked and then get rewarded or punished depending on that node is malicious or not [5].

## References

1. G. Sharma, S. Vidalis, N. Anand, C. Menon, S. Kumar, A survey on layer-wise security attacks in IoT: Attacks, countermeasures, and open-issues, <https://doi.org/10.3390/electronics10192365> (2021).
2. J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, W. Zhao, A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications, <https://doi.org/10.1109/JIOT.2017.2683200> (2017).
3. H. Wu, Z. Wang, Multi-source fusion-based security detection method for heterogeneous networks, <https://doi.org/10.1016/J.COSE.2018.01.003> (2018).
4. Y. Yu, K. Li, W. Zhou, P. Li, Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures, <https://doi.org/10.1016/j.jnca.2011.03.005> (2012).
5. M. S. Abdalzaher, K. Seddik, O. M. Maha ElSabrouy, H. Furukawa, A. Abdel-Rahman, Game theory meets wireless sensor networks security requirements and threats mitigation: a survey, <https://doi.org/10.3390/s16071003> (2016).
6. S. Lee, S. Kim, K. Choi, T. Shon, Game theory-based security vulnerability quantification for social Internet of Things, <https://doi.org/10.1016/j.future.2017.09.032> (2017).
7. F. Fang, S. Liu, A. Basak, Q. Zhu, C. D. Kiekintveld, C. A. Kamhoua, Introduction to game theory, <https://doi.org/10.1002/9781119723950.ch2> (2021).
8. X. Liang, Y. Xiao, Game theory for network security, <https://doi.org/10.1109/SURV.2012.0062612.00056> (2013).

9. H. Wu, W. Wang, A game theory based collaborative security detection method for Internet of Things systems, <https://doi.org/10.1109/TIFS.2018.2790382> (2018).
10. Y. B. Reddy, A game theory approach to detect malicious nodes in wireless sensor networks, <https://doi.org/10.1109/SENSORCOMM.2009.76> (2009).
11. S. Hameed, F. I. Khan, B. Hameed, Understanding Security Requirements and Challenges in Internet of Things (IoT): A Review, <https://doi.org/10.1155/2019/9629381> (2019).
12. S. Hameed, F. I. Khan, B. Hameed, Understanding Security Requirements and Challenges in Internet of Things (IoT): A Review, <https://doi.org/10.1155/2019/9629381> (2019).
13. Y. Wang, G. Attebury, B. Ramamurthy, A survey of security issues in wireless sensor networks, <https://doi.org/10.1109/COMST.2006.315852> (2006).
14. L. Yuan, G. Qu, Design space exploration for energy-efficient secure sensor network, <https://doi.org/10.1109/ASAP.2002.1030707> (2002).
15. S. Pal, M. Hitchens, T. Rabehaja, S. Mukhopadhyay, Security Requirements for the Internet of Things: A Systematic Approach, <https://doi.org/10.3390/s20205897> (2020).
16. M. R. Islam<sup>1</sup>, K. M. Aktheruzzaman, An analysis of cybersecurity attacks against Internet of Things and security solutions, <https://doi.org/10.4236/jcc.2020.84002> (2020).
17. A. Gouisseem, K. Abualsaud, E. Yaacoub, T. Khattab, M. Guizani, Game theory for anti-jamming strategy in multichannel slow fading IoT networks, <https://doi.org/10.1109/JIOT.2021.3066384> (2021).
18. C. A. Holt, A. E. Roth, The Nash equilibrium: A perspective, <https://doi.org/10.1073/pnas.0308738101> (2004).
19. C. Chi, Y. Wang, X. Tong, M. Siddula, Z. Cai, Game theory in internet of things: A survey, <https://doi.org/10.1109/JIOT.2021.3133669> (2021).
20. Z. Xu, A. Qu, K. An, Coalitional game based joint beamforming and power control for physical layer security enhancement in cognitive IoT networks, <https://doi.org/10.23919/JCC.2021.12.009> (2021).
21. F. G. Marmol, G. M. Perez, TRMSim-WSN, trust and reputation models simulator for wireless sensor networks, <https://doi.org/10.1109/ICC.2009.5199545> (2009).
22. M. Kodialam, T. Lakshman, Detecting network intrusions via sampling: a game theoretic approach, <https://doi.org/10.1109/INFCOM.2003.1209210> (2003).
23. F. G. Marmol, G. M. Perez, TRMSim-WSN, Trust and Reputation Models Simulator for Wireless Sensor Networks, <https://doi.org/10.1109/ICC.2009.5199545> (2009).
24. M. Kodialam, T. Lakshman, Detecting network intrusions via sampling: a game theoretic approach, <https://doi.org/10.1109/INFCOM.2003.1209210> (2003).
25. L. Harn, Group authentication, <https://doi.org/10.1109/TC.2012.251> (2012).
26. K. Komathy, P. Narayanasamy, Trust-based evolutionary game model assisting AODV routing against selfishness, <https://doi.org/10.1016/j.jnca.2008.02.008> (2008).
27. H. Z, Game theory in wireless and communication networks: Theory, models, and applications, [https://assets.cambridge.org/97805211/96963/frontmatter/9780521196963\\_frontmatter.pdf](https://assets.cambridge.org/97805211/96963/frontmatter/9780521196963_frontmatter.pdf) (2012).
28. A. Tabassum, W. Lebda, Security framework for IoT devices against cyber-attacks, <https://doi.org/10.5121/csit.2019.91321> (2019).

# Survey: Intrusion Detection for IoT



B R Chandavarkar, Joshitha Reddy D., Surla Lakshmi Poojitha, and Reshma Tresa Antony

## 1 Introduction

The Internet of Things (IoT) is a network of interconnected devices that can be remotely monitored and controlled [1]. Smart TVs, smart phones, and other gadgets are few examples. The primary goal of IoT is to make people's lives more comfortable and productive. On the other hand, these IoT devices are vulnerable to a variety of security threats. Denial of service (DoS), routing attacks, and man in the middle attacks are prominent among the others. These attacks have a negative impact on IoT services and smart environments. As a result, it is our primary concern to protect IoT systems, which is why IDS comes into the picture [2].

An Intrusion Detection System (IDS) is a technology which aims to detect actions that attempt to gain unauthorised access to a computer system. These attacks are also referred to as intrusions. That is why the term IDS was coined. According to M. F. Elrawy et al. [2], IDS for IoT can review data packets in real time and respond appropriately by examining them and notifying system authorities whenever a security breach is detected. This Intrusion Detection System (IDS) was designed to detect security threats to IoT services and will work in difficult conditions while also providing a quick response if any abnormal behaviour is detected, which is quite impressive. Given the significant advancements in IDS for IoT, this chapter presents a survey on Intrusion Detection for IoT, with the primary goal of deciphering the IDS categorisation taxonomy for IoT and determining which type of Intrusion Detection System is more appropriate for a given assault in a specific situation.

The rest of this chapter is structured as follows. Section 2 delves deeper into the Internet of Things (IoT) and its security challenges. Section 3 discusses the

---

B. R. Chandavarkar · Joshitha Reddy D. · S. L. Poojitha (✉) · R. T. Antony  
National Institute of Technology Karnataka Surathkal, Computer Science & Engineering,  
Mangalore, Karnataka, India

various kinds of IDS. Section 4 discusses the categorisation of IDS based on their deployment strategy and detection techniques used. Section 5 discusses the potential security threats in IoT. Section 6 examines suitable IDS in general for a specific attack, which is the survey's most important contribution. Lastly, some concluding remarks are made in Sect. 7.

## 2 Internet of Things (IoT)

IoT is a system of interconnected devices that allows smooth exchange of information between physical devices. These devices can be healthcare gadgets, vehicles, industrial products, wearable items, city infrastructures, and even everyday household items like kitchen appliances, baby alarms, temperature sensors, and smart TVs as shown in Fig. 1. To monitor such devices, there is no need to be in close proximity of them. This field has advanced as a result of the convergence of various technologies, including pervasive computing, inexpensive sensors, and machine learning.

IoT has evolved as one of the most critical technologies in the recent years. We can now link common objects to the Internet via embedded technology, allowing us to establish seamless communication between people, processes, and things. Items can share and gather data with very less human contact because of cheaper computers and advanced technologies like the cloud and big data analytics. Every interaction in today's super-connected environment can be noted and changed because of digital systems. IoT helps the digital and physical worlds that complement one other [3, 4].

### 2.1 Security Challenges of IoT

Security of IoT systems has become a hypercritical issue, because of the ever-expanding number of utilities and operators of these utilities in IoT networks. Smart devices are more efficacious when IoT peripherals and smartly set-up surroundings are combined. While diversity can provide users with a large number of devices to choose from, it is also one of the reasons for the IoT's fragmentation and many of its security concerns. Compatibility concerns have arisen as a result of the absence of industry foresight and standardisation, further complicating the security issue. Because of the portability of devices, there is a larger risk of attacks infecting several networks. The consequences of IoT security flaws are extremely unfavourable in indispensable fields such as health and industry [5].

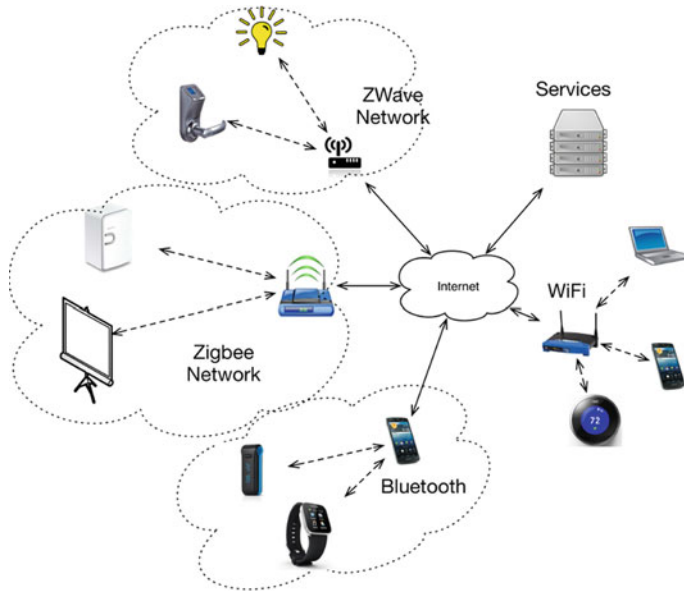


Fig. 1 Internet of Things

### 2.1.1 Factors Affecting IoT Security

*Vulnerabilities* The reason IoT devices are defenceless is because of the fact that the algorithmic capacity necessary for security is absent. The budget for finding and checking safe firmware is quite insufficient and this is another contributing factor to the extensive susceptibilities. The funds are decided based on the device prices and development cycles. As illustrated by Ripple20 and URGENT/11, vulnerable standard components harm millions of devices. Liabilities in web applications and related software for IoT devices, in addition to the devices themselves, could be the reason for a system whose integrity has been adversely impacted. Malicious software users are like hawks, always in search for new vulnerabilities, and are well-versed in previous ones.

*Malware* Despite the fact that most IoT devices have very less computer capacity, malware can still affect them. Impostors have made great use of this tactic in recent years. IoT botnet malware is both, a versatile and a profitable option for cybercriminals which makes it one of the most common types of malware. Cryptocurrency mining malware and ransomware are two other types of malware.

*Escalated Cyberattacks* Distributed denial-of-service (DDoS) attacks usually involve damaged devices. Using hijacked devices as an attack base is a way to infect new machines as well and hide malevolent activities or as an access point into a communal network for lateral mobility. Businesses are a more common target, but surprisingly there are quite a lot of attacks on smart homes as well.

*Information Threat and Unknown Exposure* Data leaks are the biggest threats in anything that involves the Internet. Connected gadgets are no exception to this. Without the user's knowledge, important data might be retained and used in these devices.

*Device Mismanagement* What further fuels these threats is security breaches, poor passwords, and just overall device mishandling. Also, not all users will be aware of the security measures that need to be taken and so the service providers and manufacturers will need to help their clients improve their security [5].

### 2.1.2 Emerging Security Issues

Due to a lack of foresight on the part of the industry, there was insufficient time to build tactics and countermeasures against common dangers in the emerging IoT ecosystems. To anticipate emerging issues, IoT security research must be done on a regular basis. Below are some of the new issues that need to be kept an eye on [5].

*Complex Environments* Complex IoT environments are characterised as a linked network of at least ten IoT devices. Because of its intricate web of interconnected processes, such an ecosystem is nearly impossible to manage and govern. In such a setting, an unnoticed misconfiguration can have disastrous effects, putting the physical security of the home at danger.

*Prevalence of Remote Work Arrangements* The Covid-19 pandemic shattered many aspirations. It ushered in large-scale work-from-home (WHF) arrangements for businesses all over the world, as well as a greater reliance on home networks. Many WHF users benefited from IoT devices. These changes have emphasised the need for IoT security approaches to be re-examined.

*5G Connectivity* There is a lot of excitement and expectation surrounding the move to 5G. It is a development that will help other technologies advance as well. The current focus of 5G research is on how it will influence companies and how they can properly use it.

## 3 Intrusion Detection System (IDS)

An IDS is a device that looks at incoming and outgoing network traffic for any signs of strange activity or security breaches. IDS solutions work by alerting the user to any activity that could affect the user's network.



## 3.1 *Types of IDS*

### 3.1.1 Network-Based IDS

NIDS are network devices that are set up at a predetermined location to examine traffic from all devices on the network [6]. Whenever strange behaviour is noticed by the IDS, the administrator is notified. NIDSs are either physical devices or software-based devices. They are linked to several network media such as Ethernet, FDDI, etc. There are two network interfaces available. The promiscuous interface is used for listening to the network conversations, and the other interface is used for control and reporting. When a network interface is set to promiscuous mode, all packets, including those not intended for the network interface card's MAC address, are delivered to the kernel for processing.

As the count of Internet nodes has increased dramatically in recent years, NIDSs have become a vital component of network security management. It can cause high-speed network traffic overflow, signature creation lag time, encoding, and scaling problems.

### 3.1.2 Signature-Based IDS

It uses a list of known threats and associated indicators of compromise (IOCs) that has already been encoded into the system. As packets move through the network, a SIDS cross checks the packets with a database of known IOCs or attack signatures, flagging any unusual behaviour.

An example of SIDS is SNORT. There are five components in the system: Packet Decoder, Preprocessor, Detection Engine, Logging and alerting system, and output modules [7]. The packet decoder gathers packets from various network ports and forwards them to the preprocessor. The preprocessor modifies the packets before sending it to the detection system. Another function of the preprocessor is defragmentation of the packets. The detection system mainly works to find out if there is an intrusion activity based on the rules defined by SNORT. In the logging and alerting system component, based on the detection engine's results, a packet is either used to generate an alert or the activity is logged. The output module saves the results of the previous component.

Signature classifications are based on previously identified intrusive behaviour. As a result, the user may quickly analyse the signature database and decide which kind of intrusive behaviour the abuse detection system is set to alert on [8]. When you install the Misuse Detection System, it immediately starts protecting your network. There are minimal false positives as long as assaults are accurately described in advance. When an alert raises, the user can immediately associate it with a specific type of network activity. For these reasons, SIDS is a worthwhile IDS.

But one of the biggest issue of SIDS is managing the traffic as each packet is compared with every single signature in the database. Therefore, it is a time-consuming process. The database has to be frequently updated to make sure all possible attack signatures have been tracked. Another problem is that the database will be very environment-specific. This is because the attack information is dependent on OS version and application.

### 3.1.3 Anomaly-Based IDS

An AIDS uses machine learning to train the detection system to recognise a normalised baseline rather than seeking for recognised threats. Rather than looking for known IOCs, AIDS simply detects any unusual behaviour and sends out alerts [9].

The different types of anomaly-based IDS are host-based anomaly and network-based anomaly. The calculation of host-based anomalies dealt with operating system call traces. The incursions take the form of anomalous subsequences of the traces (collective anomalies). Malicious programming, unlawful activity, and policy abuse are among the consequences. The data are ordered, and the alphabet is made up of specific system functions like open, close, and create. Some network type anomalies are UDP flood, ICMP flood, etc.

A UDP flood attack is a type of DoS attack. It involves sending a huge number of UDP packets to a remote host's random ports. As a result, the remote system will look for a programme that is listening on this port. The host will respond with an ICMP "Destination Unreachable" message if no application is listening on the port [10]. As a result, the affected system will be forced to send a huge number of ICMP packets in response to a high number of UDP packets, eventually rendering it unreachable by other clients. The system will go down if sufficient UDP packets reach the victim's ports. To detect a UDP flooding assault, the amount of the traffic (flow) and the count of packets (packet count) in incoming traffic must be used.

ICMP flood is a simple sort of attack in which the attacker sends a huge number of ICMP Echo Request (ping) packets of various sizes to the target host. The Ping-of-Death (PoD) assault was succeeded by ICMP flooding. PoD attempts to send an extra-large ping packet to the target in the hopes of crashing the system due to its inability to handle large ping packets [10]. Ping flood takes the attack to a new level by flooding the victim with a massive amount of ping traffic. The attacker expects that the victim will be too preoccupied with responding to ICMP Echo Reply packets, using both outgoing and incoming server bandwidth.

### 3.1.4 Distributed-Based IDS

DIDS is made up of numerous IDSs scattered over a broad network that connect one another or with a central server for better network monitoring, scenario analysis, and real-time assault data. The DIDS architecture combines centralised data analysis

with distributed monitoring and data minimisation. This is a one-of-a-kind approach among current Intrusion Detection systems. There is a DIDS Director. A single Host Monitor exists for each host along with single LAN Monitor [11]. The Host and LAN Monitors are mainly in charge of gathering evidence of unauthorised or questionable behaviour, while the DIDS Director is in charge of aggregating and evaluating it.

### 3.1.5 Host-Based IDS

Only the device’s incoming and outgoing packets are monitored by a HIDS [12], which alerts the administrator if possible fraudulent behaviour is detected. HIDSs, unlike NIDSs, have easy accessibility to data and system activities targeted by these attacks, and thus they are aware of the potential consequences.

An attacker may tamper with a host-based IDS, which is a worry. The IDS cannot be trusted if an attacker gets control of a system. As a result, unique anti-tampering protection for the IDS should be built into the host. There are a few issues with HIDS. First, a large amount of resources is utilised. This in turn affects the system’s performance. Also, the detection of the attack will not happen until it has reached the host. Usually, host-based and network-based IDSs are used together.

## 4 IDS for IoT

This section will discuss the different classifications of IDS for IoT based on placement strategies and detection methods as illustrated in the flowchart below (Fig. 2).

### 4.1 Placement Strategies

Before diving into placement strategies, it is critical to understand the structure of IoT and its components. According to B. B. Zarpelão et al. [13], the architecture

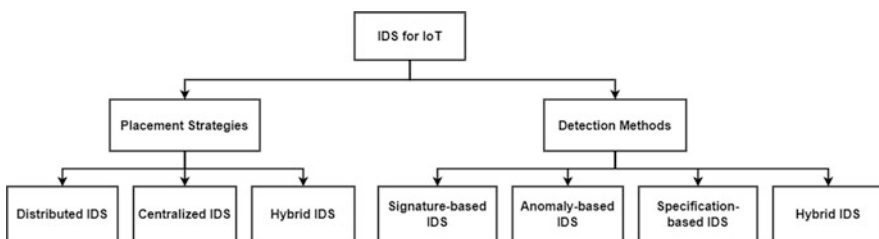


Fig. 2 IDS classification for IoT

is primarily made up of three domains: physical, network, and application. The physical level is made up of devices that perceive and connect with our surroundings, forming an LLN. The primary goal of the network domain, i.e., the second domain, is to bring together traditional network technologies and protocols for data transmission. The interfaces that allow users to interact with items in the physical domain are included in the application domain. Placement strategies deal about the location of IDSs in the IoT network. Three different placement options for IDSs are described in the subsections below.

#### **4.1.1 Distributed IDS Placement**

Le et al. [14] state IDSs are installed in every node of the LLN in this distributed placement technique. The nodes are in charge of keeping an eye on their neighbours. These nodes are categorised as leader, affiliated nodes forming a hierarchy within them. They can alter their roles depending on the type of attack. In this process, each node keeps track of a higher level node by calculating its inward and outward amount of data which moves across a network. So, whenever a security threat is identified by the IDS, it alerts all the other nodes to isolate the attacker. According to A. Khraisat et al. [6], this distributed IDS is very effective in detecting DOS assaults for high-speed networks.

#### **4.1.2 Centralised IDS Placement**

In this strategy, IDS is placed in a centralised component. Raza et al. [15] say that a central console is in charge of all IDS actions. All of the data gathered by LLN nodes are transmitted to the Internet and would pass through the border router. As a result, an IDS installed here can examine all the data flowing between the architecture components. In comparison to a distributed system, the cost of maintenance and administration is reduced. And Liu et al. [16] indicated that, most importantly, it is unable to detect harmful events occurring in multiple locations at the same time.

#### **4.1.3 Hybrid IDS Placement**

It is a hybrid of both distributed and centralised placement strategies, as the name implies. It combines centralised and distributed placement approaches to maximise their benefits while minimising their drawbacks. Lee et al. [17] proposed that the network is divided into clusters or regions using this hybrid placement method, with only the cluster's main node hosting an IDS instance (only selected nodes that are effective will be used for hosting). This node is in charge of monitoring the other nodes in the cluster. Cluster members should provide relevant data about themselves and their neighbours to the cluster leader. This technique is constructed with more resources than the previous placement strategies. Thanigaivelan et al. [16] proposed

a hybrid deployment model for IDS placement in both network nodes and border routers. It differs from the previous one because of its central component. The border router's IDS modules handle tasks that necessitate more resource capacity, whereas standard node IDS modules are frequently lightweight. Both of these methods have a number of advantages over other methods of placement.

## **4.2 Detection Methods**

Depending on the sort of detection method utilised in the system, Intrusion Detection techniques are divided into distinct categories. This chapter will go through them briefly in the subsections that follow.

### **4.2.1 Signature-Based Approach**

This approach detects the assaults by comparing the system behaviour with the predefined attack signatures in the database. As stated by Kasinathan et al. [18], it functions by using a pre-programmed list of threats and IOCs. File passwords, fraudulent URLs, and the content of subject line headings of emails are all the instances of IOCs. A signature-based IDS compares packets as they pass through the network to a list of known IOCs or threat patterns to detect any unusual behaviour. System raises an alert whenever if an activity matches with the predefined attack signatures or saved IOCs. Study [13] states that these kinds of IDS are highly good at spotting known threats. But the disadvantage is that, since there are no known matching signatures for all assaults, this technique is incapable of detecting new attacks or versions of current ones, which is one of the most serious flaws of this IDS.

### **4.2.2 Anomaly-Based Approach**

As stated by Mitchell et al. [19], anomaly-based approach has a normal behaviour which is already prewritten and it detects any illegal activity whenever the divergence of the system behaviour from the normal one exceeds a limit. This method works well for detecting new assaults, especially those involving resource exploitation. usually, ML algorithms are used to generate the normal behaviour. From source [8], basically, any activity which does not coincide with the normal behaviour is identified as an illegitimate action. Also, Study [2] states that the disadvantage of this approach is that many non-malicious actions will be identified as attacks simply because they are out of the usual. Hence, the heightened possibility of false alarms with anomaly-based intrusion detection necessitates more resources and time to evaluate all possible threat alarms.

### 4.2.3 Specification-Based Approach

A specification is a set of rules and criteria that govern how network components should behave. Study [13] proposes that when network activity deviates from specification definitions, specification-based techniques detect intrusions. As a result, it serves the same function as anomaly-based detection in terms of detecting deviations from the norm. But, there is a key difference between the two methods: in specification-based approach, we will have to manually state the rules and criterion of each specification unlike the anomaly approaches. These manually created requirements are utilised to characterise legitimate software behaviours in this method. Study [20] says this technique does not raise false alarms when unexpected (but legitimate) software actions are encountered because it is based on legitimate activities. As a result, it has a lower false positive rate than an anomaly-based IDS. It also has the ability to detect previously unknown assaults because it detects attacks as deviations from legitimate behaviour. As a consequence, it has a lesser probability of false detection compared to anomaly-based intrusion detection system. It can also identify previously undiscovered attacks since it recognises attacks as aberrations from normal behaviour.

### 4.2.4 Hybrid Approach

There have been significant developments in this hybrid intrusion detection system, which combines the attributes of all detection techniques to optimise their benefits while minimising their shortcomings. INTI, proposed by Cervantes et al. [21], is a promising and efficient hybrid IDS method for detecting sinkhole attacks, combines an anomaly-based methodology for analysing the packet traffic between nodes with specification-based approaches for extracting two forms of node evaluation: reputation and trust. Numerous studies show that INTI exceeds other methods in terms of system performance in combating sinkhole attacks.

## 5 Security Threats

The impact of IoT security threats could be a serious challenge in IoT implementation. Cybercriminals can use security flaws in IoT infrastructure to launch sophisticated cyber-attacks. Most users are unaware of the security threats, and hence do not have the means to prevent them. A few of the threats are:

### 5.1 Botnets

A botnet basically tries to gain remote access to a user's computer and spread malware [22]. Botnets are used by attackers to steal private data to initiate cyber

assaults such as DDoS and phishing. The Mirai botnet is one such botnet that affects IoT systems. A total of 2.5 million devices, including photocopiers, modems, and webcams, were impacted. This botnet was also used by intruders to perform widespread denial of service attacks against various IoT devices. Following the effect of Mirai, a number of cybercriminals have created a number of complex IoT botnets. These botnets are capable of launching sophisticated cyber-attacks on IoT devices that are vulnerable.

## ***5.2 Denial of Service (DoS)***

The main purpose of this assault is to slow down the server. It attempts to do so by sending multiple requests and causing an overflow in the victim's system [23]. A denial-of-service attack, for example, will prohibit a travel agency from accepting requests for new ticket reservations, vehicle condition inquiries, and booking cancellations. In such instances, people may opt to travel with alternative agency. The attacker effectively harmed the company's reputation in this manner.

## ***5.3 Man in the Middle (MITM)***

In this form of assault, a hacker tries to intercept the messages between two communicating systems. They hack into the communication channel between the two and hence get in the "middle" of them [24]. Attackers seize control and send fake messages to systems that are a part of the communication channel. Such attacks can be used to compromise IoT devices like smart refrigerators and self-driving cars.

MITM may be used by intruders to capture communications between several IoT devices, ending in major failure. Smart home accessories such as fans, for example, can be turned on and off by an attacker via MITM. Attacks on IoT devices, such as industrial equipment and medical devices, might have severe repercussions.

## ***5.4 Identity and Data Theft***

Attackers may now use IoT devices such as smart wristbands and smart home appliances to get more information on a range of people and businesses. Intruders can utilise this data to commit more intricate and thorough identity theft [25].

Cybercriminals, for instance, can gain access to a company's corporate network by exploiting a flaw in an IoT sensor. As a result, attackers have access to critical data from multiple organisational structures.

## 5.5 *Social Engineering*

Social engineering is used by attackers to induce individuals to divulge personal information such as passwords and bank account details. Cybercriminals may also utilise social engineering to get access to a system and discreetly install malicious software. Typically, social engineering assaults are carried out through the use of phishing emails, in which an attacker must create convincing emails in order to manipulate others [26]. In the case of IoT devices, however, social engineering assaults may be easier to carry out.

In order to give customers with a personalised experience, IoT devices, especially wearables, collect massive quantities of personally identifiable information (PII). Users' personal information is also used by such gadgets to provide user-friendly services, such as ordering things online using voice control. However, attackers can access PII to obtain sensitive information such as bank account numbers, purchasing history, and home location. This information might be used by a cybercriminal to execute a sophisticated social engineering attack against a person, his family, and friends via vulnerable IoT networks [26]. In this approach, IoT security concerns such as social engineering might be used to get unauthorised access to user data.

## 5.6 *Routing Attacks*

Routers play an important role in communication networks by allowing data transfer. Router attacks can take advantage of protocol weaknesses, incompatibilities in router architecture, and inadequate authentication. There are two sorts of attacks that can occur: distributed denial of service and brute force attacks [27]. While an attack is taking place, it has an effect on the system operations and corporate operations.

There are different types of routing attacks such as sinkhole attack, selective forwarding attack, etc. Sinkhole attacks are the most damaging routing assaults in the IoT context, among others [21]. It generates network traffic and dissipates network communication. It made use of a variety of routing metrics. Fake link quality, shortest path, and other criteria are used. Sinkhole attacks generate fictitious data and send routing requests to nearby nodes. The nodes were compromised as a result of this assault.

## 6 **Analysis of Suitable IDS**

In the previous sections, the different types of detection methods and the placement strategies that are used in IDS were discussed. Along with that the characteristics of various security threats were seen. Based on the assessment of these, the following observations have been made.



**Table 1** Summary of appropriate IDS types for various IoT attacks

S. No	Security attack	Placement strategy	Detection technique
1	Botnet attacks	Distributed-based	Specification-based
2	DoS attacks	Distributed-based	–
3	Man in the middle	Centralised	Anomaly-based
4	Sinkhole attacks	Distributed, centralised	Hybrid
5	Wormhole attacks	Distributed	–

It was found that botnets that launch Distributed Denial-of-Service [DDoS] attacks, which are triggered by internet traffic overflow, can be better spotted using Distributed IDS as this type of IDS identifies attacks based on inbound and outgoing traffic. Furthermore, because signature-based approaches cannot detect new threats and anomaly-based approaches have a high false positive rate, a specification-based approach could be employed as a detection technique here. Also, because DoS assaults have many of the same traits, these strategies are a better fit for them too. As man-in-the-middle (MITM) attacks are known to use any available technique suitable to the attacker to intercept, decrypt and exploit user's resources, anomaly-based stimulative IDS can better identify these type of attacks since they have the ability to detect out-of-the-ordinary patterns and are better at detecting resource exploitation attempts. Moreover, since this attack has no precise requirements, a centralised IDS as a placement strategy could be used as it would be less expensive than a distributed-based approach.

And as mentioned in Sect. 5.6, among the many routing attacks that exist, sinkhole attack is one. It is one of the deadliest attacks as it may act as a catalyst for other attacks. For such an attack, hybrid detection methods are more suitable. INTI [28] is a very efficient hybrid-based IDS that is used for sinkhole attacks. Wormhole attack is another routing attack. In this case, the node is targeted from many directions, making it difficult to pinpoint the location of the intruder. For this reason, a distributed IDS is suitable for this type of attack. These are summarised in Table 1.

## 7 Conclusion and Future Work

IoT has exalted expectations due to its ability to transform physical items from many application areas into Internet hosts. Intruders, on the other hand, may make use of the IoT's enormous potential by considering it a new means to endanger the privacy and security of users. Thus, IoT security solutions should be developed and IDS is one of the most important security technologies for IoT. Through careful survey, this chapter has managed to review current classifications and existing methodologies, with the help of which the most appropriate IDS for a particular attack on IoT is proposed.

Future research could concentrate on the following topics: (1) investigating more about the strengths and weaknesses of various detection methods and placement strategies, (2) developing IDS for social engineering threats, (3) addressing more IoT technologies, (4) investigating about ANN based IDS systems for Routing attacks [28], and (5) improving alert traffic and management security.

## References

1. A. A. Ansam Khraisat, A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges, <https://cybersecurity.springeropen.com/articles/10.1186/s42400-021-00077-7#Sec59>, [Accessed: 18-01-2022] (2021).
2. H. F. A. H. Mohamed Faisal Elrawy, Ali Ismail Awad, Intrusion detection systems for IoT-based smart environments: a survey, <https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-018-0123-6>, [Accessed: 18-01-2022] (2018).
3. Internet of Things, [https://en.wikipedia.org/wiki/Internet\\_of\\_things](https://en.wikipedia.org/wiki/Internet_of_things), [Accessed: 09-02-2022] (2022).
4. What is IoT?, <https://www.oracle.com/in/internet-of-things/what-is-iot/>, [Accessed: 09-02-2022] (2022).
5. IoT Security Issues, Threats, and Defenses, <https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/iot-security-101-threats-issues-and-defenses>, [Accessed: 09-02-2022] (2021).
6. Intrusion detection system (IDS): What it is; its types, <https://www.geeksforgeeks.org/intrusion-detection-system-ids/>, [Accessed: 19-01-2022] (2022).
7. Sagar N. Shah\* Ms. Purnima Singh M.E. (Computer Science & Engineering), Assistant Professor, Computer Science & Engineering, Parul Institute of Engineering & Technology, Parul Institute of Engineering & Technology, Vadodara, Gujarat, India Vadodara, Gujarat, India, Signature-Based Network Intrusion Detection System Using SNORT And WINPCAP, International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 10, December- 2012 ISSN: 2278-0181 (2012).
8. Intrusion Detection System (IDS): Signature-based vs. Anomaly-based, <https://www.n-able.com/blog/intrusion-detection-system>, [Accessed: 19-01-2022] (2021).
9. S. A. Tamara Saad Mohamed, IoT-Based Intrusion Detection Systems: A Review, <https://www.tandfonline.com/doi/abs/10.1080/23080477.2021.1972914?journalCode=tsma20>, [Accessed: 18-01-2022] (2021).
10. Vasima Khan (Computer Science & Engineering), All Saint Inst. of Tech, Bhopal, M. P., India; Anomaly based Intrusion Detection and Prevention System, International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 3, March - 2013 ISSN: 2278-0181 (2013).
11. Steven R. Snapp, Stephen E. Snaha- Haystack Laboratories, Inc. Daniel M. Teal, Tim Grance- United States Air Force Cryptologic Support Center, The DIDS (Distributed Intrusion Detection- System) Prototype, Summer '92 USENIX- June E-June 12, Igg - San Antonio, TX (1992).
12. K. Letou, D. Devi, Y. Jayanta, Host-based intrusion detection and prevention system (hidps) (05 2013). <https://doi.org/10.5120/12136-8419>.
13. Bruno Bogaz Zarpelão, Rodrigo Sanches Miani, Cláudio Toshio Kawakani, Sean Carlisto de Alvarenga, 2018. A survey of intrusion detection in Internet of Things, <http://www.ttcenter.ir/ArticleFiles/ENARTICLE/10201021.pdf>, [Accessed: 18-01-2022] (2018).
14. Intrusion Detection in IoT, <https://securityboulevard.com/2021/12/intrusion-detection-in-iot/>, [Accessed: 18-01-2022] (2021).

15. Raza, S., Wallgren, L., Voigt, T., 2013. SVELTE: real-time intrusion detection in the Internet of Things. *Ad Hoc Netw.* 11 (8), 2661–2674. (2013).
16. Liu, C., Yang, J., Zhang, Y., Chen, R., Zeng, J., 2011. Research on immunity-based intrusion detection technology for the Internet of Things. In: *Natural Computation (ICNC), 2011 Proceedings of the Seventh International Conference on*, Vol. 1, pp.212–216 (2011).
17. Lee, I., Lee, K., 2015. The internet of things (IoT): applications, investments, and challenges for enterprises. *Bus. Horiz.* 58 (4), 431–440 (2015).
18. Kasinathan, P., Costamagna, G., Khaleel, H., Pastrone, C., Spirito, M.A. 2013b. DEMO: an IDS framework for internet of things empowered by 6LoWPAN. In: *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, CCS '13*, ACM, New York, NY, USA, pp. 1337–1340 (2013).
19. Mitchell, R., Chen, I.-R., 2014. A survey of intrusion detection techniques for cyberphysical systems. *ACM Comput. Surv. (CSUR)* 46 (4), 55. (2014).
20. Le, A., Loo, J., Luo, Y., Lasebae, A., 2011. Specification-based IDS for securing RPL from topology attacks. In: *Wireless Days (WD), 2011 IFIP*, pp. 1–3 (2011).
21. Cervantes, C., Poplade, D., Nogueira, M., Santos, A., 2015. Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things. In: *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 606–611 (2015).
22. Hemanth, Jude, Xing, Ying, Shu Hui, Zhao Hao, Li Dannong, Guo Li, 2021, 2021/04/15, Survey on Botnet Detection Techniques: Classification, Methods, and Evaluation 6640499 2021 <https://doi.org/10.1155/2021/6640499>, *Mathematical Problems in Engineering Hindawi* (2021).
23. Kasinathan, P., Pastrone, C., Spirito, M., Vinkovits, M., 2013a. Denial-of-service detection in 6LoWPAN based Internet of Things. In: *Wireless and Mobile Computing, Networking and Communications (WiMob), 2013 IEEE Proceedings of the 9th International Conference on*, pp. 600–607. (2013).
24. Farouq Aliyu, Tarek Sheltami, Ashraf Mahmoud, Louai Al-Awami, Ansar Yasar, “Detecting Man-in-the-Middle Attack in Fog Computing for Social Media” Vol.69, No.1, 2021, pp.1159-1181 (2021).
25. E. Aïmeur, D. Schönfeld, The ultimate invasion of privacy: Identity theft (2011). <https://doi.org/10.1109/PST.2011.5971959>.
26. C. Bhusal, Systematic review on social engineering: Hacking by manipulating humans. *journal of information security* (2021).
27. Cervantes, Christian; Poplade, Diego; Nogueira, Michele; Santos, Aldri (2015). [IEEE 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM) - Ottawa, ON, Canada (2015.5.11-2015.5.15)] 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM) - Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things. (), 606–611. <https://doi.org/10.1109/INM.2015.7140344> (2015).
28. S. V. V. Sharma, Aiemla: artificial intelligence enabled machine learning approach for routing attacks on internet of things. (2021). <https://doi.org/10.1007/s11227-021-03833-1>.

# Human-in-the-Loop Control and Security for Intelligent Cyber-Physical Systems (CPSs) and IoT



Sanjkeet Jena, Sudarshan Sundarrajan, Akash Meena,  
and B R Chandavarkar

## 1 Introduction

It is a fact that the rapid advent and evolution of technology has revolutionized life in the modern world. Almost all the devices people use today are interconnected to this magnanimous network known as “the Internet” [1]. The continuous growth of technology usage and the increased connectivity through the Internet have paved the way for the idea of Internet of Things. Internet of -Things, or IoT, is a network of sensor equipped objects which have some computational power and use the Internet to exchange data and information [2]. This concept stems from a networking perspective and the idea of connecting the digital world with human livelihoods. On the other hand, cyber-physical systems (CPSs) [3], originating from an engineering perspective, refers to systems that consist of computational elements that work together in controlling physical processes. They deal with the monitoring and control of physical environments and phenomena and are tightly bound to the physical environment.

### *1.1 Internet of Things and Cyber-Physical Systems: A Common Ground*

Despite the contrasting origins of the two terminologies “IoT” and “CPS,” the plethora of similarities between them have led to a unifying model that can be used to describe both CPS and IoT systems [4]. Both comprise interacting digital,

---

S. Jena · S. Sundarrajan (✉) · A. Meena · B. R. Chandavarkar  
National Institute of Technology Karnataka, Mangalore, India

physical and human components, which work together on integrated logic and physics. This unified perspective gives an opportunity for unified approaches for both CPS and IoT design, and with respect to the paper's objectives, it makes things easier to design and apply a united framework.

## ***1.2 Introducing the Idea of Human-in-the-Loop with IoT/CPSs***

The human-in-the-loop cyber-physical system (HiTLCPS)/human-in-the-loop Internet of Things (HiTLIoT) [5] integrates the human into the feedback loop so that the human can participate in each phase: monitoring (assisting the system in identifying situations in the environment), analysing (collaborating with the system in decision-making), planning (assisting the system in planning a set of changing actions) and executing (handling, directing or executing tasks). This system's goal is to respond to specific human needs while taking into account people's goals, behaviours (past and future), emotions and psychological states [6].

The involvement of humans in these systems exposes a large difference between HiLT systems and traditional systems, due to the aforementioned human behaviours. From a security standpoint, bringing humans into the loop can increase the points of failure in such systems. Traditionally, humans were viewed as the "problem" [7], which led to system designers neglecting the crucial roles that a human can bring to the table and going towards automated systems that require minimal human intervention. This work, contrary to the earlier belief, proposes a unique "human-centric" framework to mitigate these issues, enabling humans as crucial components to make the CPS/IoT systems more robust and secure. Through this, the work tries to focus on the aspect of how "humans are a solution." Furthermore, the chapter looks into various components that make up the framework, how they are interlinked and how security can improve with a human in the loop combined with this framework.

The remainder of the chapter is organized as follows: Section 2 deals with the relevant work done in areas regarding the problem statement. Section 3 introduces the framework and its components and compares it with the existing frameworks. Along with that, a use case scenario is discussed to understand the application of the framework. Conclusions and future work are discussed in Sect. 4.

## **2 Related Work**

This section is split into two subsections to elaborate on the following:

- Security and Privacy Issues that affect IoT and CPS
- Human-in-the-Loop-based Security Models of CPS and IoT

## ***2.1 Security and Privacy Issues in IoT and CPS***

The unprecedented rise in connectivity and devices of IoT and CPS brings along a lot of security challenges with it. Given that such devices have a physical and a digital component to them, they are vulnerable to both physical attacks and cybersecurity threats. This section presents ideas on the threats on IoT and CPS systems that have been discussed in research papers and surveys. CPS threats can be broadly classified into three categories: perception, communication and application [8]. Perception threats deal with threats on the physical components of the CPS like sensors and so on. Threats like electromagnetic interference, failures in the equipment, DoS attacks, tampering of data and so on come under this category. Communication threats involve threats around the communication lines of CPS devices. Application threats revolve around the information that is collected from users at the application layer. Kim et al. [8] also discuss methods to preserve security in these systems, which includes hardware anchor based solutions, digital fingerprinting and malicious code detection systems.

Rachit et al. [9] also survey various attacks and the security models that are applied to secure IoT devices. Threats like DDoS, Botnets, Interception and Man-in-the-Middle attacks and weak encryption of payloads are discussed. Security models that incorporate methods like data encryption and block chain-based authentication are looked into and checked against the CIA (Confidentiality, Integrity and Availability) triad, along with trust and authenticity.

Apart from the security aspects mentioned above, the enormous amount of data that flows through the networks can cause privacy concerns. Intelligent and automated systems depend on data to assess user interaction and accordingly modify their working. Data can be used to predict the activities and behaviours of the users linked to IoT and CPS, which can enable malicious use of data by adversaries. Therefore, a focus on privacy-aware IoT applications has also been of interest in the research circles [10, 11]. An example for this is illustrated in [11], which deals with a privacy-aware architecture which relies on a trusted privacy cloudlet layer using an encoder–decoder LSTM model to obfuscate and anonymize the data that are collected.

Given that this chapter proposes a security framework that involves HiTLIoT/HiTLCPS systems, it becomes important to understand the various security threats that plague such systems and understand the mitigation policies that are employed to counter the threats. This in turn helps in bolstering the strength of the framework in protecting the systems.

## 2.2 *Human-in-the-Loop-Based Security Models of CPS and IoT*

Research in the direction of secure systems with humans in the loop is relatively minimal, but frameworks and models have been proposed to help build a secure intelligent IoT/CPS. In this subsection, these frameworks are discussed.

A framework that enables building secure systems keeping humans in the loop is proposed by Cranor [12]. The framework is designed to identify threats that come with keeping a human in the loop and help mitigate them. The framework is divided into four segments: Communication (deals with informing users about a particular security task by triggering some form of communication like warnings, status indicators), Communication Impediments (which disturbs the flow of communication and causes a failure), the Human Receiver (who receives the communication and impacts the security of the system) and Behaviour (revolves around the action taken by the receiver). This framework gives a baseline with which systems can be secured while focusing on the human aspect.

Another framework, proposed by Rohan et al. [13], deals with a 4-layer consumer-oriented security and privacy-preserving (SPP) model, which only involves the end user interacting with the IoT system. The four layers include Discovery (discovers the privacy properties of services active near the user and opens a communication channel for the user to interact with the services), Inference (gives the user an idea about what type of data is being used by systems and for what purpose), Choice (helps users to make a decision based on their choices) and Communication (communicates with the user based on outputs from the previous layers).

A privacy-preserving approach for a human-in-the-loop IoT system is discussed in the paper proposed by Rivadeneira et al. [14]. This framework is mainly focused on the privacy issues that users face nowadays because of data sharing to IoT devices through smartphone sensors. PACHA provides full privacy data sharing design between the user and IoT. It has two main parts: PACHA Privacy Orchestrator (PPO) and PACHA Privacy Interagent (PPI). The main contribution of PPO is to make connections between the user and IoT platforms. PPI is an application that allows the user to find IoT resources nearby and establish privacy preferences. The paper uses this framework in a platform named ISABELA which uses the concept of HiTLCPS to monitor students and assist them in improving their academic performance. Though it is used in one specific application, the idea behind PACHA can be utilized for different systems.

Brown et al. [15] talk about an intelligent, dynamic and adaptive intrusion detection system to deal with the constantly evolving threats that CPS/IoT systems have to face. Their framework called Blacksite combines human intelligence with a Deep Neural Network and Artificial Immune System. Their detector first detects suspicious or malicious content, the Deep Neural Network examines it further, after which the human-in-the-loop process follows. Similarly, Elmalaki et al. [16] talk about dealing with Context-Aware Adaptation Based Spyware (SpyCon). SpyCon

is a type of spyware that tries to get sensitive user information by observing the HiTLCPS/HiTLIoT systems as these systems have humans in their feedback loop. They provide VindiCo, a privacy framework to deal with it using a detection technique based on the mutual information got from context-based decisions of the user. Then, it employs a detection engine to provide a suspicion score which can help determine if the SpyCon has attacked the system.

The solutions discussed above cover a good range of security and privacy aspects that need addressing in the HiTL systems. Utilizing the framework from [12] will help in identifying potential causes of failure and problem areas that can affect the system in the future. It emphasizes on the importance of users understanding what all security and privacy options are available to them in consumer IoT devices and be aware of the threats that can affect the system. Papers [15] and [16] present intrusion detection systems that can help in detecting any kind of intruding attacker on a particular system. All these frameworks and models help in understanding different approaches that can be taken to solve human-in-the-loop security.

### **3 A Security Framework to Protect Systems**

This section presents a security framework that caters to the end user as well as incorporates a human-in-the-loop security service that can help prevent any malicious attack on the system. The framework proposed in [13] is closest to what this paper wishes to accomplish. However, this SPP framework caters only to how the end user interacts with the system and does not provide any method to incorporate a human in the loop who provides the security rather than enjoying the security services provided. The framework proposed in this paper aims to involve a human expert into it and discuss how the human can help in providing security. Moreover, a use case scenario is discussed to understand how the framework can be applied

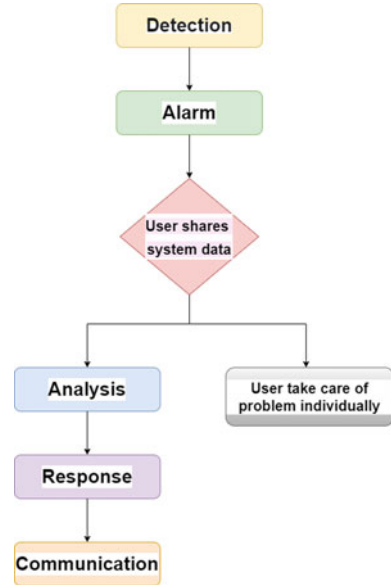
#### ***3.1 Framework Components***

This framework is divided into five phases: Detection, Alarm, Analysis, Response and Communication phase (DAARC). Each phase has been discussed in detail in the following subsections. Figure 1 shows a flow of how the framework will function.

##### **3.1.1 Detection**

The first and the most critical step of our framework is threat detection. Our framework needs to detect threats continuously so that it can respond promptly. A real-time threat detector is utilized to look for threats continuously and immediately



**Fig. 1** DAARC framework

respond once a security threat is found. Depending on the system's use case, various threat and anomaly detection systems can be utilized in order to check whether the system is infected or not. For this phase, the use of machine learning-based approaches, as discussed in the papers [17–20], can help automate the process and capture far more information than that visible to the human eye. This work does not delve into the exact methods that are used for any anomaly detection nor does it provide an all-encompassing detection system, as there are multiple solutions which cater to different systems and use cases. System designers, depending on the type of system being developed, have the freedom to select an appropriate solution in this phase of the framework. However, while choosing machine learning-based models for detection, system designers must ensure that the model is robust against adversarial samples [21].

### 3.1.2 Alarm

In the second phase of our framework, the system goes into a distressed state where it will only maintain its core functions and stop using its non-essential processes. This ensures that the threat does not start infecting other system components. After that, the user is notified of the intrusion. Then, the user is asked to share log files and system records to deal with the intrusion. If the user declines, then the user is provided with two options:

1. Solve the threat by themselves.
2. Let the system go into a hibernation state where only the core physical functions will remain active.

This way, the CPS/IoT system would become isolated and would have no connectivity with the outside world. This way, the user can use the device for the basic necessities, not worry about the threat connecting to the device's network, and wait till the issues get fixed by the company providing the CPS/IoT service or any other third party.

### 3.1.3 Analysis

If the user has consented to give the log files and the system data to the analyst, the analyst has to decide the best course of action. The analyst is a human who is well versed with security and cryptographic techniques and can deal with security threats aptly. Along with that, an autonomous response controller (ARC) is used to provide an appropriate response to the detected threat. The ARC uses a Competitive Markov Decision Process to model the appropriate response [22]. The Human Analyst is the human in the loop who will help bolster security by choosing the appropriate response by taking the security threat into account. Most consider humans in CPS/IoT a liability, but the proposed framework uses human capabilities and decision-making to bolster security.

### 3.1.4 Response

The Analyst has to consider all the factors involved and deal with the compromised system. The ARC can provide some model responses, but in the end, the Analyst has to decide what should be the most appropriate one. There can be various responses that the Analyst can take. The Analyst might decide to mitigate the intrusion if the depth of intrusion is not too much else; the Analyst might have to shut down and reformat the system if the degree of intrusion is very high. After choosing the response, the Analyst must quickly implement the chosen response to upkeep the compromised system. This step continues until the system is detected to be intrusion free. The Analyst is considered as the final authority because of the following reasons:

- (i) The current detection and the response machine learning (ML)/deep learning (DL) frameworks are unreliable and cannot be depended upon to make the right choice as they can be subject to faulty responses through adversarial examples [21].
- (ii) The human Analyst has access to various logs and system history, which can help weed out false positives detected by the automatic detection and response ML/DL frameworks.

- (iii) The log files and system history can also help the Analyst make a better and more informed decision that will be better than using just the automatic detection and response ML/DL frameworks.

### 3.1.5 Communication

Communication is essential for the proper functioning of any system. Our framework incorporates this by adding a communication section in our framework. After the analyst resolves the problems and fixes the intrusion, a report is created in the log files and sent back to the user. The report created can be used further if a similar intrusion is detected. Developers of CPS/IoT systems can also use the report to update and improve the architecture to remove the security vulnerabilities used for intrusion. One thing to note is that the end user should not feel nagged by the excessive notifications. This issue can be resolved by giving an option to the end users to select their notification preferences.

## 3.2 *Comparison with Existing Approaches in HITLIoT/HITLCPS Security*

The proposed DAARC framework is a security framework that aims to incorporate the human in the loop as a security solution rather than a human who requires the service. This is a major differentiator between most of the papers that the authors of this work surveyed in order to get an idea of the state of the art in the concerned domain. Papers [12] and [13] both deal with just a human-in-the-loop end user and not a human security analyst who provides the security to the system. The approach taken in [12] discusses how the user gets the communication from the system being used and makes use of the information to take a decided choice in a security-related task. It majorly focuses on how to communicate any threat-related information to the user and ensure that the user takes measured decisions to prevent any security threat. DAARC, on the other hand, deals with threat detection, analysis and the corresponding response. The human here is a security expert who helps in both analysis and the response to any intrusive threat. In terms of similarities, the humans in the loop for both the framework—the user in [12] and the analyst in DAARC—are the decision makers for securing the systems.

In a similar sense to [12], the framework in [13] also focuses on the end user as the human in the loop whose security and privacy is protected using a 4-stage framework. This framework helps the user discover the various security and privacy options that the IoT system offers to any user and gives the user the freedom to choose which options are better suited for the scenario. Similarities with DAARC include a human decision maker for the security/privacy of the system. DAARC differs in that it employs a human security analyst who takes care of the decision-

making process to mitigate the risk and does not involve any user input for this. As a result, it is the analyst who is in charge of securing the system and not the end user.

In paper [14], the authors propose a privacy-aware framework that is tied in with HiTLIoT systems. This framework helps in sharing data between the user and the IoT system while preserving the user's privacy. DAARC is very different from this framework as DAARC focuses on securing the system in case any threat is present in the system, while the framework in [14] focuses on privacy-preserving data transfers between the user and the systems.

Paper [15] only uses human intelligence in the form of an artificial immune system with a deep neural network in the detection part of the framework. After that, humans are considered a security threat due to their lack of training and effectiveness. On the other hand, DAARC uses humans in the analysis phase, and human intelligence is always considered a mechanism to bolster security and not hinder it. DAARC uses the human intellect to combat the intrusion, not detect it, unlike the Blacksite framework proposed in [15].

The approach in [16] talks about a detection technique, VindiCo, based on the mutual information got from context-based decisions of the user. DAARC uses the data from the user not to detect the intrusion but to find the most effective way of dealing with it. The data from the user prove helpful for the human analyst to combat the intrusion effectively. It is important to note that the Blacksite framework [15] and VindiCo framework [16] are some of the methods that can be used by system designers in the Detection phase of the proposed DAARC framework.

## 4 Conclusion and Future Work

IoT systems and Cyber-Physical Systems make use of the increasing connectedness of the world through the Internet. Along with this connectivity comes a plethora of security and privacy threats that plague such systems. Moreover, since these systems involve interaction with human beings, the points of failure increase exponentially. Addressing human-in-the-loop control and associated security issues becomes imperative as human users are ultimately the victims of any malicious attack on HiTLCPS/HiTLIoT systems. In order to address the security concerns, this chapter presents a five-stage security framework (DAARC) to mitigate any threat that attacks the system. It makes use of a human-in-the-loop expert to make the final call on how dangerous the threat is and what mitigation measures are to be employed to secure the system. It also presents a use case scenario where the framework can find its use.

The DAARC framework is a novel framework because it uses humans to bolster security rather than considering them as a point of concern. Nevertheless, there are a few areas of concern in our framework. Due to human Analyst being an essential part of the security framework, we need the Analyst to be knowledgeable, experienced and well dedicated to the task. Also, the Analyst needs to respond quickly so that the intrusion is stopped as quickly as possible. Apart from that, if

malware that can alter the logs is used, it will send wrong information to the Analyst resulting in false judgement by the Analyst. It is one of the ways by which the human nature of the Analyst can be exploited. A machine learning model could be developed that can use the logs and bug reports generated by the Analyst to classify the type of intrusion and develop a method to resolve it as quickly as possible. Incorporating an adaptive framework that can differentiate between falsely injected data and original data is something that can also be worked on in the future.

## References

1. M. Chayko, *Techno-social life: The Internet, digital technology, and social connect- edness* (2014). <http://arxiv.org/abs/https://compass.onlinelibrary.wiley.com/doi/pdf/10.1111/soc4.12190>, <https://doi.org/10.1111/soc4.12190>.
2. R. Minerva, A. Binu, D. Rotondi, *Towards a definition of the Internet of Things (iot)*, [https://iot.ieee.org/images/files/pdf/IEEE\\_IoT\\_Towards\\_Definition\\_Internet\\_of\\_Things\\_Revision1\\_27MAY15.pdf](https://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf) (2015).
3. R. R. Rajkumar, I. Lee, L. Sha, J. Stankovic, *Cyber-physical systems: The next computing revolution* (2010). <https://doi.org/10.1145/1837274.1837461>.
4. C. Greer, M. Burns, D. Wollman, E. Griffor, *Cyber-physical systems and Internet of Things* (2019-03-07 2019). <https://doi.org/10.6028/NIST.SP.1900-202>.
5. B. M. Tehrani, J. Wang, C. Wang, *Review of human-in-the-loop cyber-physical systems (HiTLCPs): The current status from human perspective* (2019). <http://arxiv.org/abs/https://ascelibrary.org/doi/pdf/10.1061/9780784482438.060>, <https://doi.org/10.1061/9780784482438.060>.
6. D. S. Nunes, P. Zhang, J. Sá Silva, *A survey on human-in-the-loop applications towards an internet of all* (2015). <https://doi.org/10.1109/COMST.2015.2398816>.
7. Kaspersky, *The human factor in it security: How employees are making businesses vulnerable from within*, <https://www.kaspersky.com/blog/the-human-factor-in-it-security/> (2017).
8. N. Y. Kim, S. Rathore, J. H. Ryu, J. H. Park, J. H. Park, *A survey on cyber physical system security for IoT: Issues, challenges, threats, solutions* (2018). <https://doi.org/10.3745/JIPS.03.0105>.
9. Rachit, S. Bhatt, P. R. Ragiri, *Security trends in internet of things: a survey* (2021). <https://doi.org/10.1007/s42452-021-04156-9>.
10. C. Perera, M. Barhamgi, A. K. Bandara, M. Ajmal, B. Price, B. Nuseibeh, *Designing privacy-aware internet of things applications* (2020). <https://doi.org/10.1016/j.ins.2019.09.061>.
11. I. Psychoula, L. Chen, X. Yao, H. Ning, *A privacy aware architecture for IoT enabled systems* (2019). <https://doi.org/10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00073>.
12. L. F. Cranor, *A framework for reasoning about the human in the loop* (2008).
13. R. Rohan, S. Funilkul, D. Pal, H. Thapliyal, *Humans in the loop: Cybersecurity aspects in the consumer IoT context* (2021). <https://doi.org/10.1109/MCE.2021.3095385>.
14. J. E. Rivadeneira, J. Sá Silva, R. Colomo-Palacios, A. Rodrigues, J. M. Fernandes, F. Boavida, *A privacy-aware framework integration into a human-in-the-loop IoT system* (2021). <https://doi.org/10.1109/INFOCOMWKSHPS51825.2021.9484634>.
15. J. Brown, M. Anwar, *Blacksite: Human-in-the-loop artificial immune system for intrusion detection in internet of things* (03 2021). <https://doi.org/10.1007/s42454-020-00017-9>.
16. S. Elmalaki, B.-J. Ho, M. Alzantot, Y. Shoukry, M. Srivastava, *Vindico: Privacy safeguard against adaptation based spyware in human-in-the-loop IoT* (2022). <https://doi.org/10.1109/SPW.2019.00039>.

17. N. M. Karie, N. M. Sahri, P. Haskell-Dowland, IoT threat detection advances, challenges and future directions (2020). <https://doi.org/10.1109/ETSecIoT50046.2020.00009>.
18. M. Woźniak, J. Siłka, M. Wieczorek, M. Alrashoud, Recurrent neural network model for IoT and networking malware threat detection (2021). <https://doi.org/10.1109/TII.2020.3021689>.
19. F. Ullah, H. Naeem, S. Jabbar, S. Khalid, M. A. Latif, F. Al-turjman, L. Mostarda, Cyber security threats detection in internet of things using deep learning approach (2019). <https://doi.org/10.1109/ACCESS.2019.2937347>.
20. M. Fahim, A. Sillitti, Anomaly detection, analysis and prediction techniques in IoT environment: A systematic literature review (2019). <https://doi.org/10.1109/ACCESS.2019.2921912>.
21. K. Ren, T. Zheng, Z. Qin, X. Liu, Adversarial attacks and defenses in deep learning (2020). <https://doi.org/10.1016/j.eng.2019.12.012>.
22. H. A. Kholidy, Autonomous mitigation of cyber risks in the cyber–physical systems (2021). <https://doi.org/10.1016/j.future.2020.09.002>.

# Survey: Neural Network Authentication and Tampering Detection



Rahul Kumar, Ashwin P, Bhumik Naveen, and B R Chandavarkar

## 1 Introduction

Deep Learning and Machine Learning methods have been a growing trend in recent years, especially for computer vision tasks. While the industry has had successful techniques in dealing with image and video content, there were still additional challenges, such as motion, temporal consistency, and spatial information. Neural networks consist of interconnected neurons with adjoining edges [1]. With an attempt to mimic the structure of the human brain, these computational models have been able to carry out a complex set of computations in the computer vision domain by use of convolutions and pooling [2, 3]. The application of neural networks in cryptography is numerous [4, 5]. Particularly, for encryption, chaos-based neural networks have been proposed [6, 7].

Neural networks are suitable for hashing or encryption because of their one-way property i.e., making it computationally infeasible to generate the input from the output of the neural network [6, 8]. Neural networks are parameter sensitive, i.e., even when small changes are made to the parameters of the neural network, the output changes greatly. This is also suitable for encryption because when we make small changes to the key, we expect the ciphertext to change significantly [9].

The rise in digital content distribution and image processing has led to an increase in image duplication, reproduction, and re-distribution at a low cost. Coupled with the rapid development in network technologies, this has posed a significant threat to the privacy and security of data. To this effect, content authentication and protection

---

R. Kumar · Ashwin P (✉) · B. Naveen · B R Chandavarkar  
National Institute of Technology Karnataka Surathkal, Computer Science & Engineering,  
Mangalore, Karnataka, India  
e-mail: [rahulkr.191cs239@nitk.edu.in](mailto:rahulkr.191cs239@nitk.edu.in); [ashwinp.191cs213@nitk.edu.in](mailto:ashwinp.191cs213@nitk.edu.in);  
[bhumiknaveen.191cs216@nitk.edu.in](mailto:bhumiknaveen.191cs216@nitk.edu.in)

against tampering has been a significant challenge to solve. According to a report in 2015 [10], nearly 1.8 billion images are uploaded daily. Besides content distribution, digital images play a significant role in medical, journalism, scientific publications, etc. These images can easily be manipulated to hide critical information or create misleading images. Tools like Photoshop make these manipulations extremely difficult to detect.

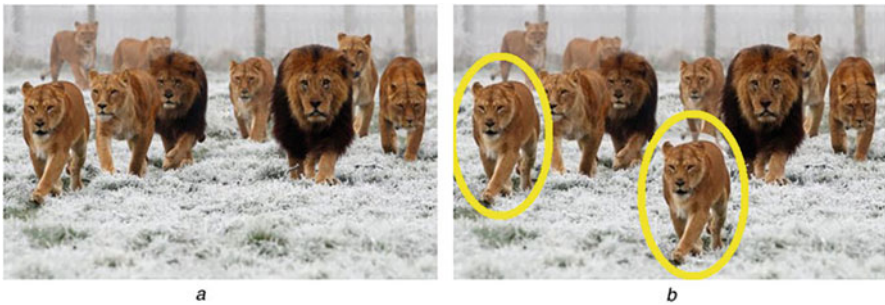
Traditional tampering detection approaches have only been able to detect specific types of tampering, but neural networks with the use of deep learning methods can detect different types of tampering at the same time by observing and extracting complex features from the image. In this study, we review some image forgery techniques and look over how neural networks are being used to detect forgery and authenticate images.

The study is organized as follows. In Sect. 2, we list commonly used methods to tamper images. In Sect. 3, we review several approaches to image tampering and authentication which involve neural networks. Section 4 concludes the chapter.

## 2 Image Tampering: Overview

With the rapid growth in technology across the world, many advanced types of image tampering techniques have been developed, making it much harder to authenticate images. Some of the most popular and effective image tampering techniques have been discussed in the following subsections

- **Copy-Move:** This is the most common technique which is used to tamper images. In this technique, a part of the image is copied, and it is pasted are some other parts of the same image. This technique is usually used to hide or conceal some part of the image, as seen in Fig. 1. This is very easy with so many image-editing applications available in the world right now. This is almost impossible for the human eye to detect this forgery [11]
- **Image splicing:** In this technique, a part of a particular image is copied, and then it is pasted into some part of another image, as seen in Fig. 2. This technique is



**Fig. 1** Example of copy-move tampering





Fig. 2 Example of image splicing tampering



Fig. 3 Example of image resize tampering

much harder to detect compared to copy-move because it is much harder to detect forgery when it is done using the attributes of a different image [12].

- **Resize:** This technique [13] is a bit complicated in which the size or part of the image is enlarged or shrunk by using geometrical transformations. This is done by altering or modifying the pixel values in the image as seen in Fig. 3.
- **Image Retouching:** This technique is used to modify or enhance some contents of the image by using image editing software. This is usually used to beautify some contents of the image without losing its original characteristics, as seen in Fig. 4 [14].
- **Cropping:** This technique is used to remove the borders of an image which are usually not important for the display of the image [15] (Fig. 5).



Fig. 4 Example of image retouching tampering



Fig. 5 Example of image cropping tampering

- **Noising or Blurring:** The tampering techniques mentioned so far might leave traces sometimes, which makes it easier to detect it, to resolve this; usually some noise or blur [16] operations are applied on the tampered sections of the image, making it almost undetectable as seen in Fig. 6.

### 3 Neural Network Role in Authentication and Tampering

Tampering detection is broadly classified into Active Detection and Passive Detection. In the active approach, information is embedded into the image by an authorized entity which can be later extracted to authenticate the image and to detect tampering, if any. Digital Watermarking [17] is one such approach. The passive approach, also known as the Blind approach, involves extracting features to detect tampering [18]. A high-level breakdown of image tampering detection techniques is



Fig. 6 Example of image noising tampering

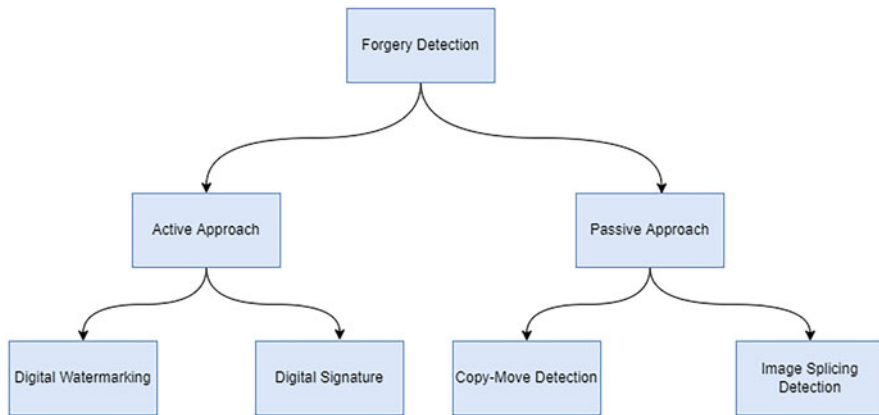


Fig. 7 Breakdown of tampering detection approaches

illustrated in Fig. 7. As such, in this section, we review several proposed approaches involving the application of neural networks in Active Detection and Passive Detection.

### 3.1 Digital Watermarking

Watermarking is the process of embedding information into data objects such as images, text, audio, or video, which is later extracted to authenticate the data object. They are broadly classified into visible watermarks and invisible watermarks. Visible watermarks, as the name suggests, are visible to the human eye and are essentially logos that are displayed at the corner of images or videos. We will limit further discussion of watermarking to invisible watermarking.

### 3.1.1 Watermarking Techniques

When dealing with digital image watermarking, we come across the following two approaches:

- Spatial Domain Watermarking
- Frequency Domain Watermarking

In **Spatial Domain Watermarking**, the pixel values of the host image are directly modified using owner-authorized data, such as a logo. The least significant bit (LSB) modification, significant intermediate bit (ISB) modification, and patchwork algorithms are some commonly used algorithms used in spatial domain watermarking.

Under the LSB modification algorithm, the most significant bit (MSB) of the watermark data is used to replace the LSB of the host/original image, thus modifying it. Chopra et al. [19] is an example of an approach that uses LSB modification by replacing the host image bits with the logo without degradation in the host image quality. Chan and Cheng [20] is a data hiding scheme that uses LSB modification.

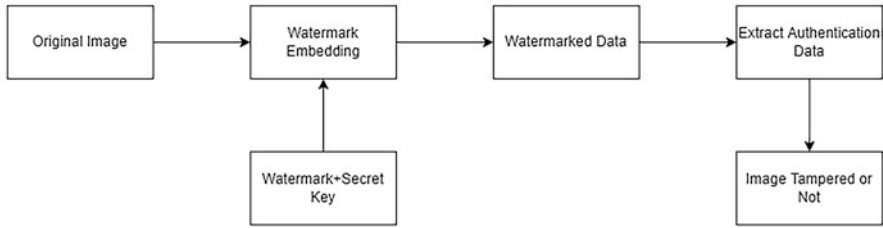
ISB modification technique is an alternative to LSB modification which chooses the best pixel value between the middle and edge of the pixel range and replaces it with the watermark data, similar to LSB modification. Zeki and Manaf [21] discusses the robustness of image watermarking with respect to attacks like blurring, compression, scaling, cropping and filtering.

Lastly, patchwork algorithms employ pattern-based embedding of the watermark data onto the host image. In [22], an additive-multiplicative patchwork algorithm is used, which was found to be robust against compression-based attacks.

In **Frequency Domain Watermarking**, the frequency of the input image is modified to hide the watermark in the image. This technique converts the image to represent it into the frequency domain and the watermarks are embedded using transformation techniques, namely Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT) and Singular Value Decomposition, among others. Finally, an inverse transformation is applied to extract the watermark, using a correct key.

Discrete Cosine Transform involves representing the image as its frequency transformation and expressing it as the sum of cosines of the frequency components. This method is widely used in image compression algorithms [23]. Hernandez et al. [24] uses DCT based watermarking for copyright protection in still images. The input image is represented as blocks, and each block is transformed through DCT. The approach was found to achieve robustness against geometric attacks, and noise-based attacks [25].

Discrete Fourier Transform takes a finite list of equally-spaced samples as input and converts them into a list of harmonically related exponential functions. In other words, DFT converts the input image in the frequency domain to discrete and periodic signals. The use of DFT for image watermarking is shown in [26] where spatial-chromatic discrete Fourier transform was used to embed a yellow-and-blue watermark. The study is able to demonstrate improved robustness against geometric attacks. DFT based watermarking has also been used in [27] for managing medical



**Fig. 8** Watermarking-based image tampering detection

images where the watermark being used are the patient's medical records which are encrypted and embedded onto the host image.

### 3.1.2 Watermarking-Based Tampering Detection

Figure 8 gives an overview of image tampering detection using image watermarking [28]. The tampering is detected by using the embedded watermark and the secret to authenticate the image at the sender side. Any significant change in the image through tampering would change the embedded watermark. Detecting these changes in the watermark would help in deciding the authenticity of the image. Tohidi et al. [29] proposed an image integrity verification technique through self-embedding of the watermark in the image and was successful in detecting and localizing the tampered region in the image without degrading the image quality.

### 3.1.3 Neural Network-Aided Image Watermarking

When it comes to neural networks, they find their use in the embedding and extraction process of watermarking. As such, multiple solutions [30–34] have been proposed to make use of neural networks in these stages.

#### Non-blind Watermarking

The work [30] is one of the earliest applications of deep learning in digital watermarking. The proposed method generates codebooks from the input image during the embedding stage, which is later used in the extraction. The host image (PosImg) is separated into its inverted image (NegImg), which is generated by subtracting one from the normalized host image. Following preprocessing involving downsampling, the PosImg and NegImg are processed through pretrained CNN AutoEncoders to generate respective codebook images. The codebook generation is shown in Fig. 9.

As shown in Fig. 10, the embedding process is as follows. The codebook images and the watermark images are randomly permuted using keys K1 and K2. The watermarked image is generated by choosing the pixel group from the PosImg

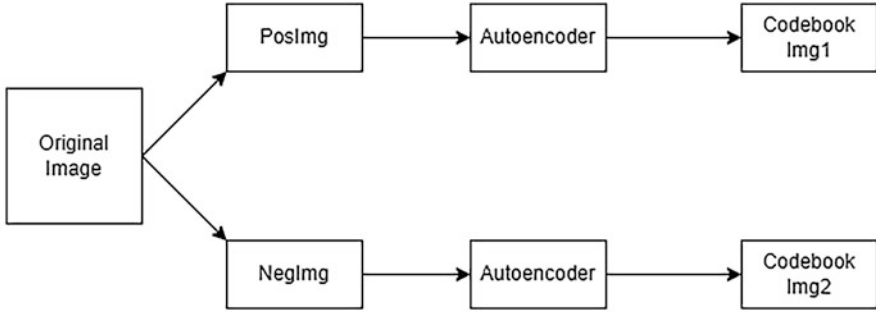


Fig. 9 Codebook generation

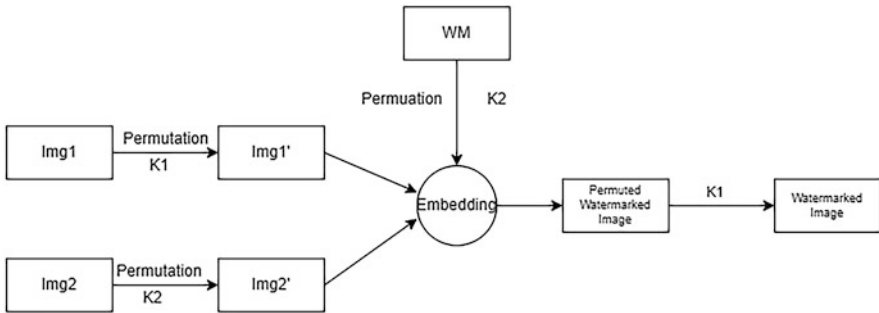


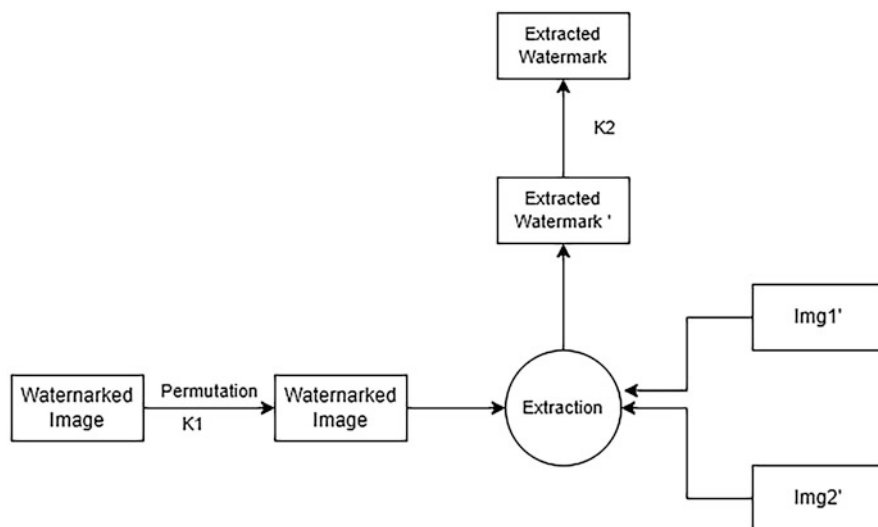
Fig. 10 Watermark embedding

codebook whenever the WM bit is 1; else, the pixel groups are chosen from the NegImg codebook. During extraction, the watermarked image is permuted through the use of key K1. The codebook vectors are regenerated from the permuted image, and the watermark image is extracted from the codebook through the use of Euclidean distance. The extraction process is summarized in Fig. 11.

### Blind Watermarking

Zhu et al. [31] propose a method named “HiDDeN,” which makes use of a combination of a Watermark Embedding Network, Watermark Extraction Network alongside a noise layer and an adversarial network, the latter of which is used to aid in watermarking. Before the embedding process, the WM data are replicated to the image resolution after being flattened to one-dimensional data. This is then concatenated with the host image and fed to the embedding network. The embedding network consists of convolution and batch normalization layers. For the extraction process, the embedded and watermarked data are fed to the WM extraction network as well as the adversary network. The embedded data are fed to the extraction layer by passing through the noise layer to introduce random noise. Finally, the extraction network downsamples the image and processes the final output through a fully connected layer [32].

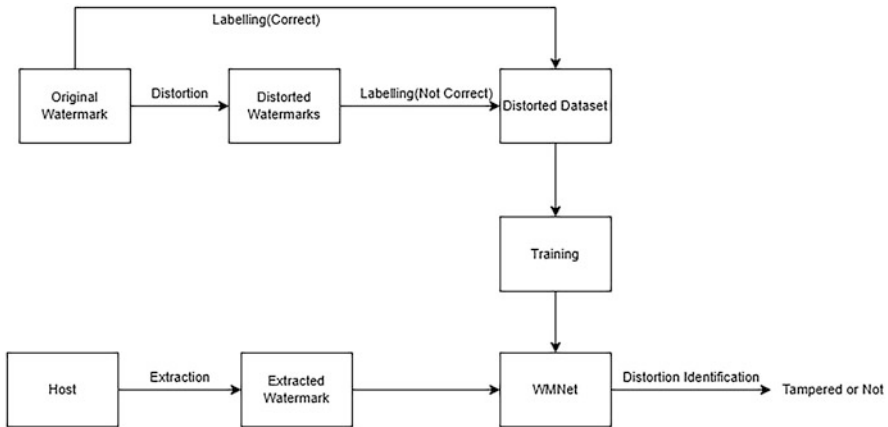




**Fig. 11** Watermark extraction

Ahmadi et al. [32] is one of the few methods which combines the use of frequency-domain watermarking alongside a neural network. The authors propose a network consisting of embedding and extraction networks with an attack simulation network to aid in robustness and training. The images are downsampled and transformed through Discrete Cosine Transform using a pretrained DCT network. The WM data are scaled and concatenated with the transformed host image, which is then fed to the embedding network, which is a Convolutional Neural Network consisting of circular convolutions and ELU activation functions. The circular convolution layers are used for upsampling the resolution of the image. The final layer is an inverse DCT layer that generates the watermarked image. The WM extracting resembles the embedding network and also consists of a DCT layer, an inverse DCT layer and a CNN with circular convolution layers. The result of this is output with the resolution as that of the WM data. The authors conclude by proposing a scheme with a DCT layer within the network instead of it being a preprocessing step.

Mun et al. [33] propose the use of AutoEncoder-based neural network framework consisting of residual blocks. These residual blocks are composed of units consisting of multiple convolution-Relu layers. In the embedding process, the original/host image is downsampled in resolution through the encoder in the network. This encoded datum is formed by adding the watermarking information in each of the layers, which is then fed to the decoder of the embedding network. The main function of the decoder is to upsample the encoded data to the resolution of the host image and reduce the number of channels to that of the host image. The accumulation of the WM information from the decoder forms the watermarked



**Fig. 12** WMNet design

image. The structure of the extraction network is similar, with the exception that it drops the pooling layers. The attacks are simulated in each mini-batch of the training.

### Watermark Verification

In [35], the authors propose a watermarking verification scheme, namely WMNet, that is a watermark verification model. The proposed model has been illustrated in Fig. 12. An attacked host image would have a distorted watermark which would help in determining the authenticity of the image. WMNet is a CNN model trained on distorted watermark images generated through simulation. The dataset being used consists of the original watermark along with the simulated distorted watermark. The attacks were simulated through JPEG Compression, Gaussian Noise Addition, median filtering, and cropping and were found robust against such attacks and with an objective determination of tampering and authenticity. Being a watermark verification model, it can be coupled with other watermarking extraction and embedding techniques.

## 3.2 Image Authentication Based on Neural Networks

In this model, [36] Fig. 13, the authenticity of images is verified by comparing the authentication codes of the sent and received images. If there is a significant difference between the authentication codes, then the image has been tampered with. This model utilizes the one-way property and learning ability of the neural network.

The original image data, original authentication code, and a key that generates a pseudo-random sequence [37] are given as an input to the neural network [1] by the sender. The neural network, in turn, gives a secret parameter which, along with



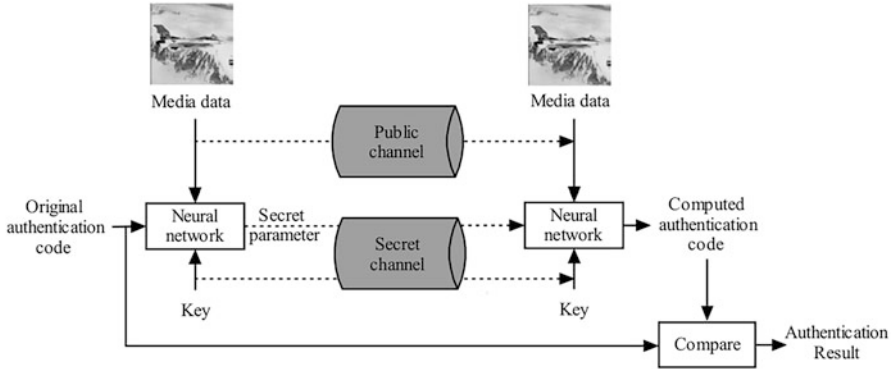


Fig. 13 The proposed architecture [36]

the key, is sent to the receiver in a secure channel, i.e., some form of the encryption algorithm (DES, AES, etc.) [38] is used to encrypt the secret parameter and key so that it has not tampered when it reaches the receiver, whereas the image data are sent or distributed freely across the network where it can be tampered with maliciously.

Now, the receiver will receive the secret parameter, key (sent in a secure channel), and the image data (sent in a public channel). Using the key, the receiver also generates a pseudorandom sequence which, along with the secret parameter and the image data, is given to the neural network as an input, and the output given is a new authentication code. The original authentication code that the sender generates and the new authentication code are compared to see if the image has been tampered with or not.

**Generation of Secret Parameter** On the sender side, the image data are divided into blocks, and each bit of the authentication code(s) is associated with these blocks, and the neural network generates a secret parameter (b) for each of these blocks.

$$b = \begin{cases} > -\Sigma w_j p_j, & \text{if } s = 1. \\ <= -\Sigma w_j p_j, & \text{if } s = 0. \end{cases} \quad (1)$$

$$b = \begin{cases} > T - \Sigma w_j p_j, & \text{if } s = 1. \\ <= -T - \Sigma w_j p_j, & \text{if } s = 0. \end{cases} \quad (2)$$

where  $p_1, p_2, \dots, p_n$  are the image pixels of an image block and  $w_1, w_2, \dots, w_n$  is the pseudorandom sequence generated using the key and  $s$  is one of the bits of the authentication code which is associated with this image block. Here,  $T$  is adjusted and set by doing many test runs and experiments.  $T$  is useful when we are comparing

the authentication codes on the receiver side. Now, the secret parameter and key are sent using the secret channel to the receiver.

**Generation of New Authentication Code** Now, the receiver generates the new authentication code using the secret parameter  $b$ , key, and the image data received through the public channel. Similar to the sender's side, the image is divided into blocks.

If the image data shared in the public channel are not altered, then it is easy to get the authentication code

$$s^1 = \begin{cases} 1, & \text{if } T > 0. \\ 0, & \text{if } -T \leq 0. \end{cases} \quad (3)$$

But if the image data are changed or tampered with, then we get the new authentication code using the following method:

$$s^1 = \begin{cases} 1, & \text{if } T + \sum w_j p_j^1 - \sum w_j p_j > 0. \\ 0, & \text{if } -T + \sum w_j p_j^1 - \sum w_j p_j \leq 0. \end{cases} \quad (4)$$

$p_1^1, p_2^1, \dots, p_n^1$  are the pixels of the image block, and  $w_1, w_2, \dots, w_n$  is the pseudorandom sequence generated using the key similar to the sender's side.  $s^1$  is generated for each image block.

The new authentication code is generated by the change; if this change is greater than  $T$ , then we will get the wrong code which means that the image has been tampered with.

### Advantages

- **Time efficiency:** This model was tested with different images of varying sizes to check its efficiency. The computer used to test it had the following specifications: 1.7 GHz CPU and 256 GB RAM. The computing and authentication time for all different types of images are less than 1 s, making this model time efficient.
- **Security:** The secret parameter and the key are encrypted using DES, RSA, AES, etc. Therefore, without having the above items, the hacker would have to guess the parameters used in the neural network, which is very difficult, making this model very secure.
- **Robustness:** The model was tested by applying Gaussian noise [39] to the image, and even then, the tampering detection rate was more significant than 80%. Compressed images [40] were passed as input to the model, and the model still had a detection rate greater than 85%.
- The model is not specific to detecting only one type of tampering technique; it can detect any form of tampering because it works on the pixel values of the image.

### Disadvantages

- The model cannot predict what image tampering technique was used on the image.

## 3.3 *Image Tampering Detection Based on Neural Networks*

With the advancements in the field of Neural Networks and Deep Learning, Neural Networks provide us with techniques for image tampering detection, which are efficient and better. We will go through some of these techniques based on Neural Networks.

### 3.3.1 **Forgery Detection Based on DCT-CNN**

This method [41, 42] divides the suspicious image into overlapping square blocks and gets the features from these overlapping blocks. After extracting the features of each such block, the feature vectors are sorted lexicographically, and blocks of the same regions are matched. To obtain high success rates, the centers of the matched block pairs are calculated instead of marking areas of the matched blocks, and these are considered reference points in both original and forged regions. Thereafter, with the use of these reference points, the two clones of the suspicious image are compared with overlapping and thus differentiated, and the intersection regions of the lowest differentials are considered pairs of the forged region.

This technique [41] uses Discrete Cosine Transform (DCT), which is then followed by the implementation of the Convolutional Neural Network (CNN) for the detection of copy–move forgery. It is an image forgery (copy–move) detection technique that handles copy–move forgeries concurrently.

1. In the first stage, the transformation of the input image into greyscale occurs.
2. Next, DCT (Discrete Cosine Transform) is used as a pre-processing step [43].
3. Once the filtered features are integrated, the model is trained using original and tampered images.
4. Then, the trained CNN [44–47] is used to detect whether there is tampering in the image or not.

Discrete Cosine Transform [48] operates in the frequency domain. It shows a set of points (data) about the summation of cosine functions having different frequencies. DCT has various applications such as lossy audio compression like MP3 and images such as JPEG and image compression and many more. Let  $L(x, y) \in R, x, y \in [0, N]$  a small part of image of the  $y$  component of original image.

1. Discrete Cosine Transform coefficients table is defined as:  $DCT(a, b) \in \mathbb{R}$ ,  $a, b \in [0, M)$ ,  $M = 8$  by the equation:  $DCT(a, b) = \begin{cases} \frac{1}{\sqrt{M}}, & a = 0, \forall b, \\ \frac{1}{2} \cdot \cos\left(\frac{a \cdot (2b+1) \cdot \pi}{2M}\right), & \text{otherwise.} \end{cases}$

2. Block wise Discrete Cosine Transform coefficients table for every  $L_{part}(x_a, y_b) \in L$ , where  $x_a \in [a \cdot M, a \cdot (M + 1))$ ,  $y_b \in [b \cdot M, b \cdot (M + 1))$ ,  $a, b \in [0, \lfloor \frac{N}{M} \rfloor - 1]$ :

$$U_{part} = DCT \cdot L_{part} \cdot DCT^T$$

3. Compute  $M^2$  histograms along with  $K$  bins of the output  $U$  image that refers to the  $L$  part to all the applied Discrete Cosine Transform coefficient.

Thus, we got the feature matrix  $f_L \in \mathbb{R}^{M^2 \times K}$

Suppose we have a tampered image with an unknown size JPEG compressed area and the location and shape of the tampering are also unknown. A significant problem with discovering this area is an oppressed area like this is not straightforward. Therefore, we cannot directly calculate the features with the help of a sliding window to obtain the location of the compressed region. So, We must have to detect the shift of the JPEG block and then utilize the Discrete Cosine Transform(DCT) algorithm mentioned above to locate the forged region.

### Advantages

- The use of DCT for tampering detection is better for JPEG images than using the PCA (Principle Component Analysis) method [49, 50] since PCA does not detect jpeg image tampering properly.
- The DCT approach has low calculation costs [51, 52]. Performing better than the method proposed by A. Kuznetsov and N. Glumov [53] for JPEG shift detection as that method failed to deliver effective results with respect to calculational complexity.
- The quality/efficiency of this method [41, 42] on JPEG compression artifacts is really high. It is better than the existing JPEG algorithms for single JPEG detection. However, it needs further investigation to detect different quality features.
- The lower length of feature vectors gets utilized effectively.
- One great advantage is its robustness against the post-processing operations on the tampered area.

### Disadvantages

- DCT based techniques [41, 42] do not perform well in the case of blurring of images and video frame reconstruction applications.

### 3.3.2 Copy–Move Forgery Detection Based on SIFT-PCA: (Scale Invariant Feature Transform—Principal Component Analysis)

The SIFT (Scale Invariant Feature Transform) algorithm [54] extracts features from the image and extracts blocks using PCA (Principal component Analysis) [55]. SIFT features are selected to set the integration phase from the beginning to the end of the relational map. But the dataset is not specified. Hence, it is not considered for more than one copy–move forgery. For more than one forgery, examining the same SIFT method and combining key points to differentiate an integrated component are considered. However, the procedure is based solely on the disclosure of cloning and not for extract alteration identification.

A SIFT algorithm [54] has been used that can detect CMF (copy–move forgery) made in the image and measure the conversion parameters used. It is able to find a collection of points for cloned regions. The image is passed to the background of the feature and the matching where the key points get drawn and then compared with the original image. Subsequently, clusters are formed using cluster collections. Additionally, the forged regions get identified.

PCA is one of the processes [55] of taking high-dimension information and using situations between variables to communicate with it in a flexible, low-cost format, without losing much data. PCA is a very powerful way to reduce such dimensions while being very least complex.

#### Advantages

- It can produce many features that overlap the image at full distance and locations. For example, it is possible to collect 2000 features in a standard image(500×500 pixels).
- Better rates of recall (accuracy) and it works well compared to older algorithms [56].
- Features are robust to occlusion and clutter.

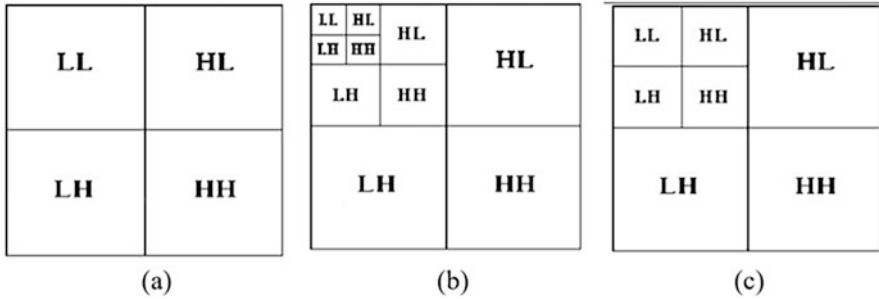
#### Disadvantages

- It is still slow. Speeded Up Robust Features(SURF) [57] performs similarly while being efficient.
- It usually does not work well with changes in lighting and dimming.
- SIFT accuracy is resistant to modern algorithms, but it is not the best option for real-time applications as it is computationally expensive.

### 3.3.3 Forgery Detection Based on DWT (Discrete Wavelet Transforms)

Discrete wavelet transforms [58] is a wavelet Transform when the wavelets are discretely sampled. DWT measurement is used to compress data when the signal is already sampled. It is an effective way to compensate for lost weights.

Wavelet conversion is used to signal into a set of basic functions referred as wavelets. A single prototype called the mother wavelet is considered in which wave



**Fig. 14** The image in DWT decomposition: (a) single level, (b) third level, and (c) second level

waves are obtained through expansion and flexibility.

$$\varphi_{x,y} = \frac{1}{\sqrt{x}}\varphi\left(\frac{s-y}{x}\right)$$

Here  $x$  refers to the scaling parameter whereas  $y$  refers to the shifting parameter. Wavelet transform for one dimension can be calculated as:

$$W_{f(x,y)} = \int_{-\infty}^{\infty} y(s)\varphi(s)ds$$

In the case of 2D Discrete Wavelet Transform [59], the image is decomposed into four parts at every level. These smaller parts are called the sub-bands. The smaller parts thus obtained are named LL, HL, LH and HH. The DWT decomposition for a single level, second level, the third level is shown in Fig. 14

In this Discrete Wavelet Transform Principal Component Analysis (DWT-PCA) algorithm [60], initially, DWT is used to extract the input image to four smaller bands. From these four bands, a band is selected, which is called the approximation band for additional processing. The selected approximation band is then divided into fixed-size blocks.

The element is removed from the scattered blocks and then kept in a matrix. The vector size of the feature is reduced by using PCA. In the case of PCA, only the target value of the feature vector is used so that the required information doesn't get lost while at the same time no unnecessary data to be a feature that enhances the vector feature. After all this, a reduced vector element matrix is filtered lexicographically. Finally, the shift vectors are computed and usually see the blocks in the picture with the same shift.

**Advantages**

- This method is robust on JPEG compression Quality level 70.
- This technique is very effective against several manipulations made over the copied segment before attaching it.

- This method can also detect regions rotated by various degrees before attaching them to create a fake image.
- The use of radix sort instead of lexicographically sorting. The lexicographical sorting has a higher complexity than the radix. This method has less time complexity.

### Disadvantages

- Cost of computing is more in case of DWT than in PCA [49].
- When we compare it to the other methods mentioned above, Discrete wavelet transforms has some major drawbacks. Discrete wavelet transforms also has higher computational costs than Principal Component Analysis. Discrete wavelet transforms show many limitations in contrast to Discrete Cosine Transform [61], such as ringing noise near edge edges in images or video frames, signal blurring, lower quality than JPEG with lower compression rates, longer pressing time, etc.
- This method [59] is sensitive to shifting performed over forged areas.
- This method cannot identify forged copy-move regions when the size of the block used for dividing the image is more than the forged area dimension.

## 4 Conclusion and Future Works

This study covered several applications of deep learning and neural networks for tampering detection in images and proposed methods for image authentication. From this survey, we found that the application of the neural network for tampering detection depends on the tampering detection approach. For active detection, namely, digital watermarking, neural networks are primarily being used to aid the embedding and extraction through a wide variety of architectures such as Convolutional Neural Networks and AutoEncoders. Furthermore, approaches exist that use neural networks for the watermark verification phase in digital watermarking. For passive detection, neural network architectures like Convolutional Neural Networks are feature extractors to detect images tampered with through techniques like copy-move and image splicing. Beyond this, methods have been proposed to use neural networks to generate authentication codes for image authentication.

As a result of this study, we concluded that neural network architectures used in tampering detection vary depending on the tampering detection approach. Both these approaches showed robustness. A hybrid model can be implemented that uses an embedding extraction network for watermarking alongside a CNN-based feature extractor model as a passive tampering detector.

## References

1. J. A. Anderson, An introduction to neural networks, MIT press, 1995.
2. N. Aloysius, M. Geetha, A review on deep convolutional neural networks, in: 2017 international conference on communication and signal processing (ICCSPP), IEEE, 2017, pp. 0588–0592.
3. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.
4. P. P. Hadke, S. G. Kale, Use of neural networks in cryptography: a review, in: 2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave), IEEE, 2016, pp. 1–4.
5. K. Shihab, A backpropagation neural network for computer network security, Journal of Computer Science 2 (9) (2006) 710–715.
6. S. Lian, Z. Liu, Z. Ren, H. Wang, Hash function based on chaotic neural networks, in: 2006 IEEE International Symposium on Circuits and Systems, IEEE, 2006, pp. 4–pp.
7. W. Yu, J. Cao, Cryptography based on delayed chaotic neural networks, Physics Letters A 356 (4–5) (2006) 333–338.
8. Z. W. Shiguo Lian, Jinsheng Sun, One-way hash function based on neural network, in: Department of Automation, Nanjing University of Science and Technology, ARXIV, 2007, pp. 2–5.
9. S. Lian, Image authentication based on neural networks, in: SAMI Lab, France Telecom RD Beijing, ARXIV, 2007, pp. 2–4.
10. R. Eveleth, [How many photographs of you are out there in the world?](https://www.theatlantic.com/technology/archive/2015/11/how-many-photographs-of-you-are-out-there-in-the-world/413389/), The Atlantic. URL <https://www.theatlantic.com/technology/archive/2015/11/how-many-photographs-of-you-are-out-there-in-the-world/413389/>
11. W.-K. C. Jen-Chun Lee, Chien-Ping Chang, Detection of copy–move image forgery using histogram of orientated gradients, Information Sciences 321 (2015) 250–262.
12. B. Xu, G. Liu, Y. Dai, Detecting image splicing using merged features in Chroma space, The Scientific World Journal 2014.
13. D. Zhangm, S. Wang, J. Wang, A. K. Sangaiah, F. Li, V. S. Sheng, Detection of tampering by image resizing using local Tchebichef moments, Applied Sciences 9.
14. V. Savchenko, N. Kojekine, H. Unno, A practical image retouching method, in: First International Symposium on Cyber Worlds, IEEE, 2002, pp. 2–4.
15. W. Luo, Z. Qu, J. Huang, G. Qiu, A novel method for detecting cropped and recompressed image block, in: IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07, IEEE, 2007, pp. 2–4.
16. P. Patidar, M. Gupta, S. Srivastava, A. K. Nagawat, Image de-noising by various filters for different noise, International Journal of Computer Applications 9 (4).
17. C. I. Podilchuk, E. J. Delp, Digital watermarking: algorithms and applications, IEEE signal processing Magazine 18 (4) (2001) 33–46.
18. G. K. Birajdar, V. H. Mankar, Digital image forgery detection using passive techniques: A survey, Digital investigation 10 (3) (2013) 226–245.
19. D. Chopra, P. Gupta, G. Sanjay, A. Gupta, LSB based digital image watermarking for gray scale image, IOSR journal of Computer Engineering 6 (1) (2012) 36–41.
20. C.-K. Chan, L.-M. Cheng, Hiding data in images by simple LSB substitution, Pattern recognition 37 (3) (2004) 469–474.
21. A. M. Zeki, A. A. Manaf, A novel digital watermarking technique based on ISB (intermediate significant bit), World Academy of Science, Engineering and Technology 50 (2009) 989–996.
22. I.-K. Yeo, H. J. Kim, Generalized patchwork algorithm for image watermarking, Multimedia systems 9 (3) (2003) 261–265.
23. P. Telagarapu, V. J. Naveen, A. L. Prasanthi, G. V. Santhi, Image compression using DCT and wavelet transformations, International Journal of Signal Processing, Image Processing and Pattern Recognition 4 (3) (2011) 61–74.



24. J. R. Hernandez, M. Amado, F. Perez-Gonzalez, Dct-domain watermarking techniques for still images: Detector performance analysis and a new structure, *IEEE transactions on image processing* 9 (1) (2000) 55–68.
25. C. Song, S. Sudirman, M. Merabti, D. Llewellyn-Jones, Analysis of digital image watermark attacks, in: 2010 7th IEEE Consumer Communications and Networking Conference, IEEE, 2010, pp. 1–5.
26. T. K. Tsui, X.-P. Zhang, D. Androutsos, Color image watermarking using multidimensional Fourier transforms, *IEEE Transactions on Information Forensics and security* 3 (1) (2008) 16–28.
27. M. Cedillo-Hernandez, F. Garcia-Ugalde, M. Nakano-Miyatake, H. Perez-Meana, Robust watermarking method in DFT domain for effective management of medical imaging, *Signal, Image and Video Processing* 9 (5) (2015) 1163–1178.
28. W. W. Adnan, S. Hitam, S. Abdul-Karim, M. Tamjis, A review of image watermarking, in: *Proceedings. Student Conference on Research and Development, 2003. SCORED 2003.*, IEEE, 2003, pp. 381–384.
29. F. Tohidi, M. Paul, M. R. Hooshmandasl, T. Debnath, H. Jamshidi, Efficient self-embedding data hiding for image integrity verification with pixel-wise recovery capability, in: *Pacific-Rim Symposium on Image and Video Technology*, Springer, 2019, pp. 128–141.
30. H. Kandi, D. Mishra, S. R. S. Gorthi, Exploring the learning capabilities of convolutional neural networks for robust image watermarking, *Computers & Security* 65 (2017) 247–268.
31. J. Zhu, R. Kaplan, J. Johnson, L. Fei-Fei, Hidden: Hiding data with deep networks, in: *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 657–672.
32. M. Ahmadi, A. Norouzi, N. Karimi, S. Samavi, A. Emami, Redmark: Framework for residual diffusion watermarking based on deep networks, *Expert Systems with Applications* 146 (2020) 113157.
33. S.-M. Mun, S.-H. Nam, H. Jang, D. Kim, H.-K. Lee, Finding robust domain from attacks: A learning framework for blind watermarking, *Neurocomputing* 337 (2019) 191–202.
34. Y. Liu, M. Guo, J. Zhang, Y. Zhu, X. Xie, A novel two-stage separable deep learning framework for practical blind watermarking, in: *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1509–1517.
35. Y.-P. Chen, T.-Y. Fan, H.-C. Chao, WMNet: A lossless watermarking technique using deep learning for medical image authentication, *Electronics* 10 (8) (2021) 932.
36. S. Lian, Image authentication based on neural networks, in: *SAMI Lab, France Telecom R<sup>1</sup>&D Beijing*, ARXIV, 2007, pp. 2–4.
37. N. S. F.J. MacWilliams, Pseudo-random sequences and arrays, *Proceedings of the IEEE* 64 (12) (1976) 1715–1729.
38. A. G. Bawna Bhat, Abdul Wahid Ali, DES and AES performance evaluation, in: *International Conference on Computing, Communication and Automation*, IEEE, 2015, pp. 2–4.
39. A. Foia, M. Trimeche, V. Katkovnik, K. Egiazarian, Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data, *IEEE Transactions on Image Processing* 17 (10) (2008) 1737–1754.
40. M. A.-M. M. Shneier, Exploiting the jpeg compression scheme for image retrieval, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18 (8) (1996) 849–853.
41. A. Kuznetsov, A new approach to jpeg tampering detection using convolutional neural networks, *International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*.
42. K. Taya, N. Kuroki, N. Takeda, T. Hirose, M. Numa, Detecting tampered regions in jpeg images via CNN, in: *2020 18th IEEE International New Circuits and Systems Conference (NEWCAS)*, 2020, pp. 202–205. <https://doi.org/10.1109/NEWCAS49341.2020.9159761>.
43. Y. Z. Pengpeng Yang, Rongrong Ni, Double jpeg compression detection by exploring the correlations in DCT domain, in: *Proceedings, APSIPA Annual Summit and Conference*, 2018, pp. 728–732.
44. Y. Abdalla, M. T. Iqbal, M. Shehata, Convolutional neural network for copy-move forgery detection, *Symmetry* 11 (10). <https://doi.org/10.3390/sym11101280>. URL <https://www.mdpi.com/2073-8994/11/10/1280>

45. A. Kuznetsov, Digital image forgery detection using deep learning approach, *Journal of Physics: Conference Series* 1368 (2019) 032028. <https://doi.org/10.1088/1742-6596/1368/3/032028>.
46. A. Kuznetsov, Digital image forgery detection using deep learning approach, *Journal of Physics: Conference Series* 1368 (2019) 032028. <https://doi.org/10.1088/1742-6596/1368/3/032028>.
47. M. M. P. Manjunatha. S, Deep learning-based technique for image tamper detection, in: *Intelligent Communication Technologies and Virtual Mobile Networks*, IEEE, 2021, pp. 1278–1285.
48. S. Duaa, J. Singha, H. Parthasarathya, Image forgery detection based on statistical features of block DCT coefficients, in: *Procedia Computer Science*, Elsevier, 2019-2020, pp. 370–378.
49. S. Kak, A. Alam, 59 sanna mehraj kak and m. afshar alam, -digital image tampering-a threat to security management, *International Research Journal of Advanced Engineering and Science*.
50. T. Mahmood, T. Nawaz, A. Irtaza, R. Ashraf, M. Shah, M. T. Mahmood, Copy-move forgery detection technique for forensic analysis in digital images, *Mathematical Problems in Engineering*.
51. Y.-D. Shin, Fast detection of copy-move forgery image using DCT, in: *Journal of Korea Multimedia Society*, Vol. 1, Korea Science, 2013, pp. 411–417.
52. D. Rohini.A.Mainid, Alka Khade, Image copy move forgery detection using block representing method, *International Journal of Soft Computing and Engineering (IJSCE)* 4.
53. A. Kuznetsov, A new approach to jpeg tampering detection using convolutional neural networks, in: *2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, 2019, pp. 0520–0524. <https://doi.org/10.1109/SIBIRCON48586.2019.8958453>.
54. D. W. Pooja Bhole, An image forgery detection using SIFT-PCA, *International Journal of Engineering Research and Technology* 9.
55. S. V. Ashima Gupta, Nisheeth Saxena, Detecting copy move forgery using DCT, *International Journal of Scientific and Research Publications* 3.
56. N. J. Ismail Taha Ahmed, Baraa Tareq Hammad, A comparative analysis of image copy-move forgery detection algorithms based on hand and machine-crafted features, *Indonesian Journal of Electrical Engineering and Computer Science* 22.
57. B. G. M. Akram Hatem Saber, Mohd Ayyub Khan, A survey on image forgery detection using different forensic approaches, *Advances in Science, Technology and Engineering Systems Journal* 5.
58. J. A. Nikhila Chacko1, Detection of image forgery in digital images using DCT and DWT, *Advancement in Image Processing and Pattern Recognition* 2.
59. R. K. G. Anuja Dixit, Rahul Dixit, Dct and dwt based methods for detecting copy-move image forgery: A review, *International Journal of Signal Processing, Image Processing and Pattern Recognition*.
60. J. T.Prabakar Joshua, M.Arrivukannamma, Comparison of DCT and DWT image compression, *International Journal of Computer Science and Mobile Computing* 5.
61. R.Mehala, Comparison of DCT and DWT in image compression techniques, *International Journal of Advanced Research Trends in Engineering and Technology* 3.

# Misinformation Detection Through Authentication of Content Creators



Kruthika K Sudhama, Sree Gayathri Siddamsetti, Pooja G,  
and B R Chandavarkar

## 1 Introduction

Misinformation is inaccurate information that has been purposefully prepared and circulated by innocent people, either consciously or subconsciously. Disinformation, fake news, rumour, click-baits, and spam are a few terminologies that are similar in nature to misinformation as described by Wu et al. [1]. If we look closely at these terminologies, we see very thin boundaries that differentiate them from each other [2, 3]. To identify the scope and limit of the challenge, you will need a clear definition, any form of inaccurate information is treated as misinformation in this work. A few advanced methods of video and image processing domains like Deepfake [4, 5] and face2face [6], face swap, and neural textures [7] pose a serious impact on the legitimacy of news on social media. Deepfake and face2face techniques can also be used to stir social or religious disputes, mislead people and sway election results, or broadcast wrong information to cause financial sector destabilization. It has been a research interest to identify the characteristics of tampered images and videos, detecting them and classifying the fake and real content.

Due to the growth of fake news and a shortage of content verification procedures, determining the reliability of information on social media is extremely difficult [8]. Firstly, original news often goes unnoticed as the fake content is a small part of the whole content the user posted [1]. Due to its sudden spread, the impact of the fake news shows up before even verifying the news leading to chaos and financial losses [9]. The third challenge is the need for a decentralized solution for detecting fake news [10]. The fourth challenge is finding authorized content from a verified user

---

K. K Sudhama (✉) · S. G. Siddamsetti · Pooja G · B. R. Chandavarkar  
Department of Computer Science and Engineering, National Institute of Technology Karnataka,  
Mangalore, India

related to the false information so as to verify the information as there are many fake websites with IP addresses very similar to original websites. Users usually find it hard to differentiate between them.

The information we consume has a big impact on our capacity to make choices, and it also has an impact on our perspective. A recent example is the proliferation of a novel coronavirus, with false information about the origin and behaviour of the virus spreading throughout the web. As more people became aware of the misleading material on the Internet, the issue became increasingly serious [11]. To tackle this problem, detection of fake news at an every early stage is important yet challenging. Although numerous methods are proposed on it earlier, the authentication of content creators using digital certificates can be a novel technique. We propose a method to detect legitimate news by authenticating whether the creator of news is legit.

The objectives of the chapter can be summarized as follows:

- Create a model to process uncertain news that can estimate the likelihood of a media story being phony or not using knowledge from past news reports.
- Provide authorization of news with their respective content creators so as to protect their creations from manipulation.

The following is how the rest of the chapter is organized: Sect. 2 will be a compilation of insights from various research papers with a brief description of different techniques used. Section 3 gives details of methodology of the proposed solution. Section 4 presents comparison and impact of the proposed model. Finally, Sect. 5 summarizes our findings and suggests some study directions for the future.

## 2 Related Work

There are many solutions proposed on detecting fake news in different categories of multimedia. Some important ones have been discussed below.

Alves et al. [12] discussed a few important algorithms to detect fake news from multimedia. These algorithms are OTP (One Time Pad) algorithm, RSA (Rivest Shamir Adleman) algorithm, and Baptista's chaos-based algorithm.

Aldwairi and Tawalbeh [13] used cross-referencing of video and speech features to resist copy attacks. All the metadata related to content features and watermarks are stored in the blockchain for the purpose of tamper-proof recording. To evaluate the embedded watermark's robustness against common signal processing and video integrity attacks, the simulations are performed. The application of ANN and ML approaches for categorizing, choosing, and reacting to prospective assaults on social media sites is discussed in this research. ANNs are made up of a large number of artificial neurons that produce, process, and evaluate data. ANNs enable them to make speedy judgements and tackle situations with a lot of uncertainty.

In a research by Megías et al. [14], the principles of an ongoing project DISSIMILAR have been proposed, which uses information hiding techniques like

digital watermarking combined with ML. The project's most notable feature is that it enables content owner recognition/proof of ownership, authentication/manipulation detection, and tracking of the share/export of the content. When a content producer posts his content in social media, they are watermarked for authentication. When a fake news creator tries to post the tampered content, various ML algorithms are used to verify the similarities and dissimilarities of this content with previously existing ones and major similarities imply that the content is tampered/fake.

According to [15], blockchain ensures data origin and the ability to be traced by establishing a peer-to-peer secure system for storing and exchanging data that is visible, irreversible, and provable. Thus, it ensures trust network and prevents fake news. In a research by Shahbazi and Byun [16], an integrated system built on blockchain and NLP to detect false information and better forecast spurious user accounts and postings using ML techniques. This procedure employs the reinforcement learning technique. The blockchain framework, which gives the structure of online media authority evidence, adds to the security. The idea behind this system is to provide a safe platform that can forecast and recognize fake stories in social media networks.

Al Shariah and Khader [17] proposed an architecture based on blockchain technology which uses cryptographic encryption–decryption techniques to detect fake images. The program will encrypt the secret image by using data owner encryption key once it is shared. Once the block content and block head have been formed, they are compacted together to form a block, which is then stored in block chain storage. Once the image has been downloaded, the system must retrieve the block body and block head individually, decrypt the block body using the approved key, and generate the secret image.

Dhiran et al. [18] proposed a blockchain-based method for video fraud detection. Because every node in the blockchain stores data collected in the form of the hash value, some cryptographic techniques are employed to locate a unique characteristic in all of the clips that can act as the hash value of the clip. When a video is manipulated with, the hash value alters, enabling any video fraud to be discovered.

Based on the responses gathered, a statistical model is proposed by Aldwairi and Tawalbeh [13] for estimating the trustworthiness of the news item. ProBlock, a blockchain approach, is developed to assure the accuracy of information disseminated. Aldwairi and Alwahedi [19] proposed a method to identifying spurious news depending on the headlines of the news. It identifies the common features in headlines of fake news and detects based on those features.

In this domain of information engineering, machine learning algorithms have showed to be quite beneficial in a variety of situations. A supervised learning paradigm has been used in the majority of ML algorithms used for spurious news detection. ML is a set of approaches that makes it possible for software products to get more precise results without requiring them to be reprogrammed. Some of the algorithms that are used to detect fake news are naive Bayes classifier [20], decision trees [21], SVM [11], random forest [22], KNN [23], and logistic regression [24].

### 3 Background

This section introduces all the techniques used in the proposed solution model including X.509 certificates, natural language processing, blockchain, and image matching techniques.

#### 3.1 X.509 Certificates

There are two main types of cryptographic encryptions, symmetric key encryption and public-key encryption. Symmetric key encryption involves a shared secret key between two parties, and they securely communicate using the shared key. In public key cryptography, each user generates a pair of private and public keys. The users do not share their private keys with anyone, and the public keys are accessible to everyone. X.509 certificate is a digital certificate in the Public Key Infrastructure (PKI) format and helps in authentication and Public Key Distribution. Many protocols like S/MIME (secure email), IP security (network layer security), SSL/TLS (transport layer security), and SET (e-commerce) use these certificates to ensure secure transfer to data. The Certificate Authority (CA) issues these certificates. It mainly contains a user ID, public key information of the user, period of validity, details about certificate issuer, and others, including a signature or message digest of all the above information encrypted by the issuer's private key [25]. There can be a hierarchy of certificate issuing authorities for a broad population range. All the fields except the signature are in plaintext format in the certificate [26]. The signature helps in authenticating a user with the certificate.

One way to ensure the legitimacy of news is to check the website's address. Fake news creators slightly modify the addresses of legitimate and known websites. X.509 certificates help filter original websites.

#### 3.2 Blockchain

Blockchain is a decentralized technology used for secure distributed data storage using cryptographic hashing and digital signatures [27]. Blockchain stores information as nodes. Each node contains the encrypted hash digest from the previous node. Cryptographic hashing is an efficient way to detect unauthorized modifications to the data. A cryptographic hash function can convert any plaintext into a unique text string called a hash digest. This hash digest is encrypted using a private key to produce a digital signature. Every node in the blockchain stores the content of that node, timestamp, and the digital signature of the previous node. The node's content can be the news data in our case and is stored in plaintext format. Blockchain secures the integrity of the data stored in nodes. To verify the integrity of the data

in the nodes, one can compute the hash digest of that message and match it with the encrypted hash digest in the next node after decrypting it with the public key [28]. The significance of the hash function is that a single letter change in the message changes many characters in the hash digest. So any tampering in the news is detected while matching the hash digests. Our model uses blockchain to store the news data entered by the verified users.

### ***3.3 Natural Language Processing***

Natural Language Processing, NLP, is a branch of Artificial Intelligence (AI) that is concerned with giving the computer the capability to comprehend the text and speech and respond accordingly with text or audio. In NLP, semantic matching techniques [29, 30] help find similarities in the implication of two texts. Semantic matching can analyze the text of the news, compare it with other similar information on the Internet, and draw valuable conclusions which help determine the legitimacy of news. Fake news is largely comprised of click-bait, which stands out in news feed with flashy headlines and arouse curiosity. When clicked on them, the content often does not match the headlines. Semantic matching can detect such news by semantically comparing the headlines with the news content. NLP gives the computer the ability to understand the messages. So, we can filter out messages meant to defame others by revealing their personal information, spreading false news, hate speech to defame a person or entity. The blockchain stores only the news from verified content creators. So if the user wants to verify the legitimacy of information created by a non-verified content creator, our model can compare it with the information in the blockchain through NLP methods and show the percentage of the legitimacy of the news to the user.

### ***3.4 Image Matching Techniques***

Another challenging issue in the domain of fake news is Deepfake images. Some part of the image is modified intentionally to produce a phony picture. Image matching techniques can identify the tampered images by comparing the fake with legit content [31]. Blockchain stores the image content from verified users. So any image is compared with these images to compute their percentage of fakeness. Deep convolutional neural networks (CNNs) extract features from verified pictures and compare them with the others. Multi-layered neurons constitute CNN. The number of layers in CNN must be optimized to achieve maximum efficiency. The verified images are first fed into this model to extract the features. The unverified image is then fed to compute the percentage of similarity and dissimilarity of this image with the available ones.

### 4 Proposed Model

This section provides a detailed outline and workflow of the proposed model (Fig. 1).

#### 4.1 Issue of Certificates

The content creators need to request certificates from the certificate authority (CA). A certificate authority is a trusted third party that verifies and authenticates websites. The browser has a set of trusted CAs. When a website is accessed in the browser, it checks whether the certificate is valid and from a trusted CA [32]. The CA verifies whether the requested entity is legitimate and issues certificates in X.509 format to only legitimate content creators. Generally, there are three types of verification, i.e., domain validation, organization validation, and extended validation. Domain validation involves verification of the requestor to be the legitimate manager of the company to which the certificate is being issued. In organization validation, the CA researches upon the information provided by the requestor and gathers additional information to confirm the company’s legitimacy. Extended validation

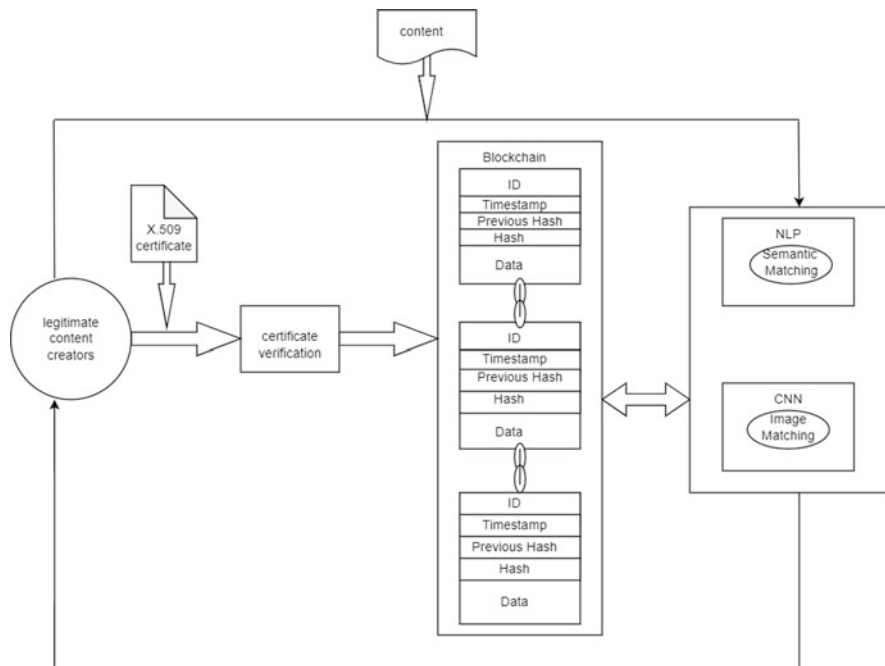


Fig. 1 Proposed model



is a 1-to-5-day validation process where the CA goes above and beyond the norms of the organization validation process to confirm that the organization is genuinely legitimate.

## ***4.2 Authentication and Content Verification***

After receiving the certificate from CA, whenever the content creator company wants to post content, it can use our model to identify it as verified content. It needs to send the certificate issued to him by CA to the model. The model will confirm whether the certificate is legit and sends a confirmation message. The content creator encrypts the news content using his private key and sends it to the model. The model decrypts the news data with the public key in the certificate. Then, the news text is forwarded to the NLP model for semantic matching to match the news content with its title and other related news stored in the blockchain. The NLP model retrieves the information stored in the blockchain to compare the present news with the earlier ones. It also processes the news for types of content like defamation, hate speech, and others. It sends a report of NLP processing to the concerned content creator if the content is found too out of order, as it might be from a hacker who hacked the legitimate content creator. image matching processing block gets images and videos in the news content sent by the content creator. They are tested for Deepfake and face-swap modifications. It reports to the respective content creator on discovering tampered image or video content. After verifying the content as legitimate, it is appended to the blockchain and stored there.

From the user perspective, we have two approaches to our model.

- Assume the user wants to filter out all the news from unverified sources. Blockchain in the proposed model contains only legitimate information from trusted sources so that the model can display only the data from the blockchain in the recent first order.
- If the user wants to compare the legitimacy of unverified news, our model can process the news in NLP block and image matching block to predict the percentage legitimacy.

## ***4.3 Workflow of the Model***

This section depicts the workflow of the model in the form of a flowchart for a better understanding of the reader (Fig. 2).

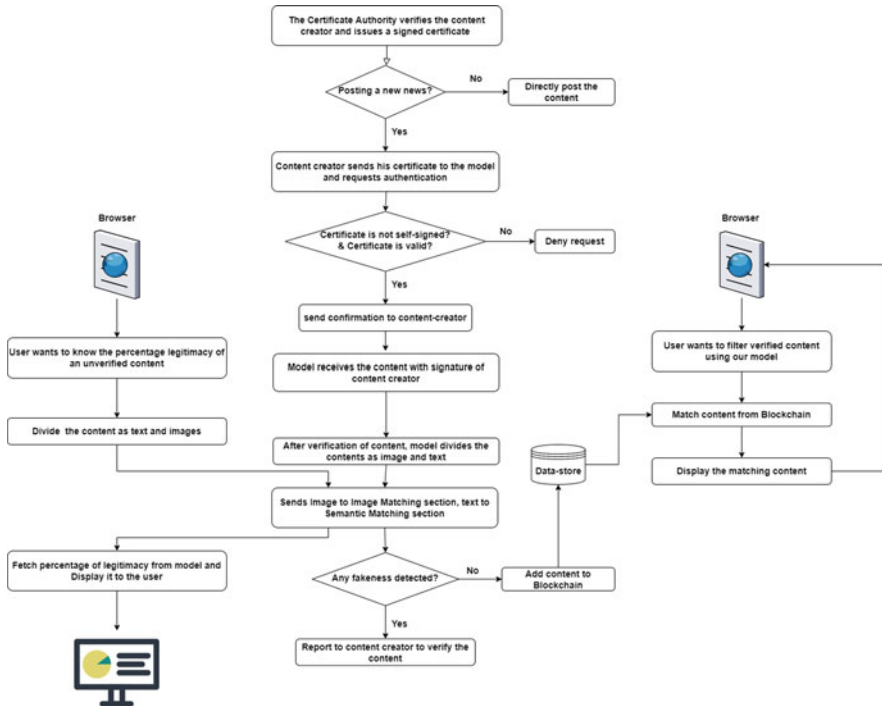


Fig. 2 Workflow of the model

## 5 Comparison and Impact

The novelties, comparison, and expected impact of the proposed model are as follows:

1. An advantage of this model is that the user need not separate the fake content; the model does it. The proposed model can be implemented as an extension to a browser (e.g., chrome extension) to filter out fake news. It avoids exposure to the potentially harmful influences of fake news, compromising our details by using a fake website that seems original.
2. Earlier research used digital watermarking to identify the content from verified sources. But this technique cannot be applied to text. In the proposed model, certificates are used to verify the originals, where the sources confirm themselves by a certificate issued by a trusted party. Verification through certificates is a widely used and secure authentication technique.

## 6 Conclusion and Future Work

In this chapter, we discuss the concept of misinformation in multimedia, challenges in detecting misinformation, related research works on detecting misinformation and proposed an efficient model to detect misinformation. The proposed model proves an efficient method to filter out fake news using certificates, blockchains, Natural Language Processing, and image matching. The proposed model adds novelty to the previous models by adding the feature of authentication of the content creator.

The possible future work includes extending the proposed model to detect fake news on social media, improving image tampering detection, and video tampering detection to include lip-syncing and other tampering techniques. Real-time fake news detection of images and videos is another potential research topic. The best solutions should be explored from various fields to detect misinformation effectively.

## References

1. Liang Wu, Fred Morstatter, Kathleen M Carley, and Huan Liu. Misinformation in social media: definition, manipulation, and detection, 2019.
2. Syed Ishfaq Manzoor, Jimmy Singla, et al. Fake news detection using machine learning approaches: A systematic review, 2019.
3. Alessandro Bondielli and Francesco Marcelloni. A survey on fake news and rumour detection techniques, 2019.
4. Ivan Perov, Daiheng Gao, Nikolay Chervoniy, Kunlin Liu, Sugasa Marangonda, Chris Umé, Mr Dpfks, Carl Shift Facenheim, Luis RP, Jian Jiang, et al. DeepFaceLab: Integrated, flexible and extensible face-swapping framework, 2020.
5. Alim Al Ayub Ahmed, Ayman Aljabouh, Praveen Kumar Donepudi, and Myung Suh Choi. Detecting fake news using machine learning: A systematic literature review, 2021.
6. Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. MesoNet: a compact facial video forgery detection network, 2018.
7. Nicolo Bonettini, Edoardo Daniele Cannas, Sara Mandelli, Luca Bondi, Paolo Bestagini, and Stefano Tubaro. Video face manipulation detection through ensemble of CNNs, 2021.
8. Zonyin Shae and Jeffrey Tsai. Ai blockchain platform for trusting news, 2019.
9. Safi ur Rehman, Muhammad U.S Khan, and Mazhar Ali. Blockchain-based approach for proving the source of digital media, 2020.
10. Zeinab Shahbazi and Yung-Cheol Byun. Fake media detection based on natural language processing and blockchain approaches, 2021.
11. Iftikhar Ahmad, Muhammad Yousaf, Suhail Yousaf, and Muhammad Ovais Ahmad. Fake news detection using machine learning ensemble methods, 2020.
12. Warley Alves, Thiago L Prado, Antonio M Batista, and Fabiano AS Ferrari. The dangerous path towards your own cryptography method, 2018.
13. Monther Aldwairi and Lo'ai Tawalbeh. Security techniques for intelligent spam sensing and anomaly detection in online social platforms, 2020.
14. David Megías, Minoru Kuribayashi, Andrea Rosales, and Wojciech Mazurczyk. Dissimilar: Towards fake news detection using information hiding, signal processing and machine learning, 2021.

15. Paula Fraga-Lamas and Tiago M. Fernández-Caramés. Fake news, disinformation, and deep-fakes: Leveraging distributed ledger technologies and blockchain to combat digital deception and counterfeit reality, 2020.
16. Zeinab Shahbazi and Yung-Cheol Byun. Fake media detection based on natural language processing and blockchain approaches, 2021.
17. Njood AlShariah and Abdul Khader. Detecting fake images on social media using machine learning, 01 2019.
18. Aditya Dhiran, Dinesh Kumar, Anshul Arora, et al. Video fraud detection using blockchain, 2020.
19. Monther Aldwairi and Ali Alwahedi. Detecting fake news in social media networks, 2018.
20. Alim Al Ayub Ahmed, Ayman Aljabouh, Praveen Kumar Donepudi, and Myung Suh Choi. Detecting fake news using machine learning: A systematic literature review, 2021.
21. Z Khanam, BN Alwasel, H Sirafi, and M Rashid. Fake news detection using machine learning approaches, 2021.
22. Uma Sharma, Sidarth Saran, and Shankar M Patil. Fake news detection using machine learning algorithms, 2020.
23. Alim Al Ayub Ahmed, Ayman Aljabouh, Praveen Kumar Donepudi, and Myung Suh Choi. Detecting fake news using machine learning: A systematic literature review, 2021.
24. Sohan Mone. Fake news identification CS 229: Machine learning: Group 621, 2017.
25. X.509 certificates format. [https://www.brainkart.com/article/X-509-Certificates\\_8470/](https://www.brainkart.com/article/X-509-Certificates_8470/), 2018. [Online; accessed 18-March-2022].
26. X.509 certificates. <https://sectigo.com/resource-library/what-is-x509-certificate#:~:text=Share%20this-,An%20X.,internet%20communications%20and%20computer%20networking>, 2018. [Online; accessed 18-March-2022].
27. Blockchain. <https://blockgeeks.com/guides/blockchain-cryptography/>, 2018. [Online; accessed 18-March-2022].
28. Blockchain. <https://101blockchains.com/blockchain-cryptography/>, 2018. [Online; accessed 18-March-2022].
29. Semantic matching in NLP. <https://medium.com/georgian-impact-blog/an-introduction-to-semantic-matching-techniques-in-nlp-and-computer-vision-c22bf3cee8e9>, 2018. [Online; accessed 18-March-2022].
30. Fausto Giunchiglia, Mikalai Yatskevich, and Pavel Shvaiko. Semantic matching: Algorithms and implementation, 2007.
31. Srikar Appalaraju and Vineet Chaoji. Image similarity using deep CNN and curriculum learning, 2017. [Online; accessed 18-March-2022].
32. Casey Crane. What is a certificate authority (CA) and what do they do? URL <https://www.thesslstore.com/blog/what-is-a-certificate-authority-ca-and-what-do-they-do/>.

# End-to-End Network Slicing for 5G and Beyond Communications



Rohit Kumar Gupta, Sudhir Kumar, Praveen Kumar, and Rajiv Misra

## 1 Introduction

Nowadays, every person wants to do the work in automatically environment, for that, we are using sensors, micro-controller, and networks so that we can communicate all the sensors and transfer the data from one place to another place with low latency. To provide the services like security to the IoT applications, data transfer required dedicated networks, and therefore it reduced the latency of the network. To achieve this aim, one thing comes into the picture called MEC. MEC is basically used to reduce the computational latency from the server-side, and another hand it can also reduce the traffic of the network. In the early days, MEC was installed on the server-side and with the help of the MEC. We were able to do all the computational power and give the response to the client as soon as possible. But the problem is that several clients are connected to the server and the server responds one by one, which leads to generating traffic in the network [9, 10]. Due to this reason MECs install inside the network, and therefore, all the MECs are able to communicate with the server for the first time only, and after that, it can be able to give the response to the query. With the help of this mechanism, MEC does not communicate with the server in a very frequent manner, which leads to reduced traffic in the network [12]. In the current scenario, we are moving forward and trying to use a smart environment like a smart home, smart city, smart car, smart state, etc. Due to that reason, sensors are playing a major role. If we want to monitor and control the sensors. We can use the controller, and all the controllers are connected to the Internet. With the help of the Internet, we can monitor the controller and also

---

R. K. Gupta (✉) · S. Kumar · P. Kumar · R. Misra

Department of Computer Science and Engineering, Indian Institute of Technology, Patna, India  
e-mail: [1821cs16@iitp.ac.in](mailto:1821cs16@iitp.ac.in); [sudhir\\_2221cs14@iitp.ac.in](mailto:sudhir_2221cs14@iitp.ac.in); [praveen\\_2221cs11@iitp.ac.in](mailto:praveen_2221cs11@iitp.ac.in);  
[rajivm@iitp.ac.in](mailto:rajivm@iitp.ac.in)

control the controller by the end-users. To control things, data can be analysed first. After these analysed data can be transferred from one place to the other place. To transfer the data, we need to connect the sensors to the networks [7]. Nowadays, lots of IoT applications are running that require lots of the network to transfer the data. If all the applications or sensors or micro-controllers transfer the data at the same time, definitely traffic and latency will increase. To deal with the heavy network traffic and low latency issue, a special kind of mechanism is required. SDN and NFV are used to do the slicing of the network, which leads to the allocation of the dedicated networks to the particular application or service. With the help of this mechanism, we can reduce the latency. 4G does not support SDN and NFV services, but 5G supports this service. In this chapter, the author depicts how we do the virtualization in the 5G network [1, 2, 8].

The chapter follows the structure in the very first part author depicts the introduction, in the second part, the author explains the various technologies used in virtualization with the help of previous work done by the various authors, in the third part, the author shows system models for the slicing, in the fourth part, various results are shown, and the fifth part concludes this chapter.

## 2 Related Work

Nowadays, everyone uses personal electronic gadgets and wants to live in a smart environment like smart homes, smart cities, and smart agents to reduce human work. For monitoring and controlling all these applications, we required more bandwidth, high data rate, low latency, etc. Thus, we are looking to move forward from 4G to 5G. In 5G, we can deal with flexibility, mobility, high data rate, latency, slicing, etc. This chapter mainly focuses on network slicing [1]. In 5G, we are using slicing to improve the data rate, better connectivity, and system capacity. Software-defined networks (SDNs) and network function virtualization (NFV), these two techniques play a huge role to achieve 5G slicing without changing any physical structure of the network [5, 6]. SDN is used for software abstraction and NFV is used for the virtualization of the network. These 2 services are provided in the 5G network to meet the demands of today's network. The taxonomy is based on different factors, such as network nodes, slicing scope, slice isolation, and slice management, which are the types of use cases served by the 5G network and enabler techniques, namely SDN and NFV [2, 4]. For the slicing, we try to identify a few things—first thing is to try to understand the area of the network where slicing will implement and the second one understanding the use case so that we analyse how many slices are required [9, 11]. The main feature of SDN and NFV is to create a slice, and each slice has a different kind of attribute. The purpose of slicing is to reduce the latency and high throughput. It takes live streaming as an example, so this scenario required low latency and high bandwidth for the high-definition video. Network slicing implementation on a radio access network (RAN) in a 5G environment is a very difficult task, and hence we required a complex network design. So we required

technologies, and therefore we can achieve the slicing without lots of changes in 5G architecture. In 5G, the number of slices created totally depends on the demand of the telecommunication network. Slicing means it is a small kind of network. Instead of using the whole network, we are using part of the network, and another part of the network is used by different applications like IoT applications, data transfer of user, etc. [13]. The 5G network is envisioned to comprise small granularity, loosely coupled, highly cohesive modular network function services. Each service is realized by a specific functionality and self-contained, which makes it possible to update individual services independently with less impact on other services. The smaller and modular network function components (NFCs) can be flexibly chained and connected to form larger network functions or end-to-end network slices on demand [3, 14]. We used SDN enable to generate the softwarization with the same kind of functionality and NFV used for virtualization. So that we slice virtual networks for use in different ways in real-world applications. SDN enables a common infrastructure to support multiple clients instances efficiently. As the client context provides a complete abstract set of resources isolated by the SDN controller's virtualization, it is natural for SDN to support network slicing In this chapter, the author depicts the functionality and reason to move 5G with the help of SDN and NFV. Mainly, we find out the 2 key points for the slicing listed followed. The first considered point is which area of the network is required to slice in the 5G network, and another one is how many slices are required to full the demand of the requirements [12] (Fig. 1).

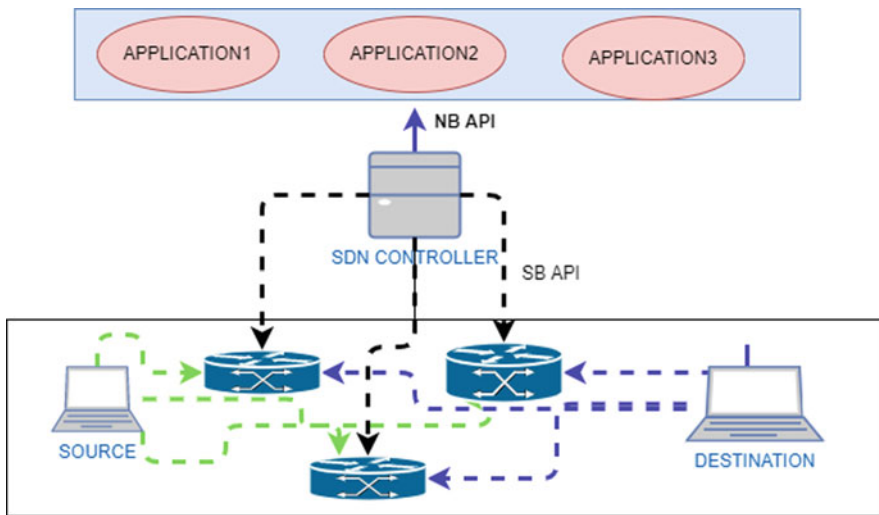


Fig. 1 SDN architecture

### 3 System Models

The network is a combination of 3 technologies. The first one is RAN, the second thing is CORE NETWORK, and the third is USER. First, we try to understand our requirements, after that we create the network to fulfill the necessity and demands. Then, we focus on a few parameters like mobility, resource management, security, low latency, and high bandwidths. In this chapter, we explained the network concept of SDN, NFV, and its use case. SDN is simply an abstraction for depicting the components and their functions, as well as the set of protocols for managing and forwarding the data. The NFV architecture emphasizes in Fig. 2 as the use of virtualization for various network node functions. Virtualization totally depends on the requirement of the demands of the application. Users are connected to the network through RAN, RAN is connected to the core network, and then the core network is connected to the public Internet. NFV works on the sub-network concept between the core network and the public.

Each sub-network creates a service instance according to the service required. And the last one is the use case. If we have a basic network, then we are able to use it in any applications. The emergence of IoT has added to the brackets of use cases that the 5G network must serve. In IoT, we have multiple types of use cases, so to

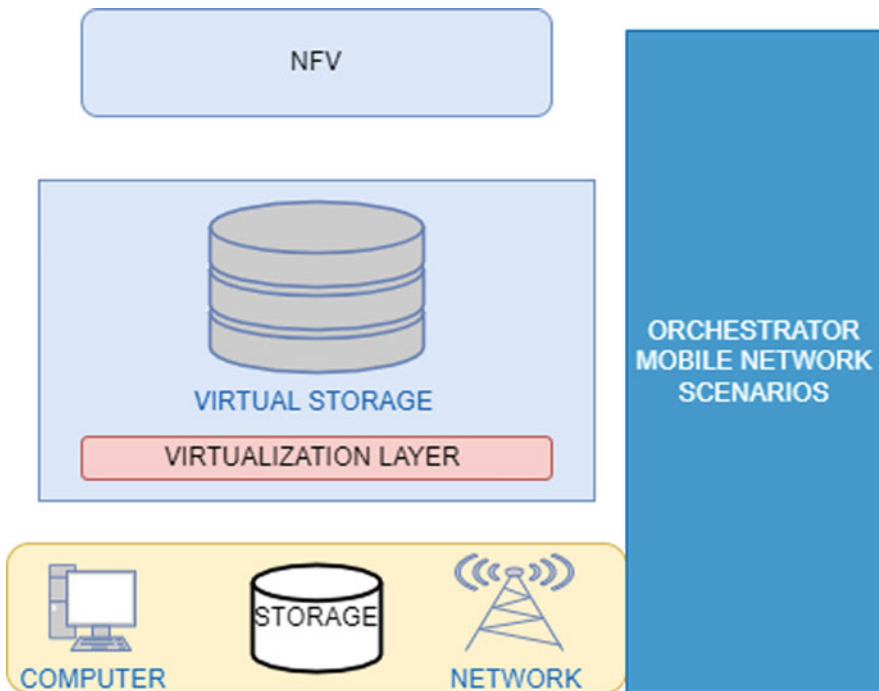
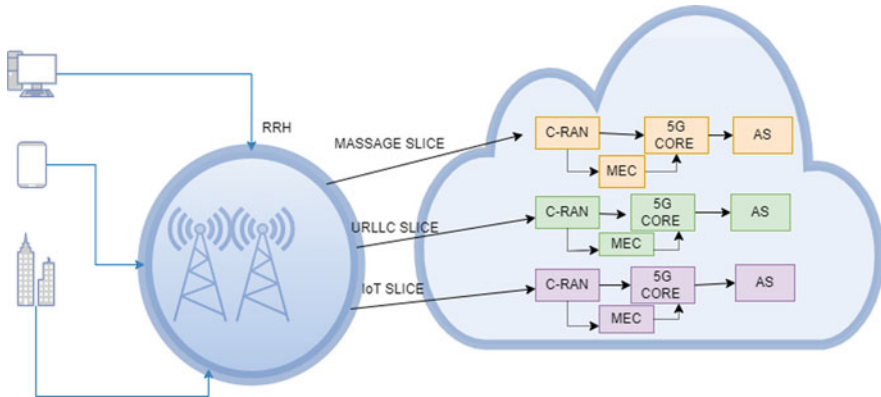


Fig. 2 NFV architecture





**Fig. 3** RAN network architecture

deliver these use cases we are using network slicing according to the requirements. So nowadays we are using dynamic use cases so that it increases the complexity level of the slicing in the network because it will be created at the run time using reinforcement learning. Hence, slicing required more flexibility and is dynamic in nature. Each use case is having different kinds of attributes; some are required high bandwidth, ultra-reliable low latency, very low latency, and use of IoT applications. Core network—the core network has more functionality, and connecting the RAN with a third party is a kind of functionality of the core network. With the help of serving and packet data gateways, core networks were able to connect with the public network. The main components within RAN are the base stations and the antennas. Figure 2 of NFV depicts that all the users are connected to the RRH, and according to the requirements, slicing is done in the network. Nowadays, to reduce the latency of the network, we used the MEC. Figure 3 shows that MEC is installed in between the C-RAN and 5G core, so using this model of architecture definitely reduced the latency of the network.

In the current scenario, everyone required less latency inside the network for trying to achieve less latency. Hence, we used multi-access edge computing. With the help of this, we reduced the computational power to search for the relevant information inside the controller. If we set up MEC inside the controller, then it reduced the time complexity but not up to the marks so that we are set up the MEC on the network part. Therefore, we are able to achieve the minimum latency in the given network. If any UE wants to communicate to the controller, then the controller is able to transfer the data in a very quick time.

For achieving a flexible and realistic AI-based scenario, we are using CAI. It provides flexibility between different network topologies and quick deployment. SDN and RAN controller provides the information to the agent and the agent acts accordingly. ORCHESTRATION is used to create a cluster with available machines, and Container Network Interface (CNI) is used to interconnect them to make the application for deployment and orchestration of the cluster nodes. Two types of

nodes are used by each cluster in this model: the very first is the master node and another one is the worker node. Master responsible to execute the container orchestrator commands makes ready for deployments of application. Workers are responsible for executing the containers from applications requested by the master nodes. An often-utilised technique is called MONO, which is used to virtualize networks. If the requirements are not higher side so we used Kubernetes in CNI. With the help of this, we can easily make the cluster and deployed the node inside the clusters. We are using the Docker platform for the implementation of the core and RAN networks. The Kubernetes assign numbers to each cluster machine to identify which machines are executed in each module which is defined by each cluster. Now, we deployed the node according to the machine specification and requirement. Kubernetes verify which machine has a USRP board connection. Based on Mininet, we can implement a testbed in both front haul and backhaul through virtual network deployment (VND). Virtual network deployment depends on Mininet. First, we created virtual Ethernet devices (VETHs) and established the connection between them, generate the traffic, and forward data between two machines through the Mininet topology.

RAN programmability, also called SD-RAN, works as an abstraction of the RAN resources by providing an API that enables the Services of Orchestrator entity to dynamically manage the RAN resources and provide information about the mobile network. For controlling and management of RAN resources, we used Flex RAN APIs. We make sure that it is enabled while developing application. CAI TESTBED use cases are available for improvement in mobile network functions. we are using the structure of CAI TESTBED use cases and integrating it with the AI agents, so we can apply it in a different scenario. We used AI agents in two different uses cases. First, we are using RAN slicing application to monitor how many resource blocks are allocated to each requested slice. And the second application is where we place the VNFs in a cluster machine. In this chapter, we are using the C-RAN architecture used for the design and prototypes of the network slice solutions. The objective of this architecture is to allow us to distribute the spectrum among each slice based on the requirements. To identify application requirements, providing the services according to the requirement of the real-time states. It was achieved by the FLEXRAN. For the implementation of C-RAN architecture, we used a few steps. In the first step, simplified topology for front haul and backhaul and the second step is Not to consider any type of network Delay. If we have low latency, then link delay definitely occurred in front haul and backhaul, and it also affects throughput. In this chapter, the author proposed a testbed to give stringent results for slice application. We are using FLEXRAN controllers and core networks to contain the information. We feed the information in ML models that are provided by the core and FLEX RAN and do the slicing provided by the mobile network. In this chapter, FlexRAN is used to implement the RAN slicing scenario and consider eNBs in CRAN architecture. Mininet used to virtualize the backhaul and core network elements is used to adaptively perform eNodeB assignment. Mininet allows us to run a virtual network with routers, switches, and hosts. FleaxRAN gives information about how many numbers of mobile phones (UE) are connected. The provided

source code gives details about all the inputs to the AI agent, which include the number of UEs associated with each slice, isolation of resources, and percentage of RBs for each slice. This chapter achieves 3 interfaces of radio blocks allocated to each slice with the help of the output of the ML model. We use DNN (deep neural network) used for RAN slicing. We can also use other learning techniques like reinforcement and a decisions tree. We first trained the agent using Kubeflow, and then we used deployed pipeline components to describe it. Once we deploy, we are able to generate the ML output. In this chapter, we used mobile and called a UE. Each UE is connected to the RRU, Fig. 3. Each UE client requires different amounts of throughputs.

## 4 Results

In this chapter, the author used two approaches for virtualization and creating clusters. The first approach for the virtualization used Dockers and Kubernetes used for the clustering. With the help of this two, we are able to create virtualization. For the virtualization, the first thing need to do is to install Ubuntu in the virtual box, then install the Docker, and configure it. The total time of the experiment was 179 ms. All the 3 UEs are active in the first 20 ms, so if all are active, then we are able to achieve maximum throughput. UE3 is disconnected for 20 to 40 s. Once the 40 ms are completed, UE2 is disconnected. Once UE2 was disconnected, we are able to connect the UE3. Hence, the same kind of traffic was generated with the help of these 2 UEs with strategies for RB allocation. If we are not using any kind of slicing, then the scheduler provides an equal number of ration of radio resources among all UEs those are connected. Let us assume an instance of time all the UEs are able to achieve the 6 Mbps in starting 20 ms. If any UEs are disconnected (let us assume UE 3 was disconnected), then the rate is divided into two equal parts and UEs are able to reach 9 Mbps. If we are using AI agent allocation, then we used 2 things, the first one is CBR and the second one is MBR so that all the requirements are fulfill in the first 20 milliseconds, and then AI agents keep the bit rate into the CBR slice to provide the constant bit rate. And remaining RBSs are maintained by the MBR slice. All these things happened in 20 to 40 ms. In the last 20 ms, more resources are connected to the MBR. These all are fewer priority slices for achieving the Mbps.

In Fig. 4, we are able to make a three virtual machine interface, and according to the requirement, we make the clusters, and therefore, if we got the same type of the request, then we put it in the same cluster, otherwise we make it a different cluster, and all the cluster are made in the run time.

In this result, Fig. 5, we can see the amount of time required to allocate the interfaces to a particular request. According to our requirements, we can create an interface. In this chapter, three interfaces were created, and for allocation of these three interfaces, we required some amount of the time slot which is already mentioned in Fig. 5 and also shown the graphical way mentioned in Fig. 6. In Table 1, we show the comparison between the standard result and our performed

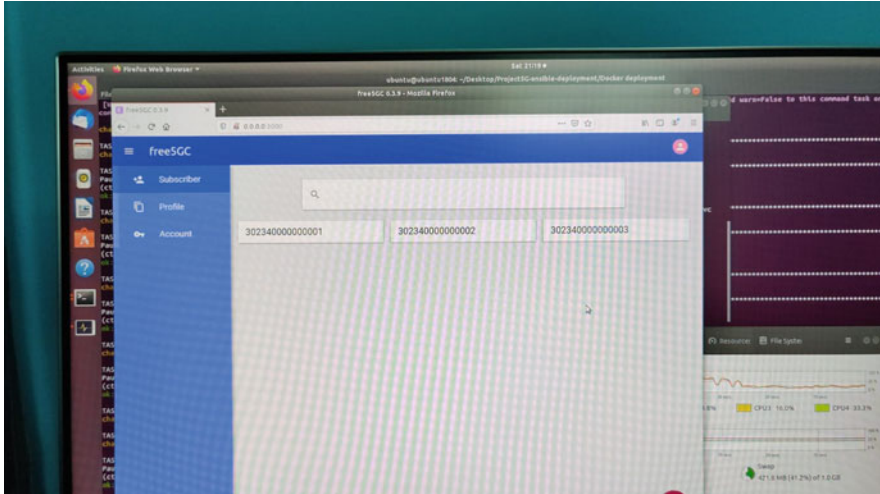


Fig. 4 The interface of 5G network using Docker

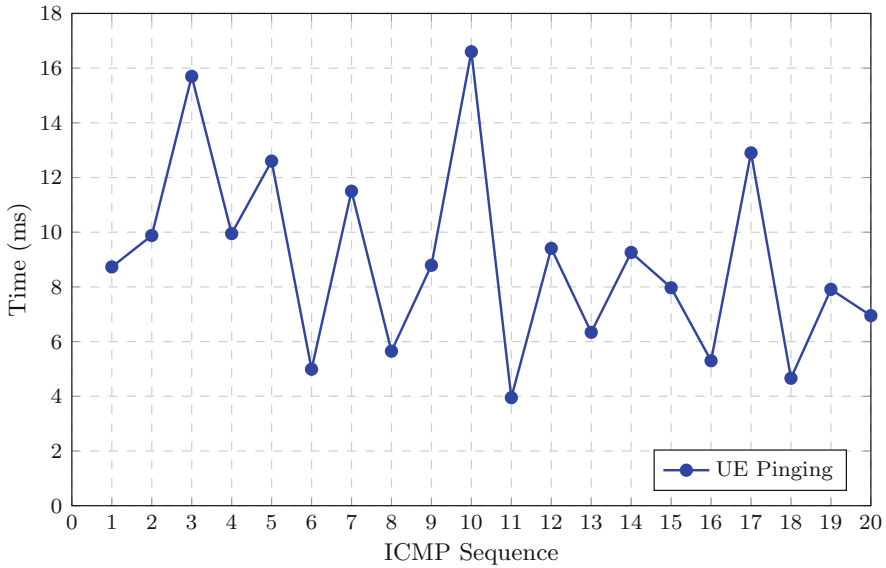
```
PING google.com (142.250.196.46) 56(84) bytes of data:
64 bytes from maa03s45-in-f14.1e100.net (142.250.196.46): icmp_seq=1 ttl=118 time=8.73 ms
64 bytes from maa03s45-in-f14.1e100.net (142.250.196.46): icmp_seq=2 ttl=118 time=9.88 ms
64 bytes from maa03s45-in-f14.1e100.net (142.250.196.46): icmp_seq=3 ttl=118 time=15.7 ms
64 bytes from maa03s45-in-f14.1e100.net (142.250.196.46): icmp_seq=4 ttl=118 time=995 ms
64 bytes from maa03s45-in-f14.1e100.net (142.250.196.46): icmp_seq=5 ttl=118 time=12.6 ms
64 bytes from maa03s45-in-f14.1e100.net (142.250.196.46): icmp_seq=6 ttl=118 time=4.99 ms
64 bytes from maa03s45-in-f14.1e100.net (142.250.196.46): icmp_seq=7 ttl=118 time=11.5 ms
64 bytes from maa03s45-in-f14.1e100.net (142.250.196.46): icmp_seq=8 ttl=118 time=5.65 ms
64 bytes from maa03s45-in-f14.1e100.net (142.250.196.46): icmp_seq=9 ttl=118 time=8.79 ms
64 bytes from maa03s45-in-f14.1e100.net (142.250.196.46): icmp_seq=10 ttl=118 time=16.6 ms
64 bytes from maa03s45-in-f14.1e100.net (142.250.196.46): icmp_seq=11 ttl=118 time=3.95 ms
64 bytes from maa03s45-in-f14.1e100.net (142.250.196.46): icmp_seq=12 ttl=118 time=9.41 ms
64 bytes from maa03s45-in-f14.1e100.net (142.250.196.46): icmp_seq=13 ttl=118 time=6.34 ms
64 bytes from maa03s45-in-f14.1e100.net (142.250.196.46): icmp_seq=14 ttl=118 time=9.26 ms
64 bytes from maa03s45-in-f14.1e100.net (142.250.196.46): icmp_seq=15 ttl=118 time=7.97 ms
64 bytes from maa03s45-in-f14.1e100.net (142.250.196.46): icmp_seq=16 ttl=118 time=5.30 ms
64 bytes from maa03s45-in-f14.1e100.net (142.250.196.46): icmp_seq=17 ttl=118 time=12.9 ms
64 bytes from maa03s45-in-f14.1e100.net (142.250.196.46): icmp_seq=18 ttl=118 time=4.66 ms
64 bytes from maa03s45-in-f14.1e100.net (142.250.196.46): icmp_seq=19 ttl=118 time=7.91 ms
```

Fig. 5 Result of UE ping

results. So we are able to see the ping time of the network is very less than the standard result.

## 5 Conclusion

In this chapter, the author depicts the virtualization of the network with the help of SDN and NFV. We can create the three networks using tools like Docker. We can divide the RAN network into three parts, and we are able to connect the IoT applications to every virtualized network and sensors or micro-controllers are able



**Fig. 6** Result of UE pinging

**Table 1** Comparison between standard value and the proposed results

ICMP sequence	Standard value in ms	Result value in ms
1	45.5	8.73
2	43.1	9.88
3	44.7	15.7
4	43.6	9.9
5	42.7	12.6
6	44.7	4.99
7	42.1	11.5
8	42.1	5.65
9	43.2	8.79
10	43.0	16.6
11	42.1	3.95
12	42.5	9.41

to transfer the data with a dedicated channel, and hence we definitely reduced the latency. In future, we could work towards ultra-reliable low latency communications for 5G and 6G networks.

## References

1. Barakabitze, A.A., Ahmad, A., Mijumbi, R., Hines, A.: 5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges. *Computer Networks* **167**, 106984 (2020)

2. Bonfim, M.S., Dias, K.L., Fernandes, S.F.: Integrated NFV/SDN architectures: A systematic literature review. *ACM Computing Surveys (CSUR)* **51**(6), 1–39 (2019)
3. Gupta, R.K., Choubey, A., Jain, S., Greeshma, R.R., Misra, R.: Machine learning based network slicing and resource allocation for electric vehicles (EVs). In: *International Conference on Internet of Things and Connected Technologies*. pp. 333–347. Springer, Cham (2020)
4. Gupta, R.K., Misra, R.: Machine learning-based slice allocation algorithms in 5G networks. In: *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)*. pp. 1–4. IEEE (2019)
5. Gupta, R.K., Ranjan, A., Moid, M.A., Misra, R.: Deep-learning based mobile-traffic forecasting for resource utilization in 5G network slicing. In: *International Conference on Internet of Things and Connected Technologies*. pp. 410–424. Springer (2020)
6. Gupta, R.K., Sahoo, B.: Security issues in software-defined networks. *IUP Journal of Information Technology* **14**(2) (2018)
7. Kaur, K., Garg, S., Aujla, G.S., Kumar, N., Rodrigues, J.J., Guizani, M.: Edge computing in the industrial Internet of Things environment: Software-defined-networks-based edge-cloud interplay. *IEEE communications magazine* **56**(2), 44–51 (2018)
8. Laghrissi, A., Taleb, T.: A survey on the placement of virtual resources and virtual network functions. *IEEE Communications Surveys & Tutorials* **21**(2), 1409–1434 (2018)
9. Li, B., Fei, Z., Zhang, Y.: UAV communications for 5G and beyond: Recent advances and future trends. *IEEE Internet of Things Journal* **6**(2), 2241–2263 (2018)
10. Mozaffari, M., Saad, W., Bennis, M., Debbah, M.: Mobile unmanned aerial vehicles (UAVs) for energy-efficient internet of things communications. *IEEE Transactions on Wireless Communications* **16**(11), 7574–7589 (2017)
11. Nahum, C.V., Pinto, L.D.N.M., Tavares, V.B., Batista, P., Lins, S., Linder, N., Klautau, A.: Testbed for 5G connected artificial intelligence on virtualized networks. *IEEE Access* **8**, 223202–223213 (2020)
12. Shah, S.D.A., Gregory, M.A., Li, S.: Cloud-native network slicing using software defined networking based multi-access edge computing: a survey. *IEEE Access* **9**, 10903–10924 (2021)
13. Subedi, P., Alsadoon, A., Prasad, P., Rehman, S., Giweli, N., Imran, M., Arif, S.: Network slicing: a next generation 5G perspective. *EURASIP Journal on Wireless Communications and Networking* **2021**(1), 1–26 (2021)
14. Zhang, S.: An overview of network slicing for 5G. *IEEE Wireless Communications* **26**(3), 111–117 (2019)

# Transparency in Content and Source Moderation



Adithya Rajesh C., Chathanya Shyam D., Pranav D. V.,  
and B R Chandavarkar

## 1 Introduction

Millions of users are active on several social media platforms such as Twitter, Facebook, and Google. Due to this, there are bound to be polarizing views of people on certain topics, which may be more harmful than constructive. Moderation of such content is critical to the safety of the community. More importantly, fair moderation is of utmost importance. In central platforms, there is complete authority over the content and moderation while having no form of transparency. This brings about the possibility of bias of content moderation leading certain segments of the population to believe that these platforms are biased. This paper aims to achieve a more fair and transparent content moderation system.

The current central platforms are not held accountable for their actions nor provide complete transparency about their decision-making process. In the past few years, society has become more polarized [1], and everywhere across the world, there has been an increase in violent protests and riots. Hence, it is crucial to develop an effective content moderation process that is transparent and offers proper explanations for content removal. Such a system will improve the trust of the general population on these platforms and lead to less radicalization.

The current content moderation scheme is a mixture of machine learning models and manual moderation by a selected group of people (moderators). There are problems with both techniques. The machine learning models cannot catch all harmful posts, and there will always be false positives and false negatives. Some models are also inherently biased toward certain protected classes of people such as gender, race, etc. Correction factors have to be added to such models. Manual

---

Adithya Rajesh C. (✉) · Chathanya Shyam D. · Pranav D. V. · B. R. Chandavarkar  
National Institute of Technology Karnataka, Mangaluru, India

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023  
R. Misra et al. (eds.), *Advances in Data Science and Artificial Intelligence*,  
Springer Proceedings in Mathematics & Statistics 403,  
[https://doi.org/10.1007/978-3-031-16178-0\\_31](https://doi.org/10.1007/978-3-031-16178-0_31)

445

moderation by moderators is much more accurate than models, but they are not scalable as they require human resources. There is also no transparency over the actions of moderators. The current moderation system is very subjective to bias, and the community's safety often depends on the quality of arbitrarily chosen moderators.

We introduce a novel algorithm of an adaptive NLP model that acts as the first layer of harmful content detection. The design of this model is such that it gets better results with each post. The dataset grows with the growth of the community. So eventually, we expect this model to reach a good accuracy in detecting harmful content. The second part of the algorithm contains the working of an ELO-based self-balancing system of user trust. We employ trust calculations of each user and determine the quality of the user as a person with powers. Only the most trusted users will have to ability to act as moderators. The trust of each user is determined by a weighted voting algorithm described below.

The chapter has been organized in an orderly manner. Section 2 contains an overview of related approaches to solve this problem and their drawbacks. Section 3 is the main part of the paper which presents the working of the proposed algorithm. Each component in the algorithm has been described in a separate subsection. Section 4 goes over the analysis of the different working components of the algorithm. Finally, Sect. 5 presents the conclusion of the proposed approach and the possible improvements that can be researched further.

## 2 Related Works

Natural Language Processing (NLP) is a field of artificial intelligence that aims to understand the meaning of text. Many different subdomains such as parts of speech (PoS) tagging, named entity recognition (NER), semantic role labelling (SRL) come under NLP. Some of the most common applications of NLP include chatbots and machine translation. The advent of deep learning in recent years has led to the discovery of convolutional neural networks (CNNs) [2] and recurrent neural networks (RNNs) [3], both of which have demonstrated a significant increase in performance compared to any rule-based and statistical model. Moreover, the development of specialized architectures such as LSTMs and GRUs has led to further advancements in this field. An extensive review of these topics and their recent advancements can be found in [4].

Models such as word2vec and GloVe have been created to obtain a representation of words in a corpus as vectors in a manner that preserves the association between them, and this can be helpful to train machine learning models. Text classification is another area of NLP that aims to classify any given text into several different categories. Examples of this include spam detection for emails and flagging abusive content on social media. Sentiment analysis can provide insight into the subjective meaning of a given text to understand the general sentiment toward a topic. Using a



combination of these techniques and many others, it is possible to construct a robust model for moderating online content.

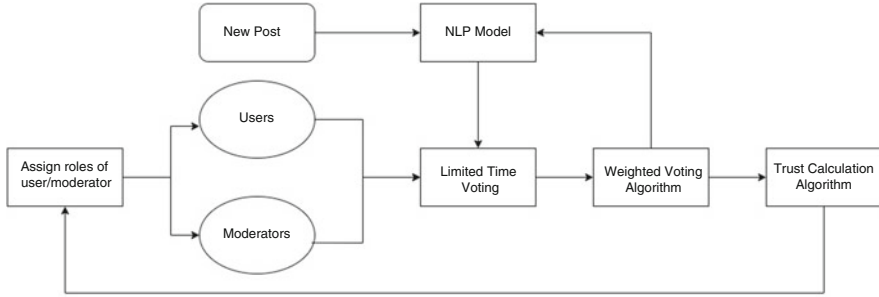
Large-scale social media platforms often employ workers whose sole task is to perform content moderation. These moderators set and enforce rules make tough decisions regarding sensitive content. Their choices are a direct reflection of the platform's philosophy. Sometimes, volunteers and freelancers also assist moderators on such tasks, and some platforms allow users to directly report or flag inappropriate behavior so that it can be scrutinized further. It is also possible for the users to regulate content directly by their overall response in a distributed content moderation system. In fact, this method has demonstrated that it can result in more civil online forums [5].

A similar implementation of trust networks is the reputation system of EigenTrust. EigenTrust is a system based on a peer-to-peer network wherein the calculation of the trust of every user depends on the popularity of their uploads. Such reputation-based trust management systems drastically reduce the probability of malicious content being spread by other users. Trust-based moderation systems have also been explored in decentralized networks, such as TrustNet and PeerTrust. In this system, moderation capacities are only given to those users who have achieved a high level of trust in the community. Trust is calculated using Appleseed which utilizes various metrics in a weighted graph. Implementing the trust management systems of such decentralized networks in a centralized context can prove to be useful in terms of transparency.

Manual moderation of content can be labor-intensive, time-consuming, and mentally detrimental to the moderators, especially as social media platforms grow larger every day. In order to have efficient and scalable moderation, automated techniques have been developed. Many of such models [6–8] are based on NLP and machine learning and have shown promise of replacing thousands of human workers involved in moderating content. Moreover, their influence on the behavior of users has also been studied [9]. The main drawback that most of them suffer is the inability to learn from false negatives since malicious intent can be very nuanced, complex, and subtle. The system proposed in this chapter aims to address this issue with a transparent and trust-based algorithm to provide feedback to the model and allow it to learn and improve its performance continuously.

### 3 Methodology

This section describes the proposed system in detail and goes over the four main components that are involved. As shown in Fig. 1, these components are the guidelines, the NLP model for abusive content detection, the weighted voting algorithm, and the trust calculation algorithm. All users of a social media platform that employs this system have to be informed about regulations on posting content through the guidelines. These regulations are set by moderators, who are the most trusted users on the platform. Each user is associated with a trust rating calculated



**Fig. 1** High-level design

by an algorithm, and only the top percentage of users become moderators. Once the users are aware of the guidelines, they are free to post content. This is first pre-moderated using the NLP model trained to detect if the post is abusive or offensive in any manner. If that is true, the post is rejected, and the user is informed of the same. Otherwise, the post shows up on the platform for all other users to see and judge. Here, users may choose to upvote the post if they feel it is appropriate or downvote it and cite which regulation they think it has violated. The aggregate of the votes given by moderators is used to determine if the post must be removed or not. Once the decision has been made, the trust rating of all voters (not just the moderators) is updated based on whether their vote agrees with the moderators and with each other.

### ***3.1 Guidelines and Standard Protocol***

Guidelines and Standard Protocols must be communicated to all end users of the platform. They should leave no rule for ambiguity and should be strictly enforced on all content posted and sources cited. Some examples of rules to be mentioned in the protocol are:

1. Do not post content that promotes hate, bullying, and threats of violence.
2. Ensure that the content does not target marginalized communities.
3. Do not post or encourage the posting of sexual or suggestive content involving minors.
4. Avoid posting anything illegal according to law.

While voting, if a post has to be rejected, it is necessary to pick the rule from the standard protocol it has violated. This is done in order to main transparency and convey a clear message for why the moderators wanted to remove the posted content.

The reason for only allowing moderators vote to count was to ensure that a situation never arises in the system where a few users could vote in abusive content during a time where most other users are not active.

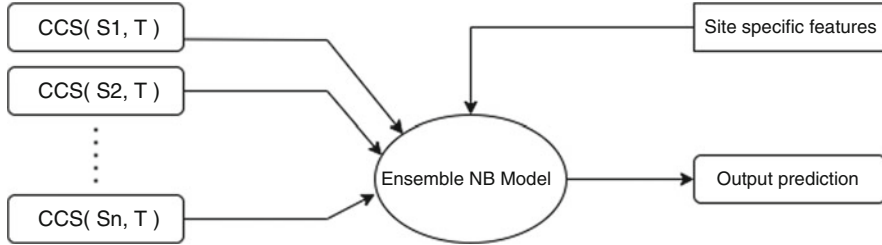
### 3.2 *Initial Moderation Through NLP Model*

Pre-moderation of any posts by users is done through automated models. The primary purpose of this model is to make sure that posts are not being hateful or abusive to any individual or group as described in the guidelines. Posts inferred as offensive are rejected by the system, and the users are informed of the same. If the post is inoffensive, then it is uploaded onto the website. Such models are necessary because a large number of posts may be uploaded on the site every minute, and a self-policing system alone will take a long time to evaluate all of them. At the same time, they are not sufficient on their own since the models may predict a small number of false negatives, i.e., posts that are abusive but were predicted as normal, and these can be moderated by the community efficiently using the algorithms described in Sects. 3.3 and 3.4.

The automated model chosen for the system proposed in this chapter is based on the Bag of Communities (BoC) approach from [10]. This model has the ability to be used off the shelf for any generic social media platform and, more importantly, improves its performance significantly as site-specific abusive content is gradually fed back to train the model. The authors of this paper point out that the addition of human in the loop for such systems shows promise for enhanced detection of abusive content over time. This is why it works well with the self-policing component since it provides false negative feedback.

The main feature of the BoC model is that it relies on the similarity between the target social media platform and existing platforms and forums through a metric called cross community similarity (CCS). To calculate these scores, each post in the corpus of a community is represented as a feature vector. This is based on the bag of word (BoW) approach, where a binary vector of length equal to the vocabulary of the corpus represents the presence or absence of words and phrases. This is simple and effective in capturing the linguistic representation of a post. Using these vectors, a statistical model  $M_S$  is trained to estimate the probability that any given post belongs to a particular community  $S$ . Then, for a given post  $p$  in a community  $T$ , the value of  $CCS(S, T)$  is obtained using  $M_S(p)$ . This is based on Granger causality.  $CCS(S, T)$  is also a value between 0 (entirely dissimilar communities) and 1 (entirely similar communities).

Since there are a large number of diverse communities that already exist, and each one has its own level of moderation, there is the freedom to pick only those communities that are appropriate for the target community to be compared to. These similarity scores are used as features for an ensemble model that also uses site-specific information on posts (such as the words and phrases used in the target community) to classify the post as offensive or clean as shown in Fig. 2. This model



**Fig. 2** NLP model

is dynamic in the sense that it iteratively sees data from the target community and improves its performance through online learning. This datum is used together with the similarity scores to make subsequent predictions.

### 3.3 *Weighted Voting Algorithm*

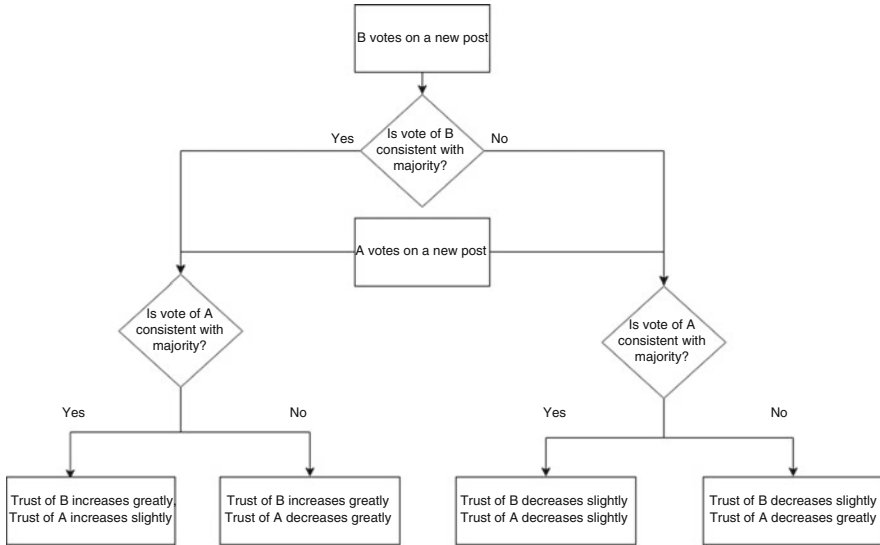
When some new content is posted on the platform, all users will be allowed to vote to accept or reject the content for a certain initial time period. Although only the vote of the moderators will count in the decision-making process, other users are also allowed to cast their vote based on which their trust score can change. Then, the votes of the moderators will be tallied, with each vote weighted according to the trust score. Depending on whether accept or reject is in the majority, the posted content will be kept or removed. Another important requirement is that when someone votes to reject a post, they will have to attach the appropriate rule(s) from the community guidelines which they felt have been violated.

### 3.4 *Trust Calculation Algorithm*

In our social network model, all users will be given a trust rating based on their reputation and previous actions. This trust rating is crucial to the voting process as the votes of each user will be weighted by their trust value.

Each new user will start with a score of 0 and can participate in the voting process for moderating content to improve their trust rating. The reason for making new users start with a 0 trust score is to discourage people from creating new accounts if their trust rating has reduced from the starting score. If the initial score is 0, then there is no advantage (or optimal profit) for being a newcomer.

After the voting is over, we rank all those who voted (both moderators and users) based on their choice and current trust score. So, for example, if the result of a vote was to accept a post, then all those who voted accept will be ranked in descending



**Fig. 3** Effect of Elo-based trust calculations

order of their current trust score, and below them, all those who voted reject will also be ranked in descending order of their current trust score.

We shall apply the Elo Rating formula to compute the change in the user’s trust score. The formula ensures that the “rich keep getting richer” scenario [11] does not occur since it is incrementally harder to keep increasing your score. Moreover, it was necessary that the algorithm satisfies these two conditions to maintain a competitive system where everyone has a fair chance of improving and also mistakes are penalized more the higher rated you are. The conditions are as follows:

1. If user A has worse trust than user B, and if they performed the same or worse, then A’s rating should still be lesser than B.
2. If user A did better than user B, then the change in rating of A should be greater than or equal to B.

For a better understanding of the effect of an Elo-based rating calculations, Fig. 3 can be referred where it is assumed that user A has a greater trust score than user B.

We make some modifications to the standard Elo Rating [12] to apply it in a scenario with more than 2 people. The formula states that if there exist 2 players with rating  $R_1$  and  $R_2$ , then the probability that player 1 does better than player 2 is given by

$$P_{1,2} = 1/(1 + 10^{(R_2 - R_1/400)}) \tag{1}$$

After the voting time period is completed for everyone who participated, we can calculate their expected rank by finding P with every other user who voted. Assume

the person for whom we want to find the expected rank is  $i$  and the total number of participants in the voting process is  $n$ , then the formula will be

$$E_i = 1 + \sum_{j=1}^n P_{j,i}(j \neq i) \quad (2)$$

(The +1 is because ranks are in 1-indexed format)

After voting is over, if a participant (moderator and user) has a better rank than their seed value, then their trust score will increase, else it will decrease. The change in their trust score is equal to the difference between their expected rank and actual rank. Let trust score for user  $i$  be  $T_i$  and their actual rank be  $A_i$ . Then, the change is equal to

$$T_i = T_i + (E_i - A_i) \quad (3)$$

As mentioned earlier, the most trusted users in the platform would be the moderators (assume top 10%). After every rating change, some moderators may get demoted to normal users while some users become moderators. Such a fluid system ensures that power is not permanently given to any person and can always change based on performance.

## 4 Analysis

This section contains the justification and reasoning behind the algorithms that were chosen. The first part contains information about the NLP model chosen and the technique used to train the model and how these choices are suitable for the task that is being addressed. The second part provides a comparative overview of the different types of moderation based on a few key factors.

### 4.1 Justification of NLP Model

A Naive Bayes (NB) classifier is used as the base model in the NLP part of the proposed solution. This is a probabilistic classifier that is based on Bayes theorem and assumes that the different features provided to it are entirely independent. Although this assumption does not always hold true, it is observed that the classifier manages to provide good results in many scenarios.

In the context of the NLP problem of this chapter, the features denote the frequency of occurrence of different words and  $n$  grams in the corpus of the entire dataset. More concretely, if a certain diagram such as “kill yourself” is present 3 times in a certain post, then a corresponding feature  $x_i$  for that post will have value

3. This is called the bag of word (BoW) approach and is often used in conjunction with the Naive Bayes model.

Apart from the site-specific information fed to the classifier through BoW, CCS scores are also provided. As mentioned in Sect. 3.2, these scores are themselves calculated through Naive Bayes models, and this means that the final model is an ensemble classifier. The initial models serve to extract features and the final model uses those features along with additional information to make the final prediction. The success of such an ensemble model in this case is due to the fact that the constituent feature extractors are trained on diverse datasets (each dataset contains posts from one community).

Finally, one of the most important aspects of this model is the fact that it can improve performance significantly with online learning. The feedback loop from the voting algorithm provides false negatives, and these can be used in batches to regularly retrain the model while still retaining information that was used initially. This is crucial since social media platforms are dynamic and constantly evolve with new terms and linguistics being added all the time. A static model will quickly become ineffective in such cases and collecting new data and retraining it from scratch will be time-consuming and resource-intensive. To overcome such difficulties, it is imperative that continual learning is used.

## 4.2 Comparison with Other Moderation Systems

The 3 major categories moderation systems can be classified into are: manual moderation, fully automated moderation, and hybrid moderation (which is a mix of both manual and automated), and these are shown in Table 1. Based on the below factors, it is clear that hybrid moderation gives the best of both worlds without any major drawbacks if well designed. Hence, the type of moderation system chosen for the proposed approach was a hybrid one as it could provide a high level of transparency and accuracy with low bias while not requiring as much manpower and time as a manual system.

**Table 1** Comparison of moderation systems

Factors	Manual	Fully automated	Hybrid
Manpower	High	Zero	Some volunteers
Time consumption	Very high	No time	Requires little time
Accuracy	High	relatively poor	High (keeps improving)
Room for bias	Very high	Extremely low	Extremely low
Transparency	Low	Moderate	High

## 5 Conclusion and Future Work

In conclusion, this chapter presents a self-correcting system of moderation with total transparency. The semi-supervised NLP model that was introduced gets more accurate with each post and interaction with the users. The weighted voting algorithm also gets better as the system gets a more accurate estimation of the trust scores of all the users. With enough users and enough time, the system becomes tamper-proof, and it is extremely difficult for any central party to have any control over moderation of the system. This is the crux of how transparency in content moderation is established. The current algorithm extends Elo Rating to multiple players. In the future, we could look into developing a more dynamic trust calculation algorithm that considers more parameters for updating the trust scores. Another area of improvement is to provide more features like rating cited sources to check if they are accurate and assigning different moderators for each category.

## References

1. J. Tucker, A. Guess, P. Barbera, C. Vaccari, A. Siegel, S. Sanovich, D. Stukal, B. Nyhan, Social media, political polarization, and political disinformation: A review of the scientific literature, SSRN Electronic Journal <https://doi.org/10.2139/ssrn.3144139>.
2. R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, *J. Mach. Learn. Res.* 12 (2011) 2493–2537.
3. J. L. Elman, Finding structure in time, *COGNITIVE SCIENCE* 14 (2) (1990) 179–211.
4. T. Young, D. Hazarika, S. Poria, E. Cambria, Recent trends in deep learning based natural language processing (2018). <http://arxiv.org/abs/1708.02709>.
5. C. Lampe, P. Zube, J. Lee, C. H. Park, E. Johnston, Crowdsourcing civility: A natural experiment examining the effects of distributed moderation in online forums, *Government Information Quarterly* 31 (2) (2014) 317–326. <https://doi.org/10.1016/j.giq.2013.11.005>.
6. J.-Y. Delort, B. Arunasalam, C. Paris, Automatic moderation of online discussion sites, *International Journal of Electronic Commerce* 15 (3) (2011) 9–30. <http://arxiv.org/abs/10.2753/JEC1086-4415150302>, <https://doi.org/10.2753/JEC1086-4415150302>.
7. H. L. Hammer, Automatic detection of hateful comments in online discussion, in: L. A. Maglaras, H. Janicke, K. Jones (Eds.), *Industrial Networks and Intelligent Systems*, Springer International Publishing, Cham, 2017, pp. 164–173.
8. H. Watanabe, M. Bouazizi, T. Ohtsuki, Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection, *IEEE Access* 6 (2018) 13825–13835. <https://doi.org/10.1109/ACCESS.2018.2806394>.
9. S. Jhaver, A. Bruckman, E. Gilbert, Does transparency in moderation really matter? user behavior after content removal explanations on reddit, *Proc. ACM Hum.-Comput. Interact.* 3 (CSCW). <https://doi.org/10.1145/3359252>.
10. E. Chandrasekharan, M. Samory, A. Srinivasan, E. Gilbert, The bag of communities: Identifying abusive behavior online with preexisting internet data, in: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI '17*, Association for Computing Machinery, New York, NY, USA, 2017, p. 3175–3187. <https://doi.org/10.1145/3025453.3026018>.
11. Open codeforces rating system, <https://codeforces.com/blog/entry/20762> (2015).
12. A. Ebtakar, P. Liu, Elo-MMR: A Rating System for Massive Multiplayer Competitions, *Association for Computing Machinery, New York, NY, USA, 2021*, p. 1772–1784. URL <https://doi.org/10.1145/3442381.3450091>



# A New Chaotic-Based Analysis of Data Encryption and Decryption



Fatema Tuj Johora, Alamin-Ul-Islam, Farzana Yesmin,  
and Md. Mosfikur Rahman

## 1 Introduction

Data transmission technology is improving and becoming increasingly capable of managing large amounts of data in a short amount of time. As a result, data security is the most crucial concern to keep these data safe from any flaws or harm. The most common and widely utilized methods for protecting data are encryption and decryption. Symmetric and asymmetric keys are the two types of encryption and decryption. Private and public keys encryption are other names for them. Both the public and private keys are employed in an asymmetric algorithm [1]. Public keys are commonly used in encryption, as are private keys for decoding [2]. AES, AES-256, DES, 3DES, and RSA are some examples of encryption protocols. They are all based on symmetric or asymmetric keys, but some of them combine the two, using public keys for encryption and private keys for decryption. RES is a good example. Position permutation, value transformation, and a combination of both position permutation and value transformation are the three primary encryption techniques. In this research, we propose new text encryption and decryption algorithm that falls under the value transformation category. We focus on generating a random seed for the random keys as a security key for encryption and decryption according to approaches of Hussain et al. [3]. Because of security concerns, any static key

---

F. T. Johora (✉)

Department of Computer Science and Engineering, Green University of Bangladesh, Dhaka, Bangladesh

e-mail: [fatema@cse.green.edu.bd](mailto:fatema@cse.green.edu.bd)

Alamin-Ul-Islam · F. Yesmin · M. M. Rahman

Department of Computer Science and Engineering, Faculty of Science and Information Technology, Daffodil International University, Dhaka, Bangladesh

e-mail: [alamin15-944@diu.edu.bd](mailto:alamin15-944@diu.edu.bd)

is ineffective for encryption and decryption. However, a computer system cannot produce its own random number. However, it may generate a random key using a millisecond value from the system as a random seed. This type of key is used in our encryption and decryption process. Some encryption methods, such as SHA256 and PKDBF2, are used to store user data by encrypting it with a specific salt. Furthermore, these algorithms are irreversible. Apart from user passwords, we are unable to save any other information about users using encryption. Anyone with access to the database can steal a large amount of user data in a short period of time. To save all data in the database, we will need reversible encryption and decryption method. For user privacy and security, data transactions are also critical. As a result, we can employ a strategy that ensures data security by allowing users to transact with one another while preventing the third party from seeing the real data. SALT is already used in existing encryption methods. This SALT can be modified, but we utilized one-time passwords in our technique because they are intimately involved in the encryption and decoding processes. The procedure of creating this one-time password is messy. That it is not like salt in that it is not static. Without those jumbled keywords, the encryption and decryption procedure is impossible. Rathod et al. [4] described the entire process of encrypting and decrypting the chaotic keyword, as well as the efficiency of our solution in terms of key length, algorithm complexity, and the optimal way of attack. We briefly reviewed several sorts of data security challenges, methods and data encryption, and decryption algorithms in this work. Using milliseconds as a random seed, our suggested approach will boost data security and alleviate random seed concerns.

In this research paper, the sections were approached through Motivation where a detail summary will be provided; that is why we need to do this research and what will be its contribution. In Literature Review some of the papers were reviewed competing with this proposed system. With Methodology and Analysis section, this paper concludes with actual output which is mentioned in Experimental Results. Finally this paper concludes with Conclusion.

## 2 Motivation

The number of fast data transmission methods, procedures, and devices is growing these days. It is critical to ensure secure transmission and keep secrets safe. E-commerce, e-governance, social media (e.g., WhatsApp, Twitter, Facebook, Viber, Instagram, LinkedIn), agencies, institutions, and industries all require protection against data loss, theft, and illegal use and alteration. Cryptography may be the finest way for keeping the secret hidden. Data integrity, authentication, and secure transmission are all provided by cryptosystems using encryption methods. As a result, providing security in the data transmission network is crucial. The third world war, according to forecasters, will be a cyberwar. As a result, data licking and theft are typical occurrences. This drove us to learn more about data security methods.

### 3 Literature Review

In terms of data security or cryptosystems, there are three key challenges. The key problem is ensuring data privacy from beginning to conclusion, also, safeguard against any hacking or malware issues. Another option is to locate and eliminate unauthorized access. The most popular encryption methods are Advanced Encryption Standard (AES), Data Encryption Standard (DES), and Rivest-Shamir-Adleman (RSA). Borodzhieva et al. [5] described the software implementation module can be encrypted and decrypted using RSA. Another approach is used by both AES and RSA, and according to Mahalle et al. [6], the combination technique is what it is called. It is critical to access, store, and manipulate information in a secure and non-threatened manner. Data is encrypted before it is used for privacy and control purposes. As a result, providing customers with security services that are also efficient is a significant issue. Several researchers have proposed a variety of schemes and models based on various systems. All of these research publications put in a lot of effort and time to come up with solutions like high scheme efficiency, stateless verification, unlimited usage of queries and data irretrievability, and leveraging DNA sequence for data encryption and decryption according to Pushpa et al. [7]; among other things Reshma et al. [8] present a decentralized multi-authority attribute-based method that is both efficient and effective. Some of them devised a hybrid algorithm that combines the P-AES and RSA algorithms. To protect big data analysis, methods like ARIA are used according to Youngho et al. [9]. Many businesses and academic organizations are investing in this technology since it is changing the working environment in the field of information technology [10–12]. As a result, it is critical to ensure that data is transferred, stored, and owned in a secure manner.

#### 3.1 *Advanced Encryption Standard (AES)*

For its faster speed and easier hardware-software installation, the Advanced Encryption Standard (AES) is the most used data encryption method. Sundor et al. [13] described another benefit of this technique is that it is simple to implement on various platforms, particularly small devices. It is now found in a number of different security systems. Depending on the key size, AES encrypts 128-bit data blocks in ten, 12, or 14 rounds according to Kansal et al. [14]

#### 3.2 *Rivest-Shamir-Adleman (RSA)*

Ronald Rivest, Adi Shamir, and Leonard Adleman proposed the first public key cryptosystem (symmetric) in 1978, according to Koc [15]. There are three steps to this method. These are the following:

- (i) Primary key generating
- (ii) Encryption step
- (iii) Decryption step

## 4 Methodology of CRSA

When we want to encrypt some data, we need a random number or string to carry out an operation based on a specified method. However, how does one get a random number? A random number can be tracked in several ways. The propagation of random numbers is not completely “random.” It is deterministic, and the conformity it generates is a result of the seed value you give a random function:

```

random.seed (109)
x = random.random ()
y = random.random ()
1st random value = 0.27958303860586786
2nd random value = 0.45927897430988984

```

The value 109 is now static. We provide a method for selecting this random seed from our system’s time by millisecond in our algorithm. A second consists of 1000 milliseconds. So, from 0 to 1000, there are 999 numbers. In science, there are only 999 numbers; therefore, getting a random millisecond is difficult. However, if we repeat iterations many times, we will end up with extremely deep random numbers based on the original seed. So here is what you do:

Step 1: As a seed, carry a millisecond number.

Step 2: Run that seed and use the method to generate a new seed.

Step 3: Repeat step 2 iteratively until you get a highly complex random number. The algorithm becomes more difficult as the seed length increases, but it improves overall security. All three-digit numbers from 0 to 999 are available. However, by multiplying some additional random integers, we can increase the number of digits. This method uses a chaotic random number generator as well as specialized data. A random seed generates the random number. And data is the message that one user sends to another:

- (i) A number  $R$  ( $0 < R < 1000$ ) is taken as a seed from milliseconds.
- (ii) Using that seed, a specific number of random digit  $P$  ( $99999 < P < 10^5$ , the range can be decreased or increased) is generated from an algorithm.
- (iii) A message or string is a source  $S$ . Let us assume the first character of the string is  $S_0$  and the last character is  $S_n$ . The length of the string is  $S_1$ .
- (iv) Length of random numbers =  $L$ .
- (v)  $M = \text{ASCII value}(S_n) + \text{ASCII value}(i \% L)$  (where  $0 \leq i \leq S_1$ ).
- (vi)  $Q = \text{Convert } M \text{ into character according to ASCII value.}$

- (vii) Put the value in a JSON.
- (viii) Repeat the steps (5–7) until the operation of  $S_n$ .

Encryption and decryption keys are more important than encryption schemes, which means random numbers are more important. Because when it comes to security, the safety of the key or random number is always our first priority. Using random seeds to generate random numbers is a simple technique. And, because of its real randomization behavior, an attack on this procedure is impossible to succeed.

As a random number, we use a six-digit number. We can generate a total of 900000 units of number. Choosing a random number from 900000 units of numbers has a probability of 1 in 900000.

$$(999999 - 100000) + 1 = 900000$$

$$1/900000 = 0.00000111111$$

However, if we divide a random number into 12 or 24 digits, the likelihood of discovering a random number decreases, as illustrated in Fig. 1.

For probabilistic algorithms, the efficiency with which a particular number of prime numbers can be generated is a critical component. The security of a cryptosystem is determined by the key value, or Seed. For attackers, determining a correct seed among the random number’s 24 digits is nearly impossible. In truth, our process for generating random numbers does not follow any sort of pattern or generate a number, in the same way, every time.

We encrypt data after creating a random number. This is the formula:

$$L = \{x \in L | \text{floor}: x > 0 \} \tag{1}$$

$$E_n = S_n - 1 + \{(n - 1) \bmod L\} \tag{2}$$

And the decryption formula is:

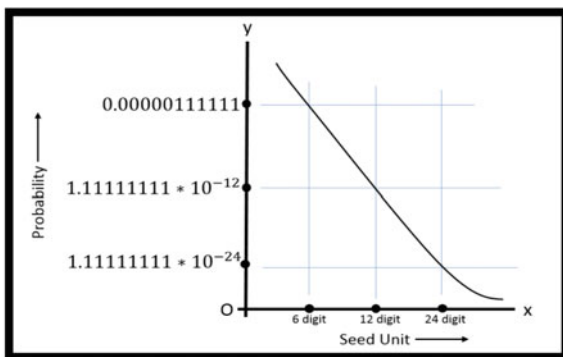


Fig. 1 The probability to find a random number with different a digit

$$D_n = S_n - 1 - \{(n - 1) \bmod L\}$$

For encryption of letter “D”:

$$\begin{aligned} D &= 68 \\ \text{element position in string, } n &= 1 \\ \text{Random seed, } R &= 548691 \\ \text{Seed length, } L &= 6 \\ S_n - 1 &= S_1 - 1 = S_0 = 68 \\ \{(n - 1) \bmod L\} &= \{(1 - 1) \bmod 6\} = R_0 = 53 \\ E_n &= 68 + 53 = 121 = y \\ \text{Chipper text “y”} \end{aligned}$$

For decryption of chipper text “y”:

$$\begin{aligned} \text{Chipper text, } y &= 121 \\ \text{Chipper text position in string, } n &= 1 \\ \text{Random seed, } R &= 548691 \\ \text{Seed length, } L &= 6 \\ S_n - 1 &= S_1 - 1 = S_0 = 121 \\ \{(n - 1) \bmod L\} &= \{(1 - 1) \bmod 6\} = R_0 = 53 \\ D_n &= 121 - 53 = 68 = D \\ \text{Original Text} &= D \end{aligned}$$

Our encryption and decryption algorithm in flowchart is as follows in Figs. 2 and 3.

The entire representation of sending message process using CRSA is shown in Fig. 4 and receiving is shown clearly in Fig. 5.

## 5 Analysis

We put our system through its paces with several units of random keys of varying lengths. A value transformation algorithm is what we have come up with. According to Hussain et al. [16], the majority of cipher text is unrelated to regular alphabets due to its greater ASCII value. As a result, there is no way to decipher encrypted material and discover any secret meaning. The result is given in Table 1.

We simulated it in C++ language to compare its performance to that of other methods. Other algorithms’ performance numbers differed from those of our suggested CRSC (chaotic random seed cryptography). For each algorithm, we track encryption, decryption, throughput, and memory usage [17].

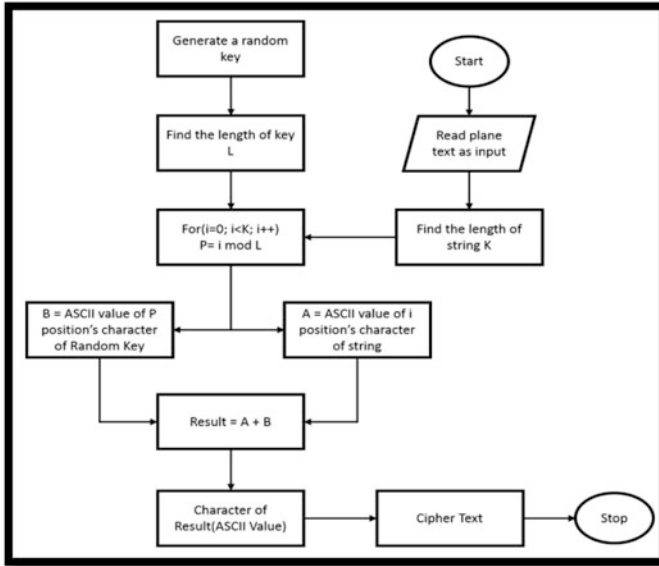
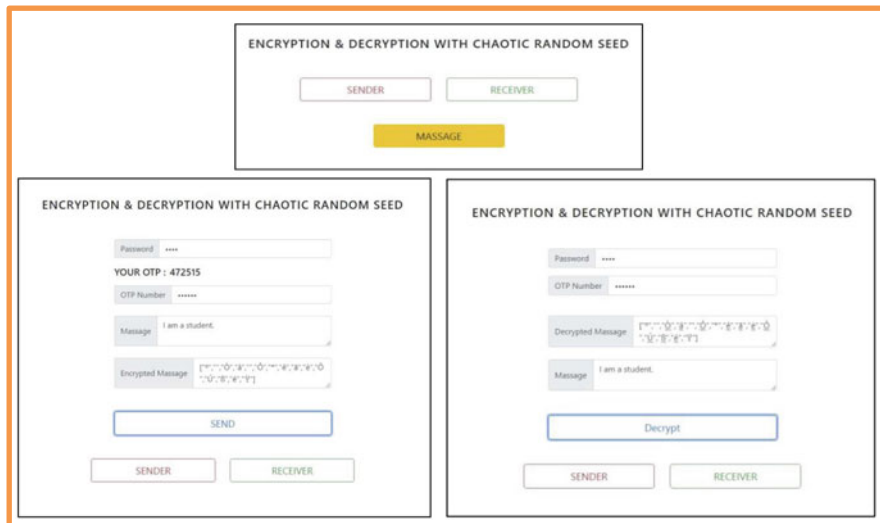


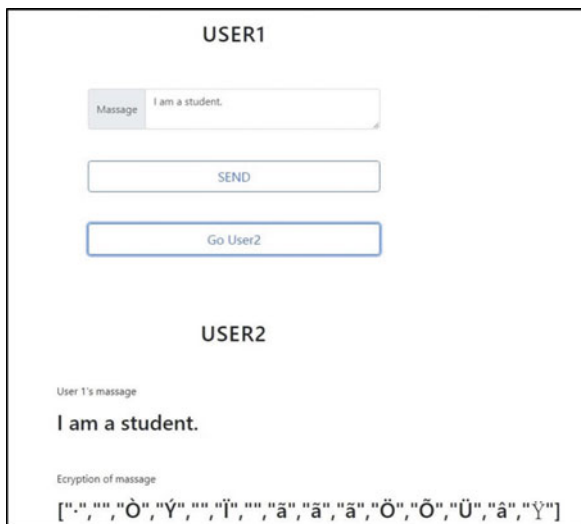
Fig. 2 Encryption process of CRSA



Fig. 3 Decryption process flowchart for CRSA



**Fig. 4** Graphical User Interface (GUI) for encrypting and decrypting a message using CRSA



**Fig. 5** Graphical User Interface (GUI) after sending a message and receiving by USER2 using CRSA

**A. Encryption Time:** The encryption time is the amount of time it takes for an encryption technique to convert plaintext into encoded text. It took milliseconds to measure it.



**Table 1** Output encrypted text of CRSA algorithm

Original text	Key length	Chipper text
Hello World!	6 digit	[“j”, “Ø”, “ä”, “ä”, “ä”, “ä”, “ç”, “ä”, “è”, “ä”, “Ø”, “”]]
	12 digit	[“4”, “×”, “Ü”, “ä”, “ä”, “ä”, “ä”, “ä”, “ä”, “ä”, “×”, “”]]
I am a student.	6 digit	[“o”, “ö”, “ä”, “ä”, “ä”, “ä”, “è”, “ç”, “è”, “Û”, “P”, “B”, “ç”, “F”]]
	12 digit	[“”, “×”, “ä”, “ä”, “ä”, “ä”, “ä”, “ä”, “ä”, “ä”, “×”, “ö”, “Û”, “æ”, “æ”]]
Stop pollution.	6 digit	[“Ä”, “ä”, “ä”, “ä”, “ä”, “ä”, “B”, “P”, “Ü”, “ç”, “è”, “B”, “Y”, “P”, “”]]
	12 digit	[“Ä”, “æ”, “ä”, “è”, “ä”, “ä”, “ä”, “ä”, “ä”, “è”, “ç”, “Û”, “Y”, “ä”, “—”]]

**Table 2** Requirements for testing different texts of different sizes

Algorithms	Test data	Performance	Configuration of hardware
DES (Data Encryption Standard)	Text file (2mb, 1mb, 50kb)	Measure encryption time, decryption time, throughput, and memory utilization	Intel(R) Core(TM) i3-4150 CPU @ 3.50GHz
AES (Advance Encryption Standard)			
CASA (Chaotic Random Seed Algorithm)			

- B. Decryption Time:** The decryption time is the amount of time it takes for an algorithm to reproduce an original plaintext from its garbled text. It took milliseconds to measure it.
- C. Throughput:** The speed of the complete encryption process is referred to as throughput in the encryption and decryption processes. The throughput-throughput equation is a mathematical formula that describes how much work is done:

$$\text{Throughput} = Tp \text{ (Kilobytes)} / Et \text{ (Second)}$$

where  $Tp$  = Total plain text (Kilobytes)

$Et$  = Encryption Time (Second)

- D. Memory Utilization:** Memory utilization refers to how much memory the encryption and decryption processes take up. In kilobytes, we measured it. Table 2 describes the simulation scenario.

## 6 Experimental Results

After analysis, the simulation demonstrates that our approach takes 98 milliseconds less encryption time than AES and 254 milliseconds less encryption time than the DES algorithm for various text sizes. Our approach requires 87 milliseconds less time to encrypt a 1MB text file than AES and 880 milliseconds less time than the DES algorithm. Our approach requires 102 milliseconds less time to encrypt a 2MB text file than AES and 1241 milliseconds less time than the DES algorithm in Fig. 6.

Our approach decrypts a 50kb text file in 129 milliseconds, compared to 97 milliseconds for AES and 97 milliseconds for DES. Our algorithm requires 20 milliseconds less than AES to decode a 1MB text file and 318 milliseconds less than the DES algorithm to decrypt a 1MB text file. Our technique takes 23 milliseconds less than AES to encrypt 2MB text files and 591 milliseconds less than DES to decrypt them as shown in Fig. 7.

CRSA (3.5 KB/s) has a higher throughput than both DES (1.87 KB/s) and AES (2.3 KB/s) for a 50kb file. CRSA throughput (2.37 KB/s) is higher than AES (1.85 KB/s) but lower than the DES algorithm (6.4 KB/s) when a 1MB text file is used as a comparison. It delivers feedback for the 2MB file, such as a lower throughput value (2.2 KB/s) than the DES algorithm (10 KB/s) and a higher throughput value (1.8 KB/s) than AES as Fig. 8. illustrated.

The graph of CRSA is shorter than the DES algorithm (9512 kb) and also shorter than the (1053 kb) AES method for a 50 kb file, according to the memory consumption comparison. When using 1 MB of data, the CRSA method requires 2235KB of RAM, 11500KB for DES, and 3747KB for the AES algorithm as shown in Fig. 9.

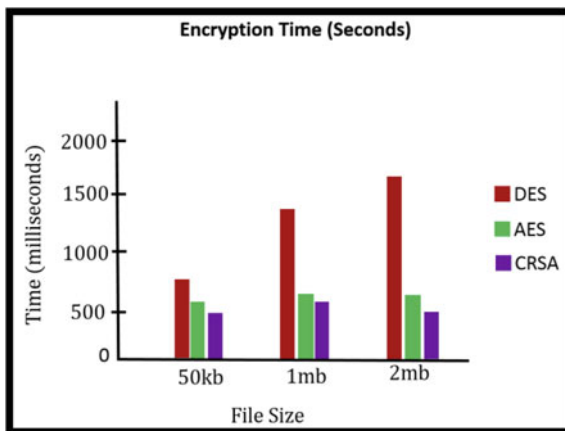


Fig. 6 Encryption time comparison for different algorithms



Fig. 7 Decryption time comparison for different algorithms

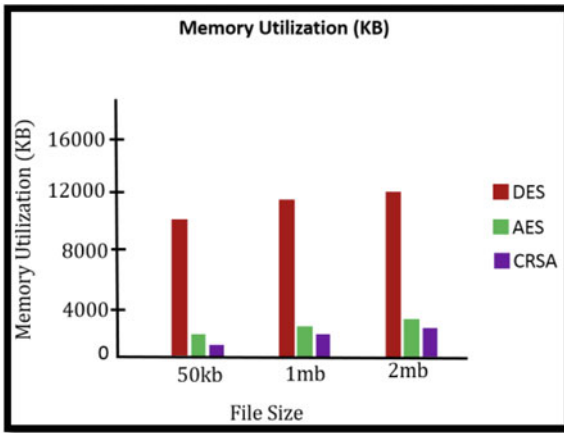
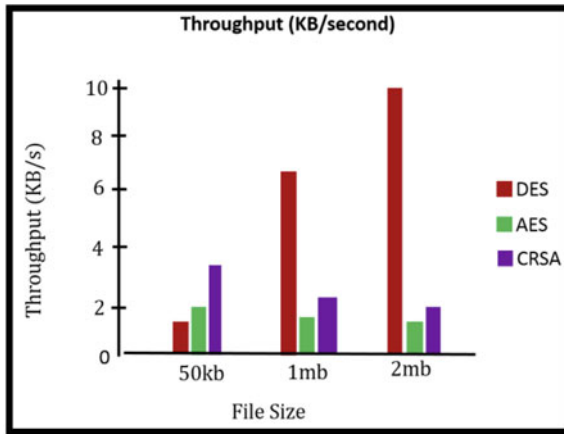


Fig. 8 Memory utilization comparison for different algorithms

The statistics reveal that our suggested CRSA algorithm outperforms both DES and AES. When comparing the AES and DES algorithms, our security is superior to both these. As a result, we have high hopes for CRSA.

## 7 Conclusion

A security key, also known as a secret key, is essential for data encryption and decryption. The computer is unable to produce a secret key on its own. For encryption and decryption, we must provide our own key. However, there is a



**Fig. 9** Throughput comparison for different algorithms

possibility that the secret key will be leaked from us. That is why we require a one-of-a-kind key generated by a machine that is not foreseeable by hackers. So, we used milliseconds in computer systems and added various forms of security keys to create a very unique key known as a random seed. Every device and time, the random seed is completely random. We prioritized the generation of a random seed over encryption and decryption techniques. Because if the key is insecure or predictable, the entire security mechanism is jeopardized. Our approach for generating random seeds, encryption, and decryption takes a little longer than AES and similar algorithms, but it is significantly more secure than other algorithms. For solely text files, we used a random seed-generating procedure and an encryption-decryption process. The comparison with various methods for a text file is shown in our experimental results. This is a stumbling block for me. However, the random seed can be used with various sorts of data, such as images, audio, and so on. As a result, random seeds will be used in the encryption and decryption of image and audio files in the future.

In the future, we will endeavor to strengthen security by using image and audio data types. Our algorithm already outperforms DES and is on par with the AES algorithm in terms of efficiency. We intend to develop it in the future so that it can outperform AES and RES algorithms. Then we will put it into a cloud-based security solution. Cloud computing has become increasingly popular in recent years. The Internet is used to transform data in cloud computing. Cloud computing security is also improved by encryption and decoding.

**Acknowledgment** This work was supported in part by the Center for Research, Innovation, and Transformation of Green University of Bangladesh.

## References

1. Hasib, Abdullah Al; Haque, Abul Ahsan Md. Mahmudul (2008). [IEEE 2008 Third International Conference on Convergence and Hybrid Information Technology (ICCIT) – Busan, Korea (2008.11.11-2008.11.13)] 2008 Third International Conference on Convergence and Hybrid Information Technology – A Comparative Study of the Performance and Security Issues of AES and RSA Cryptography, 505–510. <https://doi.org/10.1109/iccit.2008.179>
2. Carlson, Jay Alan “Method for secure communication using asymmetric and symmetric encryption over insecure communication” 2019 Patent No: US 10,187,361 B2
3. Hussain, Iqra; Negi, Mukesh Chandra; Pandey, Nitin (2018). [IEEE 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) – Noida, India (2018.8.29-2018.8.31)] 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) – Proposing an Encryption/ Decryption Scheme for IoT Communications using Binary-bit Sequence and Multistage Encryption, 709–713. <https://doi.org/10.1109/ICRITO.2018.8748293>
4. Rathod, C., Gonsai, A. (2022). A Detailed Comparative Study and Performance Analysis of Standard Cryptographic Algorithms. In: Khanna, K., Estrela, V.V., Rodrigues, J.J.P.C. (eds) Cyber Security and Digital Forensics. Lecture Notes on Data Engineering and Communications Technologies, vol 73. Springer, Singapore. [https://doi.org/10.1007/978-981-16-3961-6\\_26](https://doi.org/10.1007/978-981-16-3961-6_26)
5. Borodzhieva, Adriana Naydenova (2016). [IEEE 2016 XXV International Scientific Conference Electronics (ET) – Sozopol, Bulgaria (2016.9.12-2016.9.14)] 2016 XXV International Scientific Conference Electronics (ET) – Software implementation of a module for encryption and decryption using the RSA algorithm. 1–4. <https://doi.org/10.1109/et.2016.7753464>
6. Mahalle, Vishwanath S; Shahade, Aniket K (2014). [IEEE 2014 International Conference on Power Automation and Communication (INPAC) – Amravati, India (2014.10.6-2014.10.8)] 2014 International Conference on Power, Automation and Communication (INPAC) – Enhancing the data security in Cloud by implementing hybrid (Rsa & Aes) encryption algorithm. 146–149. <https://doi.org/10.1109/INPAC.2014.6981152>
7. Pushpa, B R (2017). [IEEE 2017 International Conference on Intelligent Computing and Control (I2C2) – Coimbatore, India (2017.6.23-2017.6.24)] 2017 International Conference on Intelligent Computing and Control (I2C2) – A new technique for data encryption using DNA sequence., 1–4. <https://doi.org/10.1109/I2C2.2017.8321834>
8. Reshma, R.S., Anjusha, P.P., Anisha, G.S. (2022). Implementing the Comparative Analysis of AES and DES Crypt Algorithm in Cloud Computing. In: Smys, S., Bestak, R., Palanisamy, R., Kotuliak, I. (eds) Computer Networks and Inventive Communication Technologies. Lecture Notes on Data Engineering and Communications Technologies, vol 75. Springer, Singapore. [https://doi.org/10.1007/978-981-16-3728-5\\_24](https://doi.org/10.1007/978-981-16-3728-5_24)
9. Youngho Song; Young-Sung Shin, Miyoung Jang; Jae-Woo Chang, (2017). [IEEE 2017 IEEE International Conference on Big Data and Smart Computing (BigComp) – Jeju Island, South Korea (2017.2.13-2017.2.16)] 2017 IEEE International Conference on Big Data and Smart Computing (BigComp) – Design and implementation of HDFS data encryption scheme using ARIA algorithm on Hadoop., 84–90. <https://doi.org/10.1109/BIGCOMP.2017.7881720>
10. K. Rani and R. K. Sagar, “Enhanced data storage security in cloud environment using encryption, compression and splitting technique,” 2017 2nd International Conference on Telecommunication and Networks (TEL-NET), 2017, pp. 1–5, <https://doi.org/10.1109/TEL-NET.2017.8343557>.
11. S. Kaushik and C. Gandhi, “Cloud data security with hybrid symmetric encryption,” 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT), 2016, pp. 636–640, <https://doi.org/10.1109/ICCTICT.2016.7514656>.
12. Shetu, S. F., Saifuzzaman, M., Moon, N. N., & Nur, F. N. (2019). *A Survey of Botnet in Cyber Security*. 2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT). <https://doi.org/10.1109/icct46177.2019.8969048>.

13. Sunder, A., Shabu, N., Remya Nair, T. (2022). Securing Big Data in Hadoop Using Hybrid Encryption. In: Karuppusamy, P., Perikos, I., García Márquez, F.P. (eds) Ubiquitous Intelligent Systems. Smart Innovation, Systems and Technologies, vol 243. Springer, Singapore. [https://doi.org/10.1007/978-981-16-3675-2\\_39](https://doi.org/10.1007/978-981-16-3675-2_39)
14. Kansal, S., & Mittal, M. (2014). Performance evaluation of various symmetric encryption algorithms. 2014 International Conference on Parallel, Distributed and Grid Computing. <https://doi.org/10.1109/pdgc.2014.7030724>
15. Koç, Ç.K., Özdemir, F., Ödemiş Özger, Z. (2021). Rivest-Shamir-Adleman Algorithm. In: Partially Homomorphic Encryption. Springer, Cham. [https://doi.org/10.1007/978-3-030-87629-6\\_3](https://doi.org/10.1007/978-3-030-87629-6_3)
16. Hussain, Iqra; Negi, Mukesh Chandra; Pandey, Nitin (2018). [IEEE 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) – Noida, India (2018.8.29-2018.8.31)] 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) – Proposing an Encryption/ Decryption Scheme for IoT Communications using Binary-bit Sequence and Multistage Encryption., 709–713. <https://doi.org/10.1109/ICRITO.2018.8748293>
17. Panda, M. “Performance analysis of encryption algorithms for security,” 2016 *International Conference on Signal Processing, Communication, Power and Embedded System (SCOPE5)*, 2016, pp. 278–284, <https://doi.org/10.1109/SCOPE5.2016.7955835>.

# Trust and Identity Management in IOT



Lakshmi Aashish Prateek, Riya Shah, Alan Tony, and B R Chandavarkar

## 1 Introduction

IoT (Internet of Things) has become very ubiquitous and is found in devices like autonomous cars, home assistance systems, sensor networks in industries, and others. It is projected that the total installed base of IoT devices will be 30.9 billion units worldwide by 2025 [1]. More than 872 million IoT cyberattacks have been recorded in the past few years [2], mainly owing to the fact that more and more functionalities are becoming digitized. Hence, parameters such as privacy and security should be given top priority.

The root concept behind IoT is the reduction of human–computer as well as human–human interactions in order to improve the efficiency, speed, and accuracy of tasks. An IoT cluster consists of Internet-enabled smart devices that have various sensors and transceivers to process, send, and receive the data that are collected by them, a gateway, or any similar edge device which may allow for edge computation, or else, the data is sent directly to the cloud for analyzing. The IoT devices are also functionally capable of communicating with other IoT nodes present in the cluster and can act on the information exchanged between each other. Hence, a breach in the security of these devices can have catastrophic consequences not just on the device that has been breached but also can potentially hamper the working of the other devices present in the cluster which interact with the infected node. As a result of that, a breach in security and privacy of the user’s data is observed. Hence, before establishing any sort of communication between the IoT devices, it becomes imperative for the identity of the devices to be validated as well as trust to be established between the devices. An Identity Management System

---

L. A. Prateek · R. Shah · A. Tony (✉) · B. R. Chandavarkar  
National Institute of Technology Karnataka Surathkal, Computer Science & Engineering,  
Mangalore, Karnataka, India

helps in ensuring only the authorized users have access to the information. This chapter studies the factors that determine the quality of an Identity Management System (IdMS), such as security, scalability, and interoperability and discusses their significance in an IoT IdMS framework. In most IdMS implementations, trade-offs have to be made between the parameters depending on the situations and environment of deployment, as highlighted in [3]. Since sensitive information is being shared around by IoT devices, it is important to manage who or what device accesses the data. So, the need for a proper trust management framework arises. Najera and Lopez [4] and Najib and Sulisty [5] stress on the importance of trust establishment before communication in order to ensure privacy and integrity of the users' data. In this chapter, a hybrid-based trust management framework is presented that deals with the computation of trust between IoT nodes present in the cluster. This approach makes use of direct trust computation, indirect trust computation, and gateway computation. Gateway computation is used in situations where direct trust, as well as indirect trust, might yield inaccurate feedback of a node due to certain factors.

A proposal for a model is made for encrypted secure secret session key (Ks) exchange and identity verification, as trust establishment between nodes is not enough to ensure security and privacy. The proposed model aims to make the communication/data exchanged between the IoT devices and between IoT nodes and the gateway encrypted using the session key(Ks) in order to ensure the privacy and security of the data flowing between the IoT devices in the cluster. This model also ensures that before availing services/data from other nodes, the identity of the nodes involved in the communication is validated and verified in order to prevent common and typical attacks such as “man in the middle attack,” “relay attack,” and also the “Replay attack.” It is also proved that the proposed model is resistant to such attacks.

The remainder of this chapter is organized as follows. Section 2 presents the scope, challenges, and related work carried out in identity management, trust management, and the cryptographic exchange of secret session keys and identity verification. Section 3 presents various parameters that need to be taken into account for an IdMS framework and discusses their significance for the same, while also discussing the interdependence of an IdMS system and a trust management model. Section 4 discusses Che et al.'s approach toward trust management and attempts to improvise on the model by proposing an additional method of trust computation by introducing a gateway in order to overcome the shortcomings of the framework in [6]. Section 5 proposes a framework for identity validation and secure cryptographic exchange of secret session keys between devices part of the IoT cluster. The final section, Sect. 6, presents conclusions and future work to be carried out.



## 2 Related Work

### 2.1 Identity Access Management (IdMS) in IoT

IdMS applications in the field of IoT are still in its initial stages relative to other fields; there are quite a few challenges associated with it. The lifetime of IoT devices often exceeds the timeframe of a few decades. Thus, the need for an IdMS framework that takes into account parameters such as key sizes for encryption, protocols for authorization, and algorithms is required. It is also essential to take into account the scope for technology in the coming future decades, such as cryptanalysis techniques, new types of attacks, computational prowess for brute force techniques, etc. while planning for the parameters mentioned in order to ensure that the IdMS framework can function hassle-free for decades to come.

There are many innovative solutions designed in this domain, e.g., Suriadi, et al. [7] propose an improvement to Federated Single Sign On (FSSO) systems, which is a technology that enables users to log into different services without having to use separate login credentials for each service. Their work has ensured requirements like communication security, minimal data disclosure, and accountability. But the model lacks a secure data storage mechanism, whereas a permissioned blockchain-based approach that enhances the security of One-Time Codes in SMS messages has been introduced in [8].

The bandwidth of the network is also a crucial factor in an IoT environment as heavyweight data transfer is imminent in an IoT environment which puts a strain on the network bandwidth and creates a limitation on the potential of an IoT framework. Addressing this issue, Shah et al. [9] propose a framework targeted at wireless devices in MANETs (Mobile Ad Hoc Networks), which has taken into consideration bandwidth limitations and privacy.

Thus, keeping in mind the limitations of current IoT devices such as computational power, hardware limitations like data storage while also taking into account the scope of technology in the upcoming decades, the variety of applications of IoT devices, and the types of services they offer; IdMS systems must be designed using lightweight and efficient algorithms. Efficient data storage mechanisms must be looked into to handle all the data collected by the devices and also ensure that no attacker has access to the identities of the IoT devices, which could be used for malicious purposes.

### 2.2 Trust Management in IoT

The term “trust in IoT” refers to the study of the behavior of devices that are connected to the same network. The future interactions of the devices depend upon the amount of trust between them and so trust management plays a very crucial role in IoT. Various security concerns in the IoT environment are discussed in [4], and

therefore the need to establish trust between IoT devices is essential. The authors in [4] continue to say that security, privacy, and identity management are all important trust elements in the IoT ecosystem. Quality of service (QoS) and social trust are essential parameters to compute trust [5]. Many of the trust algorithms evaluate trust mainly through self-observation or recommendations from other IoT devices.

One of the biggest challenges in the development of a trust management model is that trust management in IoT devices should be very analogous to the patterns humans follow to trust other humans, such as history with the other person, the personality of the person, and recommendations from other people. Hence, the same pattern of human reasoning should be applied in order to establish trust between IoT devices.

A framework for a scalable trust management system is proposed [10] using a fuzzy approach that resembles human reasoning. Trust of devices, experience, classification of device, and recommendation are the parameters that are used for the trust score of devices. Results indicated that the throughput and packet delivery ratio was consistently good. The energy consumption was also lesser on average than in most systems.

Another significant barrier on the path to establish trust is that the IoT environment is subject to constant changes such as the addition of new devices, services, etc. Hence, a trust management framework must account for changes to the environment dynamically.

A Bayesian-based approach [11] is used to build a scalable and survivable trust management system by utilizing parameters such as changes to community interest and already existing trust-related information. This approach showed good results in achieving dynamic adaptability and good accuracy and convergence behavior for a newly added IoT node. This framework did not account for a cloud environment of IoT devices like gateways and edge computing.

A trust algorithm based on encryption and authentication was shown in [12], which relied on trust evaluation in which the sensors are considered anonymous. The work proposes a Light and Trust Anonymous Mutual Authentication Algorithm (LTAMA) for IoT. The algorithm design is based on identity-based encryption, Private Key Generator, and Elliptic Curve Cryptography (ECC). This approach satisfies the most security requirements such as anonymity of sensors, trust between communicated sensors, and privacy of data.

Thus, the design of a trust management model involves taking into account several factors such as scalability, lightweight computation, and dynamic adaptability. Scalability plays a considerable role, as trust management is not necessarily restricted to one IoT cluster. Trust management should also provide a model wherein IoT devices from one cluster can avail services from another IoT cluster associated with another gateway.

### ***2.3 Identity Verification, Validation, and Secure Session Key Exchange in an IoT Cluster***

The establishment of trust between two IoT devices now allows the IoT devices to avail services/information from one another. The process of communication between these two devices must be as secure as possible such that no third party (attacker) can gain access to alter or harm the integrity of the data being exchanged. It should also be taken care that the devices verify and validate each other's identities before exchanging secret session keys, which would otherwise put the data of their respective clients at risk.

A space-efficient, authenticated key exchange model for an IoT environment for key exchange between an IoT device and the gateway was proposed by Rabiah et al. [13]. The model discards the symmetric session key at frequent intervals, and data hashing is done based on predefined intervals of time. This framework has not accounted for the encrypted communication of the session key between two or more IoT devices, as an IoT device can take the services of another IoT device and interact with it.

An approach was proposed by Rabiah et al. [14] that makes use of a Physically Unclonable Function (PUF) Challenge Response Pair (CRP)-based mutual authentication, which is a two-pass key exchange protocol. It uses a challenge-based response mechanism for unique device identification. Simultaneously, it also maintains security in communication and authentication. This approach deals with security in communication only between an IoT node and the server. It does not show a framework for communication between gateways and nodes, as well as secure communication between two IoT nodes themselves.

Further innovative applications and development of encryption algorithms must be sought to ensure the utmost security and privacy of the data exchanged in an IdMS framework, simultaneously ensuring the scalability of the framework. Other aspects being considered are resistance to brute force attacks by accounting for the development of computational power in the next few decades and the latest and near future cryptanalysis techniques.

## **3 Concepts in Identity Management**

Identity management is referred to as the set of policies and technologies adopted to safeguard systems from unauthorized access. It ensures that only authorized entities (any node in the network) can access a particular piece of information.

Modern trust management systems should not be dependent on a third party, which can pose security risks. Moreover, a centralized IdMS is prone to a single point of failure. Large amounts of data can be stolen by attackers. There are various challenges faced by IdMS like security, scalability, and interoperability. Often there are trade-offs between each requirement. IoT devices consist of any device with

computing, sensing, or communicating capabilities [3], and these devices need not be directly operated by human beings. Identities of heterogeneous devices cannot be handled by traditional IdMS techniques.

### ***3.1 Characteristics of an IdMS***

Discussed below are integral factors to be taken into consideration while designing an Identity Management System (IdMS):

#### **3.1.1 Security**

IdMS should shield IoT devices from these attacks, which could be either active or passive. The identity of an entity should remain private to other non-desirable entities. In other words, entities should be able to go into an anonymous mode if the application theoretically allows it (like Blockchain Enabled EVS, Cryptocurrency, etc.). On the other hand, IdMS should eliminate loopholes that can be exploited by attackers to perform identity spoofing.

#### **3.1.2 Scalability**

As the number of standalone IoT devices, clusters, and superclusters increases, the requirements posed to the edge also increase. The network capacity should be able to accommodate multiple devices. The bandwidth and latency should be low as the information passed by many of the IoT devices is time-critical. One should also consider factors like device operating systems and software infrastructure.

#### **3.1.3 Interoperability**

Along with the number, even the types of IoT devices are on the rise. There are a rising number of use cases deploying heterogeneous devices. Hence, it is inevitable that the IdMS that is employed should support all these types of devices and not have separate IdMS for each type of IoT device. There is very little research that aims to implement an IdMS for a heterogeneous IoT ecosystem.

Although there are a lot of proposed methods, IdMS in IoT is a new field and there are relatively fewer implementations compared to traditional solutions. There are also trade-offs in existing implementations between various factors like security, scalability, and interoperability. Therefore, the IdMS systems have to be fine-tuned to suit a particular application.

### 3.2 *The Interdependence of Trust Management and IdMS*

Trust establishment and IdMS are not two independent frameworks in an IoT environment. They are closely related and dependent on each other to ensure the utmost safety, privacy and integrity of data, information, and devices in an IoT environment. An IdMS system ensures that only devices which are validated and authenticated can be a part of the IoT environment. It is responsible for managing the IoT devices' credentials in a safe, secure, and scalable. A trust management model is responsible for providing parameters and algorithms for trust computation between devices. An IdMS framework cannot function independently as validation and verification of identities is not enough to provide foolproof security and integrity of data. A scenario is assumed wherein all the devices in the IoT cluster are validated, and their credentials are verified. One of the nodes in the cluster has become malicious (e.g., botnet) or unreliable (e.g., damaged sensor). This device is already verified and validated. But it might take part in incorrect transactions with other devices. Unless this is not accounted for, bad transactions will continue to take place. With the presence of trust management, the malicious nodes can be removed from the cluster, and the faulty nodes can be serviced. Each node can transact with reliable nodes based on the computed trust value. Thus, in a dynamically changing IoT environment, it is essential to design a framework that makes use of an IdMS as well as a trust management system.

## 4 Trust Management Framework

This section gives an overview of trust and discusses Che et al.'s [6] trust management framework. To overcome the drawbacks of that method, another trust management framework is proposed.

IoT devices interact with each other on a huge scale, so to share information securely a certain guarantee is needed that the device with which communication is carried out is a trusted one. Trust gives a certain level of assurance that the device will act as it is directed to be. So each IoT framework should have an effective trust management approach to protect it from malicious attacks and, as a result, ensure security and reliability. The trust value based on its previous behavior can be used as a factor for deciding whether to engage with the device or not.

### **Characteristics of trust:**

1. Trust is dynamic: The trust between devices is subject to change based on their behavior. If Device A trusts Device B today, it is not a necessity that it will trust Device B in the future.
2. Trust is subjective: It is dependent on the point of view of the trustor. If device A trusts device B, it is not a necessity that device C will also trust device B.
3. Trust is asymmetric: If Device A trusts Device B, it should not be concluded that Device B also trusts Device A.

The two main sources of trust proposed by researchers [5] till now are:

**Direct trust:** It represents a value that is calculated based on the previous interaction between the two devices. To calculate the direct trust between Node B and Node C, the number of previous successful transactions ( $\alpha_{BC}$ ) and the number of failed transactions ( $\beta_{BC}$ ) between the devices are taken into account.

$$D_{BC} = (\alpha_{BC} + 1) / (\alpha_{BC} + \beta_{BC} + 2) \quad (1)$$

**Indirect trust:** It represents a value that is based on the recommendation of other devices about the target device.

Suppose Node B wants a recommendation for Node C, then Node B requests all the nodes that are the common neighbors of Node B and Node C. Let  $1, 2, 3, \dots, x, n$  be the common neighbors of B and C and Node B only requests them about the recommendation for Node C since they are the nodes which Node B trusts, and they have interacted with Node C a decent amount of times to give recommendation about Node C. Say Node B requests Node  $x$  to send recommendations about Node C. Then, Node  $x$  sends its direct observation records ( $\alpha_{xC}, \beta_{xC}$ ) to Node B, and then to get the indirect trust it combines it with its own values ( $\alpha_{Bx}, \alpha_{Bx}$ ) about node  $x$ . It is described by the below equation [6]:

$$R_{BC}^x = (R^x \alpha_{BC} + 1) / (R^x \alpha_{BC} + R^x \beta_{BC} + 2) \quad (2)$$

If there are  $n$  recommended nodes, there is a need to combine the recommendations of all the nodes to get the final indirect trust. Since it cannot be guaranteed that all nodes will be honest, the weights of different recommendations are considered. These weights for different indirect trust values are distributed using the entropy theory [6]. Final indirect trust is given by the equation:

$$R_{BC} = \sum_{x=1}^{x=n} (W_x * R_{BC}^x) \quad (3)$$

According to [6], the total trust involves both indirect trust and direct trust, and to make the algorithm as lightweight as possible, indirect trust is used only when the direct trust is not credible enough. If the confidence level of direct trust is greater than the threshold, then direct trust is taken as the total trust, otherwise it should be calculated using both direct and indirect trust as shown by the equation given below:

$$T_{BC} = D_{BC}, \text{ if } \gamma \geq \gamma_0 \quad (4)$$

( $\gamma$  is the confidence and  $\gamma_0$  is the minimum confidence threshold of Direct trust), else:

$$T_{BC} = w_D * D_{BC} + w_R * R_{BC} \quad (5)$$

### 4.1 Proposed Improvisation of the Above Method for Trust Calculation

The drawback of the above approach is that if there are not enough common neighbors between B and C, then the values of indirect trust calculated would not be credible enough. To solve this issue, this chapter proposes an improvisation to the above method which includes the Gateway, and hence it becomes a hybrid based trust management system. So when the indirect trust is not credible enough, Node B requests for the recommendation of Node C from the Gateway. The Gateway is connected to each node in the network. The Gateway contains the recommendation about each node from its neighbors. As and when an interaction takes place between the neighbors, they update their observation records ( $\alpha, \beta$ ) about that particular node on the Gateway. Suppose you need a recommendation about Node k from the gateway. Say 1,2,3,...i,...,n are the neighbors of k. Figure 1 illustrates the

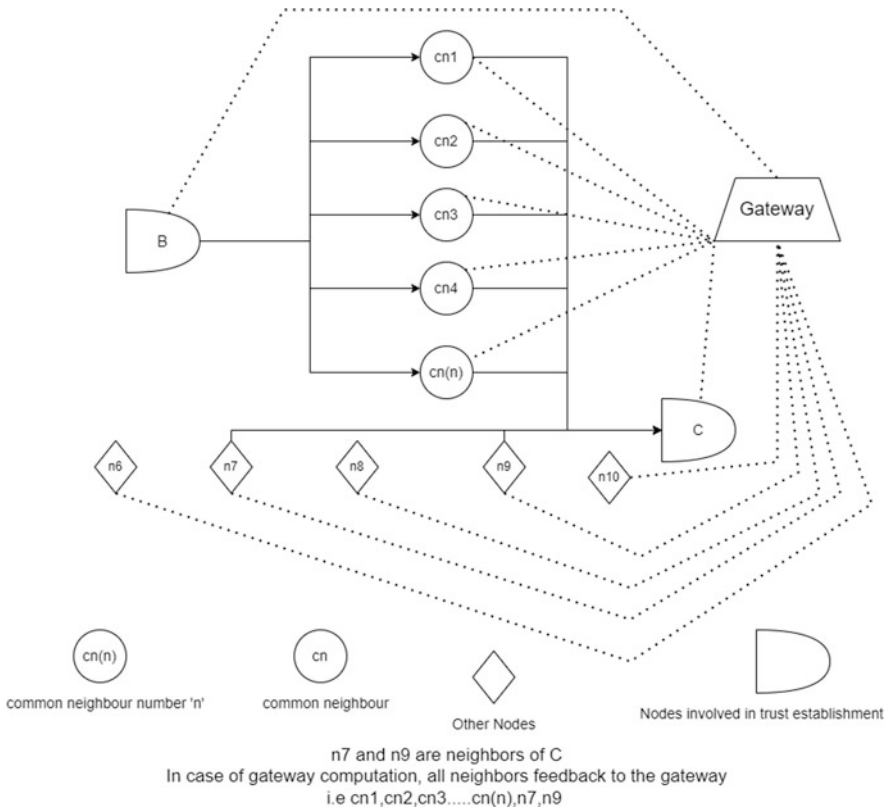


Fig. 1 Architecture for the improvised model where gateway is connected to every device

improvised model involving the gateway which relies upon neighbors of C to provide feedback for Node C

$$R'_{ik} = (\alpha_{ik} + 1) / (\alpha_{ik} + \beta_{ik} + 2) \quad (6)$$

Overall, recommendation for Node k can be calculated by the formula:

$$R'_k = \sum_{i=1}^{i=n} (w_i * R_{ik}) \quad (7)$$

The weights here are distributed based on the entropy theory. To calculate the total trust between B and C, the equation given below is used:

$$T_{BC} = D_{BC}, \text{ if } \gamma \geq \gamma_0 \quad (8)$$

( $\gamma$  is the confidence and  $\gamma_0$  is the minimum confidence threshold of Direct trust), else:

$$T_{BC} = w_D * D_{BC} + w_R * R_{BC}, \text{ if } \delta \geq \delta_0 \quad (9)$$

( $\gamma$  is the confidence level and  $\gamma_0$  is the minimum confidence threshold of Indirect trust), else:

$$T_{BC} = w'_D * D_{BC} + w'_{R'} * R'_C \quad (10)$$

( $R'_C$  is the recommendation that is obtained from the gateway)

The weights of trust used in the three equations are calculated based on the entropy values as shown below:

$$H(D_{BC}) = -D_{BC} \log 2 D_{BC} - (1 - D_{BC}) \log 2 (1 - D_{BC}) \quad (11)$$

$$H(R_{BC}) = -R_{BC} \log 2 R_{BC} - (1 - R_{BC}) \log 2 (1 - R_{BC}) \quad (12)$$

$$H(R'_C) = -R'_C \log 2 R'_C - (1 - R'_C) \log 2 (1 - R'_C) \quad (13)$$

$$w_D = (1 - H(D_{BC}) / \log 2 D_{BC}) / ((1 - H(D_{BC}) / \log 2 D_{BC}) + (1 - H(R_{BC}) / \log 2 R_{BC})) \quad (14)$$

$$w_R = (1 - H(R_{BC}) / \log 2 R_{BC}) / ((1 - H(D_{BC}) / \log 2 D_{BC}) + (1 - H(R_{BC}) / \log 2 R_{BC})) \quad (15)$$



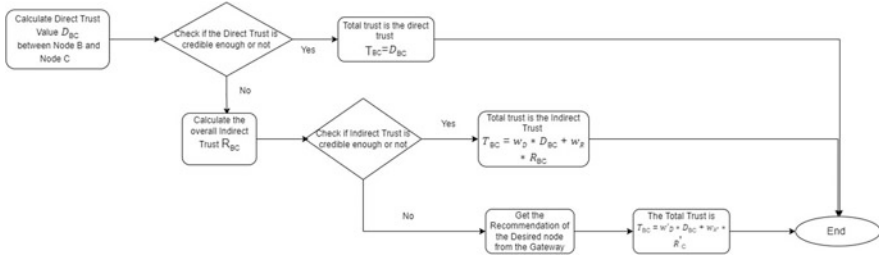


Fig. 2 Control flow of the trust management model

Table 1 A table of performance comparison between Che et al. [6] model and the proposed improvisation model

Model performances			
Criteria	Che et al.'s [6]'s framework	Proposed framework	Reasons
In case of good history of communication between the nodes	Yes	Yes	Because direct trust is credible enough in both models
In case of less history between the two nodes, but they have enough common neighbors	Yes	Yes	Indirect trust would be credible enough in both models due to enough neighbors
In case of less history between the two nodes and not enough common neighbors	No	Yes	Gateway trust calculation is used in proposed model as indirect trust is unreliable due to lack of enough neighbors
Trust computation of an existing node when requested by a newly added node	No	Yes	A new node would not have enough history for reliable direct trust and insufficient or close to zero neighbors for reliable indirect trust. Hence, gateway is useful in providing the trust value of requested node by new node

$$w'_D = (1 - H(D_{BC}) / \log 2D_{BC}) / ((1 - H(D_{BC}) / \log 2D_{BC}) + (1 - H(R'_C) / \log 2R'_C)) \tag{16}$$

$$w_{R'} = (1 - H(R'_C) / \log 2R'_C) / ((1 - H(D_{BC}) / \log 2D_{BC}) + (1 - H(R'_C) / \log 2R'_C)) \tag{17}$$

Figure 2 gives an overview of the proposed trust management model. This method ensures more accuracy and still keeps the algorithm lightweight to a certain

extent, since the gateway is only being used if the indirect trust is not credible enough.

Also shown below in Table 1 is a performance comparison analysis between Che et al. [6] model and the proposed improvisation model.

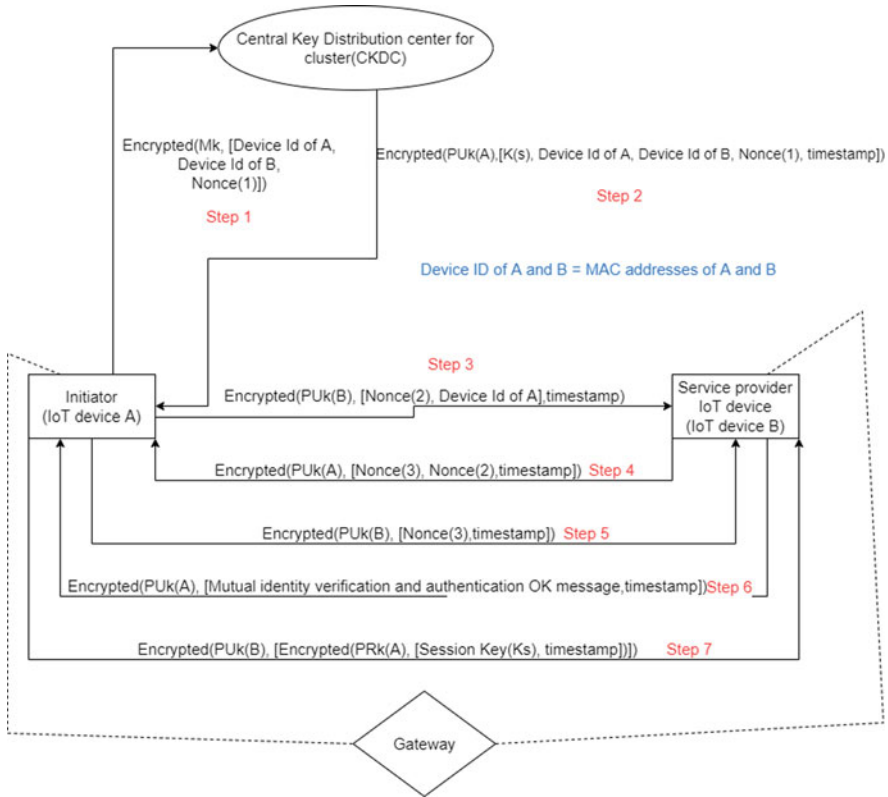
## 5 Proposed Cryptographic Secure Session Key Exchange and Identity Verification Model

This section proposes a session key exchange model as well as IoT device identity verification mechanism prior to the exchange of services/data between IoT devices. This section also emphasizes upon how the proposed model is resistant to some of the most typical attacks, such as the replay and relay attack, as well as the man-in-the-middle attack. The proposed framework includes a Central Key Distribution Center (CKDC), which is a KDC for a cluster of IoT nodes. It contains and validates the identities of all the IoT devices which are part of the cluster. It also generates session keys as and when two devices parts of the cluster need to engage with each other for services or information. Any device part of the cluster which needs to initiate communication with another device can request the CKDC for a session key. This session key is then exchanged between the two IoT devices communicating with each other securely as shown in Fig. 3.

It is also imperative to mention that the lifetime of an assigned session key should be optimal. That is, the lifetime of the key must not be too long to allow the attacker to conduct cryptanalysis or brute force and make use of the session key for malicious purposes. At the same time, the session key lifetime must not be too short as in this case repeated request must be made to CKDC for a new session key, increasing the computation overhead and latency of communication. Hence, an optimal balance between security, integrity, and computational overheads is taken care of.

### 5.1 Architecture of the Model

The cluster of IoT devices can communicate and take the services of each other after their identity verification and trust establishment. Hence, in order to make this data/communication exchange secure and private, a central key distribution center (CKDC) is required for each cluster of IoT devices which creates and issues a secret session key whenever requested by any IoT device which is part of the cluster. This CKDC along with the cluster gateway contains a list of all MAC addresses of the devices part of the cluster. As and when a new IoT node is added to the cluster by the gateway, the list of MAC addresses is updated in the gateway as well as the



**Fig. 3** Flowchart and working description of secret session key exchange and identity verification

CKDC. The CKDC then uses these MAC addresses to cross verify the identities of the devices requesting it for a secure session key. This ensures that no device outside the IoT cluster or an attacker can request the CKDC for a secret session key (Ks), hence maintaining the integrity of the data flow between devices in the cluster. Figure 3 shows the framework environment and also depicts the flow of the model. Firstly, the flow of the model starts with a secure form of key exchange of a master key (Mk) unique between each node and the CKDC which would be used in order to communicate with the CKDC only by encrypting the data/request being sent to the CKDC. This is a symmetric master key which would only be available to the IoT nodes present in the cluster and the gateway for that cluster. The Diffie–Hellman key exchange algorithm is used for this purpose.

## 5.2 Diffie–Hellman Algorithm for Secure Master Key (*Km*) Exchange

The Diffie–Hellman algorithm is a widely used key exchange algorithm as introduced in [15]. Step 0 in the model for secure key exchange is for the gateway to provide each node added to the cluster with two public numbers namely *P* and *G* (the same *P* and *G* values are provided to all the nodes in the cluster), which will then in turn be used by the IoT nodes to generate their own public keys after choosing a private key value for themselves which will not be revealed to any other node, hence making the exchange of the master key (*Km*) possible. The algorithmic explanation is shown below:

1. The CKDC chooses specific values for *P* and *G*, where *P* is a large prime number and *G* is its primitive root. These values are then passed on to the Gateway. The Gateway is then responsible for presenting these values to all the nodes present in the cluster. The nodes part of the cluster is then instructed by the CKDC to choose a private key value *Pk*.
2. Then, the public keys are generated using the above values with the below formulas and then exchanged with all the other nodes part of the cluster:

$$PUk(A) = G^{Pk(A)} \text{ mod } P \quad (18)$$

$$PUk(B) = G^{Pk(B)} \text{ mod } P \quad (19)$$

3. The nodes are then aware of the secret master key (*Km*) by using the below formulas:

$$Kmb = PUk(CKDC)^{Pk(A)} \text{ mod } P \quad (20)$$

(secret keys figured out my node A and by node B respectively, but in general it can be any other node part of the cluster too)

$$Kmb = PUk(CKDC)^{Pk(B)} \text{ mod } P \quad (21)$$

The primary purpose of Diffie–Hellman is to be able to secretly exchange a symmetric key as highlighted in [16] along with its significance. The main reason as to why asymmetric secret master key (*Km*) is generated instead of using asymmetric encryption in order to make requests/communication with CKDC is that by the property of symmetric encryption, only the parties that share the common secret key can engage in encryption and decryption unlike asymmetric encryption where anyone with access to the public key of the CKDC can make requests to the CKDC. Hence, since the individual master key (*Km*) is only be available to nodes within the cluster, hence, it is being ensured that an outside third-party device or attacker is not able to make requests for session key (*Ks*) or attacks such as the DDoS attack in order to crash the CKDC.

### 5.3 *Sequential Flow of the Proposed Model*

**Step 1** Let us assume that an IoT device A wants to take the services of IoT device B. Since A is the initiator node, A requests the CKDC (Central Key Distribution Center) to generate a session key ( $K_s$ ), which would then be used for symmetric encryption of information/messages between Device A and Device B. Device A sends a request to CKDC by sending information which includes the Device ID of A (its own ID), the Device ID of B (device with which it wants to take services from), a Nonce(1) value and also a timestamp of the message to indicate at what time the message was sent. A nonce in this case is used to check the integrity of the communication. It is a one-time message which can take a form of text, image, etc. as highlighted in [17]. All the above information is encrypted using the master key of the CKDC ( $Mka$ ) which is available to only to the device making the request which in this case is Node A. This ensures that the request for a secret session key ( $K_s$ ) can only be sent by the IoT devices and the gateway and no device outside the cluster, and let us say an attacker can make any request to the CKDC.

**Step 2** The CKDC then verifies the identities of Device A and Device B by first decrypting the request message from Device A using the secret master key ( $Kma$ ) which is a symmetric key and then cross verifying the MAC addresses (identity of an IoT device) of Devices A and B to check if they are part of its IoT cluster. If either A's or B's identity is invalidated, the CKDC does not issue any session key and immediately informs the Gateway about a third-party device (attacker) trying to take services of the IoT devices in the cluster. Once the identities are verified, then the CKDC generates a secret session key ( $K_s$ ). After doing so, the CKDC then replies back to Device A with the secret session key ( $K_s$ ), device ID of B, Nonce(1) [this nonce can be verified by device by decrypting this message using its private key to verify that the reply has in fact come from the CKDC itself and not from any intermediate attacker as the CKDC has to use its own private key in order to access the nonce(1) and then again encrypt this nonce(1) using A's public key  $PUK(A)$ ] and finally the timestamp of the message. All of the aforementioned information is encrypted by using the public key of Device A ( $PUK(A)$ ).

**Step 3** Once Device A receives a reply back from CKDC, it uses its private key  $PRk(A)$  to decrypt the message, and as mentioned earlier, it first verifies that the Nonce(1) is intact and is in fact what it has sent to the CKDC itself. After this verification process, Device A now knows that it has in fact received a secret session key ( $K_s$ ) from the CKDC itself and not from any other third-party attacker. Hence, now Device A has knowledge of the session key ( $K_s$ ) given to it. So the next step is to pass on this same key ( $K_s$ ) to device B with which it has to communicate with. But before that, it has to verify that it is in fact communicating with Device B itself. Hence, Device A sends a message containing the device ID of A, Nonce(2) [this is used by A to ensure that it is communicating with B itself because since Nonce(2) is encrypted using the public key of B, only B can decrypt this nonce(2) using its private key ( $PRk(B)$ )]. Hence, A can be assured of B's identity by verifying if the

Nonce(2) that it receives back from B is the same value that it sent to B] and the timestamp of the message. All of this information is encrypted with the public key of Device B (Puk(B)).

**Step 4** Once Device B receives the message from Device A, it uses its private key to decrypt the message, and once it does so, it has the Nonce(2) in its plain text form. It now encrypts this Nonce(2) using Device A's public key (Puk(A)). Following is the information included in the message which Device B sends to Device A: Nonce(2) [created and sent by Device A to verify that it is communicating with Device B], Nonce(3) [created and sent by Device B to verify that it is communicating with Device A itself] and the timestamp of the message. All of this information is encrypted with the public key of Device A (Puk(A)).

**Step 5** In this step, Device A receives the message sent by Device B in the previous step. It decrypts this message using its own private key (PRk(A)). After decryption, it can first verify that Nonce(2) is in fact matching with the Nonce(2) that it sent in step 3 to Device B. After its verification, it then also has access to Nonce(3) which was created and sent by Device B. Following is the information included in the message which Device A sends to Device B: Nonce(3) [used by Device B to verify that it is communicating with Device A] and timestamp of the message. All this information is encrypted using the public key of Device B (Puk(B)).

**Step 6** Device B receives the message from Device A and uses its private key (PRk(B)) to decrypt the message. After doing so, it first verifies if the decrypted Nonce(3) is matching the Nonce(3) that it had created and encrypted and sent to Device A. After doing so, both devices, A and B, using nonces have verified that they are in fact communicating with each other and no other third party. Hence, an OK message is sent by Device B to Device A concluding that the mutual verification and authentication of the devices is completed. This OK message is also encrypted using the public key of Device A (Puk(A)). Up until this step, the verification and identification process is completed.

**Step 7** This step is the most crucial step of all, as it is in this step that the secret session key (Ks) is exchanged between Device A and Device B. This step involves an encrypted message within an encrypted message. The intuition behind this step is that the session key (Ks) should only be accessed by B and no other node in the cluster. So for this reason, firstly, Device A which has the secret session key encrypts the secret session key (Ks) using its private key. Let us call this encrypted cipher text as C1(Ks). This cipher text is now encrypted again along with the timestamp of the message using the Public Key of Device B (Puk(B)). So the final cipher text is now C2(C1(Ks)). Hence, once B receives the message, the outer cipher which is C2(C1(Ks)) can now be only decrypted by Device B as only it has the private key (PRk(B)), which can be used for decryption. Hence, after B uses its private key, it now has access to C1(Ks). Now, since C1(Ks) is encrypted using the public key of Device A, hence C1(Ks) can be decrypted using the public key of Device A (Puk(A)), which is available to all the devices in the cluster including Device B. Hence this step ensures device B that the session key (Ks) has come from Device

A only and not any other attacker or a third party as Device B would only be able to decrypt  $C1(Ks)$  using the public key of A if it was encrypted using the private key of Device A. But the private key of Device A is only present with A. Thus, B can confirm that the message is sent by A only and that it has not been tampered by any third party attacker. This step also ensures Device A that only Device B will be able to decrypt the encrypted message sent by Device A as  $C2(C1(Ks))$  can only be decrypted using the private key of Device B which only Device B contains.

## **5.4 Resistance to Common Attacks**

This section discusses typical and common attacks such as Man-in-the-Middle (MiTM), Relay, and Replay attacks and how the proposed model is resistant to these attacks:

### **5.4.1 Man-in-the-Middle (MiTM) Attack**

MiTM attack involves an impersonation of one party to another when 2 parties communicate with each other. In the proposed framework, even if there exists an attacker C in the middle who eavesdrops on the communication between Device A and Device B, the attacker cannot gain access to the session key (Ks) as it is encrypted using each of the device's public keys, and by the concept of asymmetric encryption only the corresponding private key can be used for decryption of the cipher text. These private keys are only available to the respective devices. Hence, an attacker C cannot have any access to the secret session key (Ks). The presence of a nonce associated with each message also ensures that the model is resistant to an attack in which the attacker intercepts the cipher text and modifies it in a manner to tamper with the information. If this is the case, even the nonce value gets modified and hence the nonce sent and received will not match each other. Hence, no service/information is sent ensuring that the data are safe and secure.

### **5.4.2 Relay Attack**

In relay attack, an adversary intercepts the message and manipulates it. The presence of a timestamp associated with each message is used in the model. Since relaying of messages takes some additional time, the IoT devices can make use of this timestamp in order to determine if the message has been relayed via a third-party device, which in this case is the attacker by accounting for the transmission time between Node A and Node B as well as an added buffer time ( $T_b$ ), which is meant to account for traffic congestion delays. This same concept is also used for communication between an IoT device and the CKDC and also between an IoT device and the gateway.

### 5.4.3 Replay Attack

A replay attack involves replaying an old message. In this case, let us assume an attacker *C* who using some cryptanalysis technique or brute force technique has figured out the secret session key (*K<sub>s</sub>*) of an old message. But the framework proposed ensures that this type of attack is not feasible, mainly because the secret session key (*K<sub>s</sub>*) being used by any two IoT devices expires after a certain amount of time (let us say  $T(\text{expiry})$ ). Hence, after the session key is expired, device *A* again needs to request the CKDC for a new session key to be used to communicate with Device *B*, and the old session key is retired. Also, each IoT device keeps a track of the secret session keys used in the past. Hence, if an attacker attempts to impersonate an IoT device part of the IoT cluster by using an old session key, the device the attacker is communicating with (Device *B* in this case) would cross-check if the session key (*K<sub>s</sub>*) given to it has already been used in the past. If the session key has already been used in the past, Device *B* is able to detect that a replay attack has taken place, and hence communication is immediately terminated and IoT device *B* does not provide any service/data.

## 6 Conclusions and Future work

This chapter has analyzed various factors that determine the quality of an Identity Management System and looked into how different implementations that has focused on optimizing these factors. It was found that almost no work in the area of IdMS in heterogeneous IoT ecosystems to ensure interoperability. This chapter also presented a hybrid, low-latency framework for trust management in an IoT cluster with the involvement of cluster gateway by making use of direct trust, indirect trust, and gateway-based trust computation which is used when direct trust and indirect trust is unreliable due to lack of sufficient feedback from other IoT nodes present in the cluster. A model is also proposed which is used for cryptographic secure secret session key exchange and identity validation-verification between IoT devices in the cluster. The framework presented has been shown to be resistant to typical and common attacks such as the “man in the middle attack” and “relay and replay attack.” The framework ensures privacy, security and the integrity of data/communication between IoT devices within a cluster.

Future research can focus on the interoperability aspect of IdMS in heterogeneous IoT ecosystems, as more interoperability will lead to higher levels of automation, hence, resulting in reduced labor costs and reduced time to complete tasks. More work can also be done to implement and scale up the framework presented to several IoT clusters, involving multiple gateways and trust establishment between IoT devices of different clusters.



## References

1. L. S. Vailshery, Internet of Things (IoT) and non-IoT active device connections worldwide from 2010 to 2025, <https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/>, [Accessed: 13-02-2022] (2021).
2. C. Cyrus, IoT Cyberattacks Escalate in 2021, According to Kaspersky, <https://www.iotworldtoday.com/2021/09/17/iot-cyberattacks-escalate-in-2021-according-to-kaspersky/>, [Accessed: 13-02-2022] (2021).
3. M. L. C. F. R. L. W. Stork, Decentralized identity and trust management framework for Internet of Things, in: IEEE International Conference on Blockchain and Cryptocurrency (ICBC), 2020, pp. 1–1. <https://doi.org/10.1109/ICBC48266.2020.9169411>.
4. R. R. P. Najera, J. Lopez, Securing the Internet of Things, IEEE Computer 44 (2011) 51–58. <https://doi.org/10.1109/MC.2011.291>.
5. W. Warsun Najib, Selo Sulisty, Survey on trust calculation methods in Internet of Things, Procedia Computer Science 161 (2019) 1300–1307. <https://doi.org/10.1016/j.procs.2019.11.245>.
6. S. C. R. F. X. L. X. Wang, A lightweight trust management based on Bayesian and entropy for wireless sensor networks, Security and Communication Networks. <https://doi.org/10.1002/sec.969>.
7. A. J. Suriadi Suriadi, Ernest Foo, A user-centric federated single sign-on system, in: 2007 IFIP International Conference on Network and Parallel Computing Workshops (NPC 2007), 2007, pp. 1–1. <https://doi.org/10.1109/NPC.2007.64>.
8. J. C. David W. Kravit, Securing user identity and transactions symbiotically: IoT meets blockchain, in: 2017 Global Internet of Things Summit (GIOTS), 2017, pp. 1–1. <https://doi.org/10.1109/GIOTS.2017.8016280>.
9. A. A. Caroline Chibelushi, Alan Eardley, Identity and access management for the Internet of Things, Computer Science and Information Technology 1. <https://doi.org/10.13189/csit.2013.010201>.
10. D. G. R. S. Poonam Ninad Railkar, Dr. Parikshit Narendra Mahalle, Scalable trust management model for machine to machine communication in Internet of Things using fuzzy approach, Turkish Journal of Computer and Mathematics Education 12 (6). <https://doi.org/10.17762/turcomat.v12i6.5691>.
11. F. B. I.-R. Chen, J. Guo, Scalable, adaptive and survivable trust management for community of interest based Internet of Things systems, in: IEEE Eleventh International Symposium on Autonomous Decentralized Systems (ISADS), 2013, pp. 1–1. <https://doi.org/10.1109/ISADS.2013.6513398>.
12. A. B. Sarra Jebri, Mohamed Abid, Ltama-algorithm: Light and trust anonymous mutual authentication algorithm for IoT, in: IEEE 87th Vehicular Technology Conference (VTC Spring), 2018, pp. 1–1. <https://doi.org/10.1109/VTCSpring.2018.8417686>.
13. B. R. A. R. K. L. E. K. Koushik., A lightweight authentication and key exchange protocol for IoT. Workshop on Decentralized IoT Security and Standards. <https://doi.org/10.14722/diss.2018.23004>.
14. S. R. D. D. A. M. M. H. Mahalat, B. Sen, PUF based lightweight authentication and key exchange protocol for IoT, in: 18th International Conference on Security and Cryptography (SECRYPT), 2021, pp. 698–703. <https://doi.org/10.5220/0010550906980703>.
15. M. H. W. Diffie, New directions in cryptography, IEEE Transactions on Information Theory 22. <https://doi.org/10.1109/TIT.1976.1055638>.
16. J. K. Manoj Mishra, A study on Diffie-Hellman key exchange protocols, International Journal of Pure and Applied Mathematics 1. <https://doi.org/10.12732/ijpam.v114i2.2>.
17. G. M. Kjøien, A brief survey of nonces and nonce usage, in: The Ninth International Conference on Emerging Security Information, Systems and Technologies, 2015, p. 1.

# Plant Pest Detection: A Deep Learning Approach



Nilkamal More, V. B. Nikam, and Biplab Banerjee

## 1 Introduction

Agriculture is the principal sector of Indian economy. India's agriculture sector reports 18% to the Gross Domestic Product (GDP) and bestows employment to 50% of the country's workforce. Majority of the population in India rely heavily on the agricultural sector, out of which agriculture is the mainstay of Maharashtra state. According to the Indian Council of Agricultural Research, more than 35% of crop production is lost every year due to pests and diseases. Pests are cold blooded living organisms that grow unwelcome and cause damage to crops and livestock and might carry disease-causing microorganisms and parasites. Pests are the world's most diverse group of animals. Nearly 30 million species are found worldwide, out of which about 1.4 million have been briefly described. India is among the 12 mega biological diverse countries of the world, comprising nearly 7% of the world insect fauna. Current evaluation shows that out of nearly 63,760 species of insect in India, around 21,166 species are endemic. Considering species diversity, in which they come in almost every shape, most pests have a few things in common. Due to this vast diversity in pest species, their identification is a challenging task. The crop sensitivity combined with weather conditions has made pests habitual on the crops throughout the stages of its growth. Observing the pests manually is a complicated and time-consuming task due to its complex nature. Therefore, to reduce manual effort, accurate pest predictions are required for ensuring that

---

N. More (✉) · V. B. Nikam  
Veer mata Jijabai Technological Institute, Mumbai, India  
e-mail: [neelkamalsurve@somaiya.edu](mailto:neelkamalsurve@somaiya.edu); [vbnikam@it.vjti.ac.in](mailto:vbnikam@it.vjti.ac.in)

B. Banerjee  
Centre of Studies in Resources Engineering, IITB, Mumbai, India  
e-mail: [bbanerjee@iitb.ac.in](mailto:bbanerjee@iitb.ac.in)

the farmers' lives are hassle-free. Food security is intimidated by an increase in the number of outbreaks of plant pests. These pests jeopardize food security and have broad economic, social, and environmental consequences on agricultural productivity. To increase agricultural productivity, precise and on-time detection of crop pests is essential. Image classification using computer vision technology reduces the recognition cost and improves the speed and accuracy. However, the real images captured from farmlands often have high background noise, which makes it difficult to select general features that can suit all target pests. Deep learning is the emerging technology used in the past few decades for image recognition with an ability to automatically extract features and has proved to be efficient in terms of speed and accuracy which can help us in precise identification.

This paper is arranged in the following sections: Section 2 is the literature survey related to this work. Section 3 includes the proposed architectures for the system. Section 4 is the implementation and result analysis section, and Sect. 5 mentions the conclusion and future scope.

## 2 Literature Survey

The researcher evaluated faster R-CNN, single-shot and multibox detector (SSD), and multilayer perceptron (MLP) on tomato plants with a better solution, speed, and 82.51% accuracy using R-CNN [1]. This paper gives a module channel-spatial attention (CSA) merged into CNN backbone and a region proposal network (RPN) for the proportional images of the region. The contextual regions of interest (RoIs) are used for improving detection accuracy on Multi-class Pests Dataset 2018 (MPD2018) captured in the multispectral light trap covering more than 80k images categorized in 16 classes with 75.46% precision which is calculated by mean average [2]. In this paper, researchers have used a VGG19 to extract high-dimensional features from insect images and RPN to learn the location of insects in images. The dataset consists of 24 insect species with 660 images in total with 0.8922 mAP [3]. This paper performs pest identification with the complicated farmland background, using deep residual learning with pest image recognition accuracy for ResNet-101, of which 98.67% for ten classes of pest with 550 images of compound farmland background was achieved [4]. Recognition rate of 85.5% was achieved to identify insect pests under complicated environments using local configuration pattern (LCP) and support vector machines (SVM) [16] for ten categories with 40–70 sample images in each category [5]. Deep convolutional neural network using GoogLeNet model and GrabCut was used for pest classification with 93% accuracy for complex backgrounds and 98.9% for simple backgrounds for ten species of the pest containing 5629 images [6]. In this paper the researcher has used adaptive thresholding combined with VGG16 and SSD for detecting and classifying six pests. The accuracy of 84% and 86% is achieved using both the

models, respectively [7]. Total 75, 000 images associated with 102 categories of pest species for different crops. This dataset has the inter- and intra-class variance challenges and data imbalance [8]. This paper consists of six categories of apple pest using the multilayer perceptron neural network where results showed that artificial neural networks (ANN) are effective to support the process of identification [9]. The research is based on a technique that uses deep learning to detect oilseed rape pests, which improves the precision using mean average (mAP) to 77.14%, with 12 typical oilseed rape pests [10].

The researcher predicted the occurrence of extreme activities of a pest on pest surveillance datasets by applying analytical techniques to understand pest population dynamics of *Helicoverpa armigera* or pod borer on chickpea (*Cicer arietinum* L.) crop which will be advanced within a week [11]. This paper surveys the fall armyworm incidences on corn, sugarcane, etc. in different districts of Maharashtra with the level of infestation [12]. This paper determines the possibility to use the sinusoidal growing degree day model as an option to the daily temperature data mainly when fitted for monthly temperature data [13]. Correlation between different weather parameters using basic regression equations for soilborne insect pests infesting oats and population dynamic of white grub, wireworm, cutworm/armyworm is registered [14]. The developed temperature-based phenology model was introduced in this paper using the Geographic Information System (GIS) to inspect the potential future pest status of *S. litura* on soybean areas in Indian states like Madhya Pradesh, Maharashtra, and Rajasthan [15]. For this case study, we have considered a total of ten classes for pests of corn and tomato, out of which armyworm and white grub are considered for growing degree day calculation.

### 3 Proposed System

Visual observation at a single glance may not be accurate at every point of time. Because of this the farmers may use the pesticides which are not necessary and may cause harm to the crops. So, the system must be developed for classification of correct pests along with periods of instances for pest damage towards better decisions.

#### 3.1 Proposed System Architecture

Figure 1 shows the proposed system that consists of various stages which are data preprocessing, deep learning model, prediction of pest images, growing degree day calculation, and decision making.

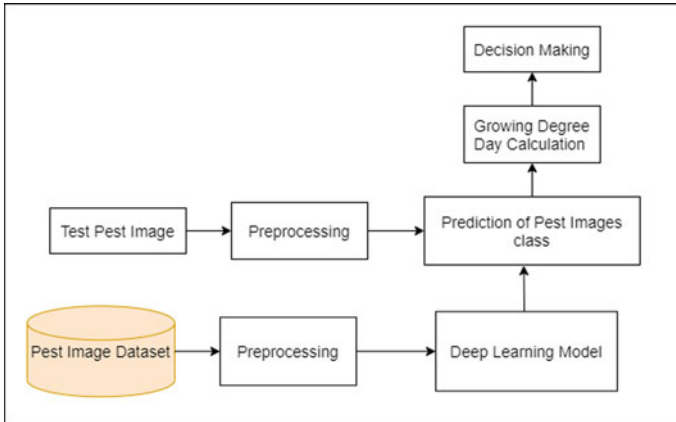


Fig. 1 Proposed system architecture

### 3.2 Deep Learning Model

The system uses the modified VGG16 model, which is the convolutional neural network architecture in deep learning. This model is applied on the dataset for pest image classification for feature extraction, and classification is done using this model. Feature extraction selects and/or combines variables into features, effectively reducing the amount of data that must be processed while still accurately and completely describing the original dataset. Classification is utilized to predict the labels/targets or categories from the input data.

*Prediction of Pest Image Class* The image for which the class of pest is to be predicted is first preprocessed and then applied to the trained convolutional neural network using the modified VGG16 model for predictions.

*Growing Degree Day Calculations* The growing degree day (GDD) is used for forecasting the time of attack of the pest on the corn crop based on temperature attributes. The life cycle of the pest and host crop should also be taken into consideration while calculating the GDD for a particular pest, and it varies for different pest species. While calculating the accumulated GDD, we have decreased the granularity by taking a month-wise mean of maximum and minimum temperature. Equation (1) is the accumulated growing degree day (AGDD) in a statistical term:

$$AGDD = \sum \left( \frac{(T_{max} + T_{min})}{2} - T_{base} \right) \tag{1}$$

where,

$T_{max}$ : Maximum temperature of the month

$T_{min}$ : Minimum temperature of the month

$T_{base}$ : Baseline temperature of pest

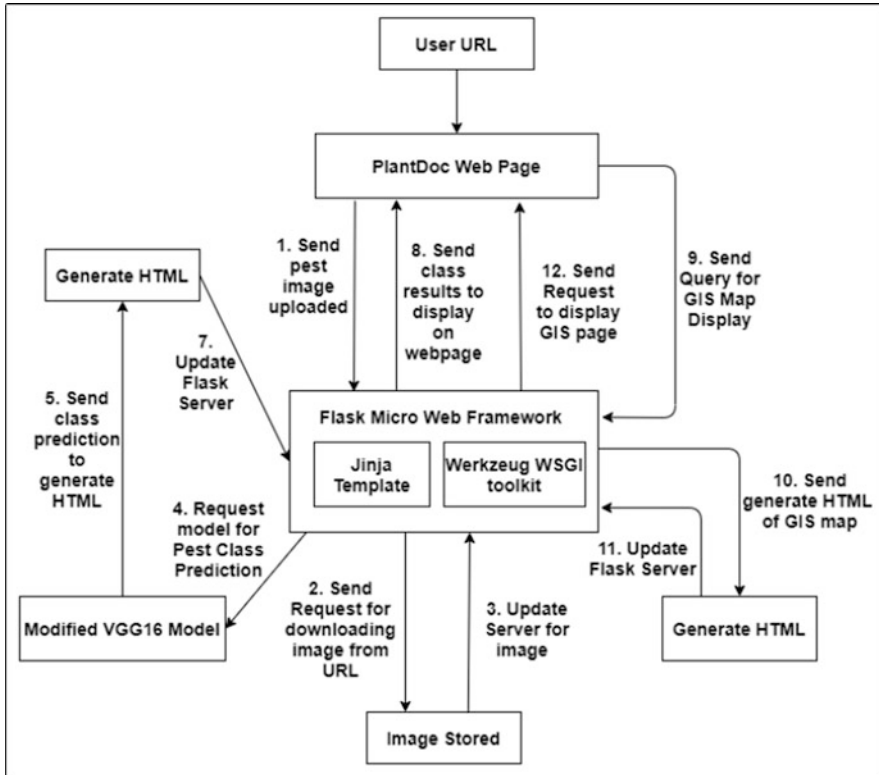


Fig. 2 Work flow diagram for the service architecture

We have calculated the accumulated growing degree day (AGDD) for different districts of Maharashtra state, and a case study of armyworm and white grub pest is considered. This is done for time and region of pest vulnerability anticipation.

*Decision Making* The decision making includes the time of scheduling of control measures that are cultural, biological, mechanical, organic, and if required chemical control measures for pest and disease infestation. Applying the pest control methods or treatment at the point that the pest is most vulnerable provides effective results.

*Service Architecture for Proposed System* Figure 2 shows the flow diagram for the service architecture. First the pest dataset is trained using VGG16 and stored for further use. After this, the model needs to be deployed and this can be done using the flask framework.

The web application is developed using a flask. When the user uses this application URL, the PlantDoc web page opens up, where the user needs to upload the test pest image. Then the uploaded image is sent to the flask framework which is downloaded and stored in the file directory that can be accessed by request object.

Once the test image query is received, it can be sent to the VGG16 model for class prediction. The prediction will be done at the backend, and then just through HTML generation, the results will be sent back to flask for displaying it on the web page. The GIS web-based application is also integrated with flask. For this the web page generated using QGIS to web plugin is converted in flask file layout, and then when user sends the query to flask for viewing the time and district of pest attack on GIS map for the identified pest, through HTML generation, the results are updated to flask server and further displayed on the web application.

## 4 Implementation and Result Analysis

This section includes the dataset collection of various pests, the VGG16 model execution, and performance examination.

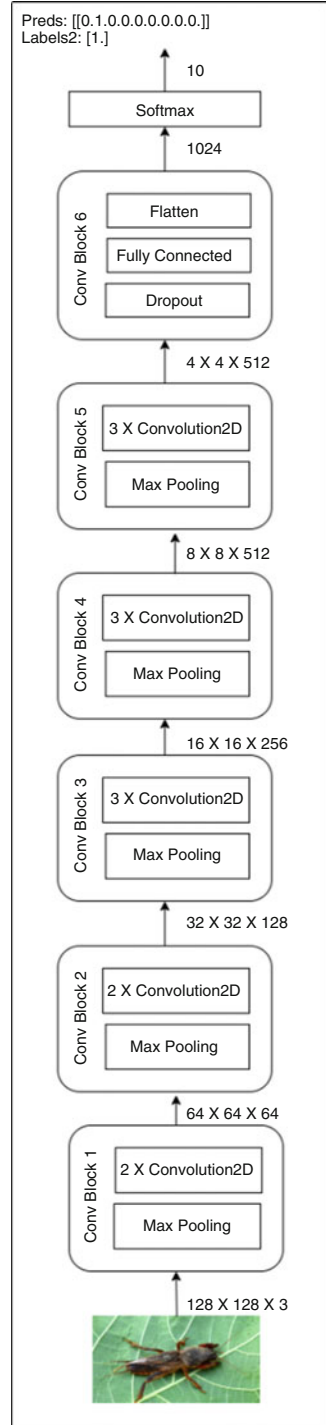
### Dataset

The dataset consists a total of 5195 images for ten categories of pest species of corn and tomato images with complex and varied backgrounds, initially collected from the IP102 dataset. But this dataset has a high imbalance ratio between classes. So, to overcome this issue, images from IPM image website and some from various sources are also added to these classes. The dataset is split in a 9:1 ratio of train and validation set. So, there are 4709 images for training and 486 images for validation. Also for calculation of growing degree day, the maximum and minimum temperature for the last 3 years is collected. This weather data is collected month-wise for all districts of Maharashtra from January 2018 to June 2020.

*VGG16 Model Implementation and Results* VGG16 is a network architecture which provides magnificent classification execution in convolutional neural networks. This VGG16 model is evolved on the grounds of the AlexNet network. It consists of 16 layers, which not only has good classification outcomes on large-scale data, but also possesses excellent expansion ability on the data sets and was considered to be the one of the excellent vision model architecture till date. The improved architecture of VGG16 has convolutional and pooling layers. They are similar to the pre-trained VGG16, but a dropout method is applied for fully connected layers which resolve the overfitting issue of the model. The dropout which is a regularization technique will help the model to learn more vigorous features. The detail architecture for the deep learning convolutional neural network model is mentioned in Fig. 3.

The image of shape  $128 \times 128 \times 3$  is the input to the model which passes through the convolutional and max pooling layers blocks. After the final max pooling layer, there is a flattened layer with output shape 8192 followed by the fully connected layer and dropout layer of output shapes 1024. Finally, the output layer is the fully connected layer using softmax activation function with output shape 10. It classifies image and provides probability values in the range of 0 to 1 for each class. In Fig. 4 the input image for mole cricket pest is passed to the modified VGG16 model

**Fig. 3** Customized VGG16 model





Layer (type)	Output Shape	Param #
vgg16 (Model)	(None, 4, 4, 512)	14714688
flatten_1 (Flatten)	(None, 8192)	0
dense_1 (Dense)	(None, 1024)	8389632
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 10)	10250
Total params: 23,114,570		
Trainable params: 15,479,306		
Non-trainable params: 7,635,264		

**Fig. 4** Parameters of customized VGG16 model

which gives output as the probability values for each class. The hyper-parameters determining the organization of the VGG16 model is shown in Fig. 4. The pest classification VGG16 model fits well on the pest dataset. The pest classification training accuracy of 92.40% using the VGG16 model is achieved, to effectively recognize insect pests under complex environments.

The results of the model from confusion matrix that describes the performance of the classifier on the validation dataset for each pest class and the classification report shown in Fig. 6. Small datasets may provide better accuracy but will not guarantee precise results when applied on real-world images.

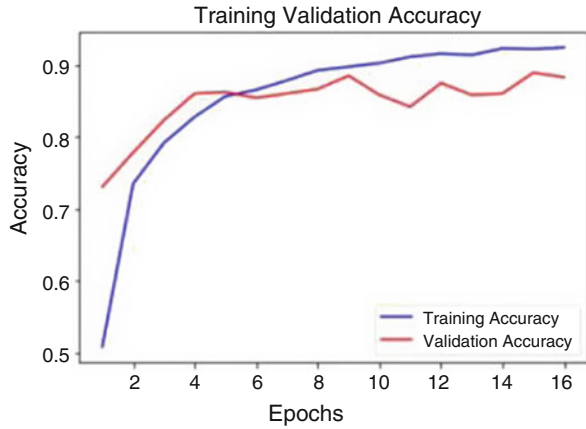
Even many datasets that are used for training are captured in a controlled environment which makes real-time detection a challenge. The problem of class imbalance ratio which was the reason for failure for getting better accuracy can be improved in this proposed model with better results.

Figure 5 gives the training validation accuracy graph for the developed model. From this graph it can be interpreted that the validation images of pests show great accuracy of prediction for the developed model. The precision, recall, and F1-score metrics are calculated from which it can be indicated that the identification for each pest is quite precise with a mean average precision of 0.96 as shown in Fig. 6.

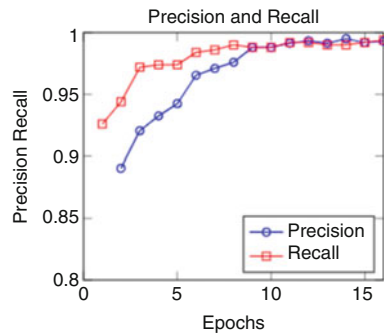
## 5 Conclusion and Future Scope

There is some improvement in the training accuracy and the mean average precision of the pest classification model compared to the prior techniques with use of dropout and single stopping mechanism in the modified VGG16 model. Foreseeing the pest attack incidences for a month in advance will help in scheduling the proper timing of management strategies. Applying the pest control methods or treatment at the point

**Fig. 5** Training validation accuracy graph



**Fig. 6** Graph for precision and recall



that the pest is most vulnerable provides effective results which can be deduced from the GIS maps.

**Acknowledgment** The author gratefully acknowledges the support of the Faculty Development Center (VJTI) and Computer Engineering and Information Technology Department of VJTI, Mumbai, for providing research and experimental platform and would also appreciate the valuable feedback and comments from reviewers.

## References

1. Aitor Gutierrez , Ander Ansuategi, Loreto Susperregi, Carlos Tub-o, Ivan RankiT, and Libor LenDa, “A Benchmarking of Learning Strategies for Pest Detection and Identification on Tomato Plants for Autonomous Scouting Robots Using Internal Databases”, Journal of Sensors Volume 2019, Article ID 5219471, DOI <https://doi.org/10.1155/2019/5219471>
2. Liu Liu, Rujing Wang, Chengjun Xie , Po Yang ,(Senior Member, Ieee), Fangyuan Wang, Sud Sudirman, And Wancai Liu, “PestNet: An End-to-End Deep Learning Approach for Large-Scale Multi-Class Pest Detection and Classification”, April 10, 2019, DOI: <https://doi.org/10.1109/ACCESS.2019.2909522>

3. Denan Xia , Peng Chen,, Bing Wang, Jun Zhang and Chengjun Xie , “Insect Detection and Classification Based on an Improved Convolutional Neural Network”, *Sensors* 2018, 18, 4169; doi:<https://doi.org/10.3390/s18124169>
4. Xi Cheng, Youhua Zhang, Yiqiong Chen, Yunzhi Wu, Yi Yue, “Pest identification via deep residual learning in complex background”, *Computers and Electronics in Agriculture* 141 (2017) 351–356, <https://doi.org/10.1016/j.compag.2017.08.005>
5. Limiao Deng, Yanjiang Wang, Zhongzhi Han, Renshi Yu, “Research on insect pest image detection and recognition based on bio-inspired methods”, *Biosystems engineering* 169 (2018) 139e148, <https://doi.org/10.1016/j.biosystemseng.2018.02.008>
6. Yanfen Li, Hanxiang Wang, L. Minh Dang, Abolghasem Sadeghi-Niaraki, Hyeonjoon Moon, “Crop pest recognition in natural scenes using convolutional neural networks”, *Computers and Electronics in Agriculture* 169 (2020) 105174, <https://doi.org/10.1016/j.compag.2019.105174>
7. Nguyen Tuan Nam, Phan Duy Hung ,”Pest detection on Traps using Deep Convolutional Neural Networks”, *ACM ISBN 978-1-4503-6470-6/18/06*, <https://doi.org/10.1145/3232651.323261>
8. Xiaoping Wu; Chi Zhan; Yu-Kun Lai; Ming-Ming Cheng; Jufeng Yang, “IP102: A Large-Scale Benchmark Dataset for Insect Pest Recognition”, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)
9. P. Boniecki, K. Koszela, H. Piekarska-Boniecka J. Weres, M. Zaborowicz, S. Kujawa, A. Majewski, B. Raba, “Neural identification of selected apple pests”, *Computers and Electronics in Agriculture* 110 (2015) , <https://doi.org/10.1016/j.compag.2014.09.013>
10. Yong He , Hong Zeng, Yangyang Fan, Shuaisheng Ji and Jianjian Wu, “Application of Deep Learning in Integrated Pest Management: A Real-Time System for Detection and Diagnosis of Oilseed Rape Pests”, *Mobile Information Systems*, Volume 2019, Article ID 4570808, 14 pages <https://doi.org/10.1155/2019/4570808>
11. Rajat Gupta, BVL Narayana, P.Krishna Reddy, G.V. Ranga Rao, “Understanding *Helicoverpa armigera* Pest Population Dynamics related to Chickpea Crop Using Neural Networks ”, *IEEE International Conference on Data Mining (ICDM’03)*
12. Ankush Chormule, Naresh Shejawal, Sharanabasappa, CM Kalleshwaraswamy, R Asokan and HM Mahadeva Swamy , “First report of the fall Armyworm, *Spodoptera frugiperda* (J. E. Smith) (Lepidoptera, Noctuidae) on sugarcane and other crops from Maharashtra, India”, *Journal of Entomology and Zoology Studies* 2019; 7(1): 114–117,
13. M.N. Elnesr, A.A. Alazba, “An integral model to calculate the growing degree-days and heat units, a spreadsheet application”, *Computers and Electronics in Agriculture* 124 (2016), <https://doi.org/10.1016/j.compag.2016.03.024>
14. Ritesh Kumar, Ishtiyah Ahad, Sheikh Aafreen Rehman and Stanzin Dorje, “Impact of weather parameters on population dynamics of soil borne insect pests infesting oats (*Avena sativa* L.) in North Kashmir.”, *Journal of Entomology and Zoology Studies*, E-ISSN: 2320-7078, P-ISSN: 2349-6800
15. Babasaheb B. Fand, Nitin T. Sul, Santanu K. Bal, P. S. Minhas, “Temperature Impacts the Development and Survival of Common Cutworm (*Spodoptera litura*): Simulation and Visualization of Potential Population Growth in India under Warmer Temperatures through Life Cycle Modelling and Spatial Mapping”, *PLOS ONE* |DOI:<https://doi.org/10.1371/journal.pone.0124682>
16. J. Westbrook, S. Fleischer, S. Jairam, R. Meagher, And R. Nagoshi, “Multigenerational migration of fall armyworm, a pest insect”, November 2019, Volume 10(11), *Ecosphere* 10(11):e02919. <https://doi.org/10.1002/ecs2.2919>

# S.A.R.A (Smart AI Refrigerator Assistant)



Sachin Singh Bhadoriya, Saniya Kirkire, Rut Vyas, Satvik Deshmukh,  
and Yukti Bandi

## 1 Introduction

The fast pace of life has increased the popularity of instant foods. The recent quarantine period has piqued the interest of many in cooking. When it comes to food, deciding what recipe to make often takes longer than the actual preparation time. To save this time and present a wide variety of options to the user, S.A.R.A, an AI-powered refrigerator assistant, has been developed. Its features encompass “Recipe Recommendation” based on the user input. This assistant can interact with the user through a progressive web application. The data was processed remotely on a server. The data required for the recommendation algorithm was acquired by leveraging the web scraping technologies using open-source Python libraries. This data was then stored in JSON files. The textual data was organized with data wrangling and analysis techniques. The recipe recommendation functionality was achieved by developing a search engine specific to food recipes through extensive natural language processing. Further, the technologies used were completely open source, and the modifications which will be required to integrate our assistant are minimal as a result of which the whole idea is extremely budget-friendly. Introducing such a feature will not just save time but also entice the user to try newer recipes. The user can explore and include these newly discovered recipes in his or her diet. The goal is not only restricted to comfort and ease but also aims toward the accomplishment of a healthier lifestyle. In addition to this, a variety of additional features can be introduced in this assistant which will ensure even more ease and benefit for the user. This paper provides the detailed approach, methodology, system

---

S. S. Bhadoriya · S. Kirkire (✉) · R. Vyas · S. Deshmukh · Y. Bandi  
Department of Electronics and Telecommunication, D.J Sanghvi College of Engineering,  
Mumbai, India  
e-mail: [yukti.band@djce.ac.in](mailto:yukti.band@djce.ac.in)

architecture, and an in-depth analysis of the functionality this assistant is capable of. Results that were achieved with this solution are also presented here.

## 2 Literature Survey

Since the early 2000s, one has been fascinated by the idea of connecting all of one's daily life utilities to the Internet (IoT). Day by day, people want to spend less time on menial jobs and want everything connected to their smartphones. This led to LG launching the world's first smart refrigerator called the Internet Digital DIOS in June 2000 [3]. Although this was an unsuccessful venture due to its high costs and unnecessary features, this instigated other companies to enter this field and develop their technologies. One of the technologies implemented was a recipe recommendation system based on ingredient availability, and a survey of such work by some of the brilliant pioneers has been carried out.

Zheng Xian Li and their team proposed a personalized hybrid recommendation system that is an amalgamation of model-based CF algorithm and content-based filtering as a supplement to increase the accuracy of recipes recommended. They made use of Apache Spark as the engine for the recommendation, MySQL for storing and managing recommendation results or metadata, and Hadoop Distributed File System (HDFS) to manage unstructured data [6]. Their experiments displayed that the recipe suggestion system has a scalable computational capability to process massive information of recipes.

Diya Lu and the team proposed a recipe search based on nutritional value. Their primary focus was to display pre-trained recipe embeddings that yielded more diverse results in comparison to searches based on keywords. They adopted the cosine similarity to measure the distance between two recipe embeddings and sacrificed accuracy for speedy retrieval by deploying top-k approximate nearest neighbors (ANN) [2]. However, their approach of searching first then choosing additional nutrition information is a naive way to do nutrition-guided recipe search, and further research can be made in this domain.

Zhen Feng Lei and the team focused on recipe suggestions accompanied by logical interpretations produced from images or videos. They proposed a multimodal recipe recommendation system via the knowledge graph (RcpMKR). Further, they constructed a recipe knowledge graph (RcpKG) using multimodality and hierarchical thought and inculcated BERT-based multimodal models to generate explanations [7]. Thus, the proposed extensible multimodal recipe fusion framework provided guidelines for improving the performance of recipe recommendations and generating reasonable and acceptable explanations derived from images or videos related to the recommended results.

Vasvi Bajaj and the team made use of the graph database of the NoSQL family of databases to recommend recipes based on the ingredients selected in the query [5]. They loaded the Python-crawled data into a Neo4j database through an R-programming script. The nodes were made up of recipe and ingredient names, while

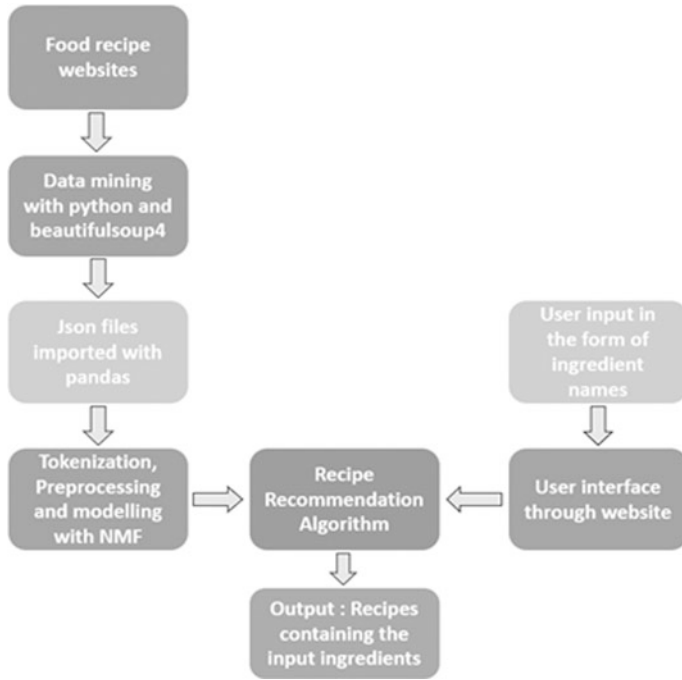


Fig. 1 Implementation flowchart

the edges represented the relationships between the nodes. They concluded that graph databases are significantly faster as they reduce the time required to compute “JOIN operation”-related queries which need to be executed in other NoSQLs and relational databases like RDBMS. However, such a system does not consider the weight of the ingredients with respect to the recipe and therefore is a very crude and not-so-customer-friendly recipe recommendation system (Fig. 1).

### 3 Implementation

The main pipeline of the system works on a singular directional system flow: the algorithm getting input from the users in the form of an array of the ingredients. The algorithm appends the top three recipes which can be made using the ingredients input by the user. The matching and ranking of recipes with respect to ingredients was achieved through extensive natural language processing during the training process for the algorithm. The data required to train the model and to build this search engine was gathered from three different food blogging websites with data mining techniques and web scraping technologies. To integrate an API with the

algorithm and to ensure that the output is sent back to the client, a web server that could be integrated easily with our system was required.

The working process of the whole project is divided into the following steps.

### ***3.1 Data Mining***

The collection of data was a crucial element in building the algorithm which is the backbone of our assistant. An adequate amount of organized data was needed to make sure that the algorithm is capable of generating useful patterns and ensure an optimum state where the model is neither overfitting nor underfitting. Manually generating a huge quantity of data consumes a lot of time, human effort, and computational resources. To tackle this problem, “Beautiful Soup,” a Python-based web scraping library was used in the first stage. It is capable of automating the entire process of data extraction from the Internet website efficiently out of XML and HTML files. Beautiful Soup works on a parse tree methodology that is very similar to the tree data structure. It allows the code to browse multiple pages concurrently in a hierarchical format owing to its structure.

The data from three different culinary websites was extracted using beautiful-soup4, spacy, and other open-sourced Python libraries and tools. The URL of these websites was fed to the scrapper which was used to extract the data including the title of the recipe, its ingredients, a picture, and the cooking instructions from the HTML web pages. The websites used were as follows:

- Food Network
- Allrecipes
- Epicurious

Multithreading was utilized to make sure this was done effectively and resulted in a total data collection of over 100,000 recipes. The data was stored in JSON file format and later imported with pandas for further processing and analysis. However, the scraping resulted in multiple data errors and required heavy preprocessing in order to be used for further algorithm implementation.

### ***3.2 Preprocessing and Tokenization***

The JSON files consisting of the raw data were directly obtained from websites through web scraping due to which many anomalies were introduced. To use the data for training, a considerable amount of cleaning was necessary. At this stage of implementation in the Jupyter notebook, the raw data stored in JSON files were imported using open-source libraries like pandas and NumPy. First, all the tuples with missing fields and the ones that contained links for recipe pictures were removed. Then, a lower limit of 20 characters was set for the instruction

```

ingredients = []
for ing_list in recipes['ingredients']:
    clean_ings = [ing.replace('ADVERTISEMENT','').strip() for ing in ing_list]
    if '' in clean_ings:
        clean_ings.remove('')
    ingredients.append(clean_ings)
recipes['ingredients']=ingredients

recipes.loc[0,'ingredients']

['4 skinless, boneless chicken breast halves',
 '2 tablespoons butter',
 '2 (10.75 ounce) cans condensed cream of chicken soup',
 '1 onion, finely diced',
 '2 (10 ounce) packages refrigerated biscuit dough, torn into pieces']

```

**Fig. 2** Preprocessing

attribute, and recipes that failed to meet this requirement were dropped on grounds of insufficient information. While extracting data, the term “ADVERTISEMENT” was included in each ingredient attribute due to the ad section of web pages which was dropped. Finally, several unnecessary elements of the data entities including words without any information value, punctuations, digits, symbols, and unwanted tabs/new lines as well as irregular entries in the data frame were removed as seen in the figure below. As the entire algorithm works on matching documents with the ingredient or recipe name fed by the user, the most suitable method was to combine all the columns of title, ingredients, and instruction into singular textual components (Fig. 2).

Extracting information from text is difficult, and hence the system needs to break the sentences into smaller chunks that are capable of being processed. There are various NLP techniques to obtain tokenized words from sentences. In the tokenization stage for our project, the spacy tokenizer was incorporated as it was comparatively faster and provided ease of customization. Python’s multiprocessing library was used for this heavy process. Here, the entire raw text was broken down into words that could be analyzed and correlated to each other to assign weights. The textual data was lemmatized and stop words were removed from the data. This tokenized text was further vectorized with a tfidf vectorizer to assign weights as per the relative significance of words in the document.



### 3.3 Creation of Word Embeddings

Conversion of these blocks of texts into quantitative figures is essential to draw accurate predictions. Word embeddings are used for this purpose. These are a type of word representations that allows words with similar meaning to have a similar representation. In the case of recipes, it is required to find the relevance of each word inside a corpus which will fetch the user recipes relevant to the ingredients they selected. Here, “Term Frequency- Inverse Document Frequency” (TFIDF) was used for the vectorization of words. This is done to find word frequencies of important words and discard words that appear a lot but have negligible meaning, for example, “The,” “a,” “an,” etc. The meaning increases with the number of times a word appears in a text data series, but it is compensated by its word frequency in the dataset. This was done using the `TfidfVectorizer` function provided by `sklearn` themselves. It provided generic weights to all the topics, which will relate to the frequency of these topics appearing inside the entire document set. In simpler words, TFIDF is a simple frequency score that tries to highlight words that are more interesting and important. The tokenized text was passed to fit inside this vectorizer and was later used to extract feature names or highlighted words. The TFIDF formula is as expressed in Eq. (1).

$$W_{x,y} = tf_{x,y} \times \log \left( \frac{N}{df_x} \right) \quad (1)$$

$tf_{x,y}$  = frequency of  $x$  in  $y$

$df_x$  = number of documents containing  $x$

$N$  = Total number of documents

### 3.4 Topic Modeling

Topic modeling is defined as a “type of statistical modeling for discovering the abstract ‘topics’ that occur in a collection of documents.” It is an unsupervised approach used for finding and observing a bunch of words (called “topics”) in large clusters of texts. Topics can be defined as “a repeating pattern of co-occurring terms in a corpus.” A good topic model should result in “health,” “doctor,” “patient,” and “hospital” for a topic healthcare and “farm,” “crops,” and “wheat” for a topic “Farming.” In the present case, these topics were the recipes and the words would be the ingredients. There are two algorithms that can be used for the topic modeling process which are well received by the community, namely, LDA [4] (Latent Dirichlet Allocation) and NNMF [1] (Non-Negative Matrix Factorization). The topic modeling algorithm which was chosen for this application was NNMF.

The choice was made based on the document score and stability of convergence. The document score refers to the number of accurate predictions one topic can do for  $x$  number of documents. The graphs for three of the topics were plotted for both LDA and NNMF.

NNMF decomposes (or factorizes) high-dimensional vectors into a lower-dimensional representation. These lower-dimensional vectors are non-negative which also means their coefficients are non-negative. Using the original matrix ( $A$ ), NMF will give you two matrices ( $W$  and  $H$ ).  $W$  is the topics it found, and  $H$  is the coefficients (weights) for those topics. In other words,  $A$  is articles by words (original),  $H$  is articles by topics, and  $W$  is topics by words. The matrix formed was later passed for text ranking (Fig. 3).

It was observed that as the document score increased, the number of documents that could be covered decreased. On comparing the above graphs, it could be concluded that:

- (a) NNMF required lesser computational power on the workbench when training and provided the ability to train for higher epochs.
- (b) NNMF was used as it provided a good convergence and had more distinction on topics.
- (c) NNMF showcased a leveling out on 200 documents; therefore as per the objective function, 200 was chosen as the document cutoff.

### **3.5 BERT**

BERT (Bidirectional Encoder Representations from Transformers) [8] is a transformer-based machine learning technique for natural language processing which was pre-trained by Google. It has been designed to help computers understand the meaning of ambiguous language in text by using surrounding text to establish context. It was considered as an alternative approach for the creation of word embeddings and topic modeling achieved through TFIDF and NNMF as it provides clusters as well as bidirectional relations (Fig. 4).

However, when BERT was trained to fit the dataset, it was observed that the clusters formed were highly scattered and were not providing the required degree of correlation between each other. The entire sequence of words is read at once by the transformer encoder (bidirectional nature) which adds to the complexity of the algorithm thereby harming the predictive capability of the recipe recommendation system. In addition to this, BERT is computationally expensive which goes against the aim of creating a budget friendly assistant. Thus, implementing the algorithm from ground up using a custom vectorization and NNMF in the end proved to be a more effective solution.

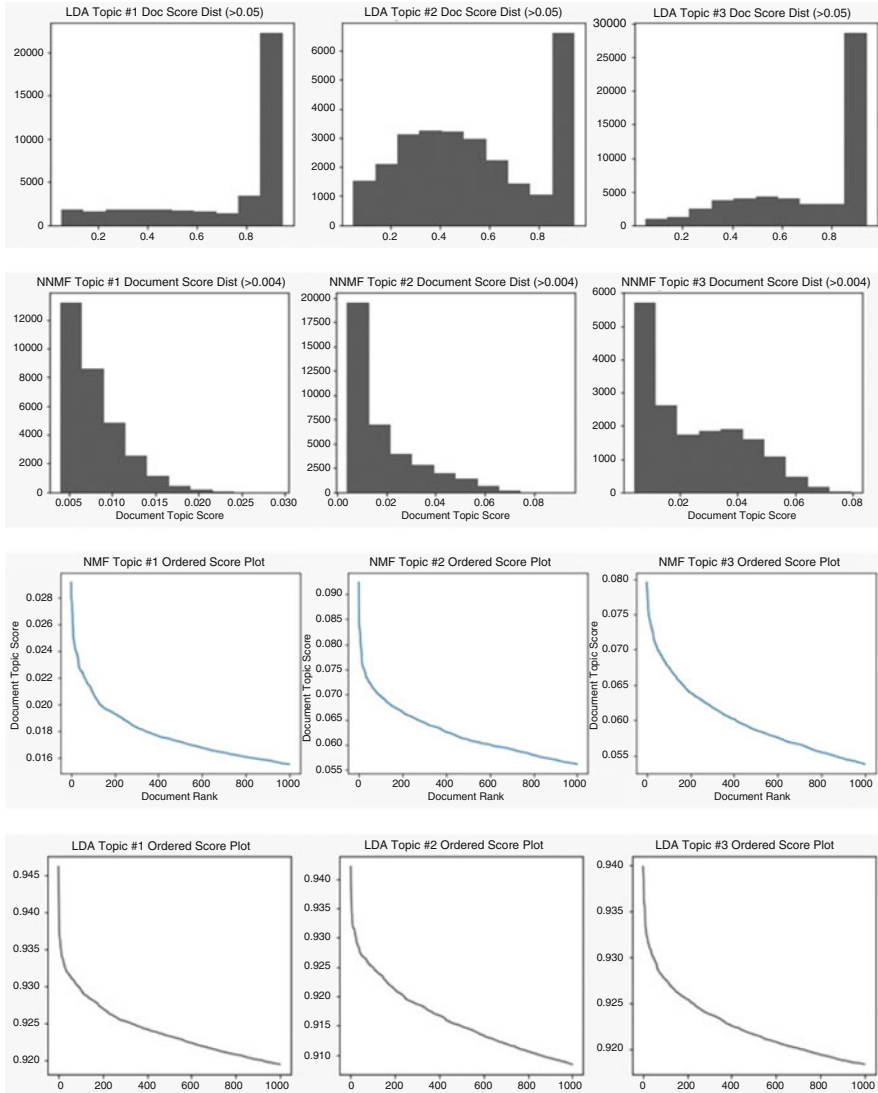
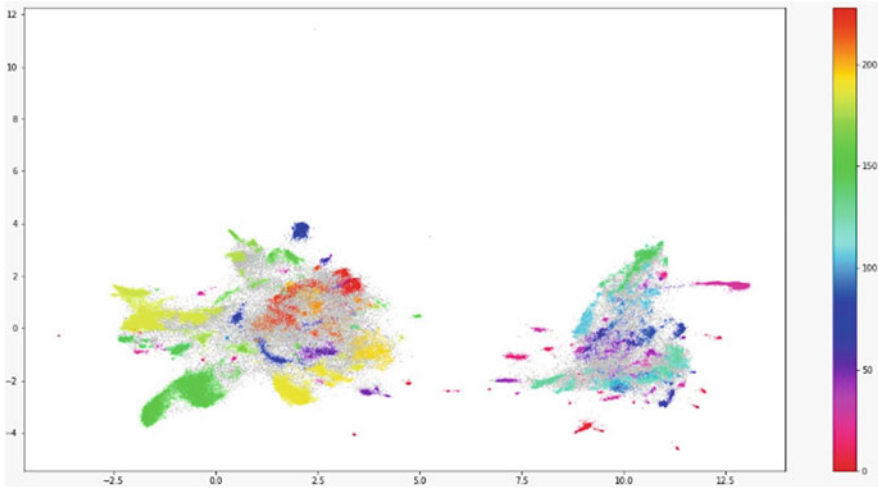


Fig. 3 Plot of LDA vs NNMF for accuracy of the algorithm with topic matching

### 3.6 Text Ranking

Text ranking consists of two functionalities in general: topic extraction and keyword extraction. The top-hundred documents ranked in similarity among topics were extracted, as after this count the similarities between them start reducing. These were sorted in a descending order to rank the higher similarities above. As recipes



**Fig. 4** Cluster mapping using BERT

are not traditional texts, they have to contain manually entered “stop words.” Thus, a couple of custom stop words for this specific use case were made. This set of stop words can be updated as per the requirement of the algorithm.

As the adjacency table was extracted from the recipes, the window was limited to four words at most, allowing the algorithm to be precise and not work with long strings. This matrix was later passed into a text ranking library called network. This generated a well-documented cluster, where one was able to find patterns between the keywords belonging to topics. Score distribution analysis was done for the NMF algorithm, and the results showcased that the range of keywords covered under the text topic “baking” extended to up to 10,000 of the top-ranked recipes falling in that category.

### 3.7 Querying Algorithm

A compilation of a system that can be used to extract recipes matching the keywords was referred to as the “querying algorithm.” This resulted in forming a compiled version of the above procedure into a storable file using panda’s library. Tags were generated for all the recipes as per their rankings with the keywords, and these tags would later

be utilized by the queries to find recipes. These keywords were indexed to shorten the searching time, much like a no-SQL database.

The query algorithm took an array of ingredients as input, these inputs were then split into different documents, and these documents went through a vectorizer sharing the same properties of that in the modeling procedure. The vectorized



Fig. 5 Query algorithm hosted on a local system using a flask server

ingredients, or keywords, were sent into a recursive function to break them down into NumPy arrays in descending weights. This NumPy array was multiplied with the previously obtained vectorized ingredients to obtain index form, which was added together using a reducer. This final vector was then used to search inside the tokenized and vectorized database to find the most appropriate score from the list and return that as the recipe to cook.

## 4 Results

This query algorithm was hosted on a local system using a flask server to utilize fast development speeds. This can easily be transferred in an AWS lambda function during the production stage. The front end was currently made with a JavaScript framework called “Next.js,” which takes in three ingredients and sends it to the flask backend using REST API. The response sent back contains an array of JSON objects. These objects are the top three recipes recommended by the system and will be rendered into the UI seamlessly. A picture to be displayed with the recipe was also reverse searched using the Google Custom Search API (Fig. 5).

## 5 Conclusion

The recipe recommendation system was developed using data gathered by leveraging web scraping technologies. The tools used during the cleaning, preprocessing, tokenization, and modeling were completely open source. The Query algorithm is similar to a search engine. It takes an array of ingredients as its input and returns the top three recipes which can be made using these input ingredients. A front-end web

application was designed and implemented to demonstrate the actual deployment of the assistant.

## 6 Future Scope

Apart from the recipe recommendation, several additional features can be integrated with our assistant. The assistant can interact with its user through a mobile application. One of the features which could be added is decay detection. Currently, the user needs to manually enter the items stored in the refrigerator. Using OpenCV and by including a camera as an additional hardware device, object detection can be used for automatically tracking all the items being stored. Further by tracking the date and time of storage, it will be possible to monitor if an item is being stored for unusually long periods and alert the user in case this is observed. For instance, if curd is stored for more than a day, then the assistant will alert the user. This will prevent food items from going bad. Alternatively, RFID tags could also be used. Tracking items can also be used to alert the user in case the stock of a particular frequently consumed item is about to get over. It can also provide a way to monitor the overall diet of the user depending on what type of food items are frequently consumed. This may help the user in interpreting the overall nutritional value his or her diet provides. In addition to this, using blockchains, it may also be possible to automatically order the items which are frequently stocked by the user. For this, the assistant will need to track the frequency with which order for a particular item is placed by the user.

## References

1. Cichocki, Andrzej, and P. H. A. N. Anh-Huy. "Fast local algorithms for large scale nonnegative matrix and tensor factorizations." *IEICE transactions on fundamentals of electronics, communications and computer sciences* 92.3: 708–721, 2009.
2. D. Li, M. J. Zaki and C. -H. Chen, "Nutrition Guided Recipe Search via Pre-trained Recipe Embeddings," 2021 IEEE 37th International Conference on Data Engineering Workshops (ICDEW), 2021, pp. 20–23, <https://doi.org/10.1109/ICDEW53142.2021.00011>.
3. "LG UNVEILS INTERNET-READY REFRIGERATOR". LG Electronics. [telecompaper.com](http://telecompaper.com). June 21, 2000.
4. "Online Learning for Latent Dirichlet Allocation", Matthew D. Hoffman, David M. Blei, Francis Bach, 2010
5. V. Bajaj, R. B. Panda, C. Dabs and P. Kaur, "Graph Database for Recipe Recommendations," 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2018, pp. 1–6, <https://doi.org/10.1109/ICRITO.2018.87488>
6. Z. Li, J. Hu, J. Shen, and Y. Xu, "A Scalable Recipe Recommendation System for Mobile Application," 2016 3rd International Conference on Information Science and Control Engineering (ICISCE), 2016, pp. 91–94, DOI: <https://doi.org/10.1109/ICISCE.2016.30>.

7. Zhenfeng Lei, Anwar Ul Haq, Adnan Zeb, Md Suzauddola, Defu Zhang, Is the suggested food your desired: Multi-modal recipe recommendation with demand-based knowledge graph, *Expert Systems with Applications*, Volume 186, 2021,115708 ISSN 0957-4174, <https://doi.org/10.1016/j.eswa.2021.115708>.
8. Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

# A Location-Based Cryptographic Suite for Underwater Acoustic Networks



Thota Sree Harsha, Venkata Sravani Katasani, Rajat Partani,  
and B R Chandavarkar

## 1 Introduction

Wireless sensor networks are an exciting new technology, making its presence felt across various applications [1, 2]. UAN (underwater acoustic network), which uses wireless sensor networks, is a new underwater network system that effectively protects military and marine information. UANs are in multiple fields, including monitoring the underwater environment, collecting oceanic data, exploring underwater resources, and disaster prevention.

Even though UANs have some characteristics in common with terrestrial sensor networks [3], such as a large number of nodes and low power consumption, they differ in several ways, including narrow bandwidth, long propagation delays, and node passive mobility. The following sections discuss the difficulties in developing secure algorithms for communication in UANs.

Batteries are typically used to power underwater sensor nodes since UANs are deployed underwater in remote locations. UANs are restricted by resource constraints [4] such as battery life and computational power. The propagation delay being huge and data transfer rates being low contribute to the limited bandwidth [5] of underwater acoustic networks. Because of amplitude modulation and multipath, an underwater channel can be easily disrupted during transmission. Furthermore, UANs are more susceptible to jamming and DoS attacks [6, 7] because the underwater acoustic channel is an open environment. Because of these characteristics, existing work in terrestrial sensor networks is unsuitable for UANs,

---

T. S. Harsha · V. S. Katasani (✉) · R. Partani · B. R. Chandavarkar  
National Institute of Technology Karnataka Surathkal, Computer Science & Engineering,  
Mangalore, Karnataka, India  
e-mail: [thotasreeharsha.191cs258@nitk.edu.in](mailto:thotasreeharsha.191cs258@nitk.edu.in); [sravani.191cs223@nitk.edu.in](mailto:sravani.191cs223@nitk.edu.in);  
[rajatpartani.191cs240@nitk.edu.in](mailto:rajatpartani.191cs240@nitk.edu.in)



posing dozens of new security challenges. To maintain integrity and confidentiality, security mechanisms and algorithms are in high demand.

UANs cannot employ the same security technology for terrestrial sensor networks due to differences in communication mediums and physical environments. Due to the limited energy, computation, and communication capabilities of UANs, as well as the characteristics of aqueous environments, secure communication techniques are required. Sensing data in UANs must be safely processed and managed. The data can be encrypted and transmitted to a sublayer before the application layer sends it to the next layer. When creating a new encrypted mechanism, lightweight cipher must be used. Our primary contribution is to address security issues caused by various attacks and threats such as tampering, stealing data.

Following contributions are made in the chapter. First, efficient solution for key generation using the sensor and base station geolocation in UANs is presented. Since the location of sensors may keep changing, this chapter proposes a session-based approach, where base station keeps updating the site of sensors in its cache after regular time intervals. Second, novel lightweight encryption algorithm suited for UANs is proposed. Our algorithm focuses on protecting the data even if one of the sensor nodes gets hacked by minimizing information stored in sensors. The encryption algorithm has a low communication overhead and consumes less energy, which is essential for UANs.

The rest of this chapter is structured as follows. Section 2 looks at works in the underwater security field. Section 3 delves into the overall topology of UANs. Section 4 introduces a new lightweight UAN encryption algorithm. Section 5 discusses the reliability of the proposed algorithm. Section 6 concludes our chapter and sketches out future work.

## 2 Related Work

The security of UANs has become an increasingly severe issue, but little study has been undertaken on studying security mechanisms in UANs. Due to various constraints, the research on UAN security is still in its initial stages. However, the need for UAN security technology is overgrowing to make underwater communication more secure. The following paragraphs present a few related works in UAN security-related technologies.

In [8], the authors discuss network security fundamentals and the major UAN security issues from the physical to the transport layer. The chapter then examines security approaches against common UAN security problems, protecting UAN protocols and cryptographic primitives structured for UANs, and UAN security structures that address various security issues effectively.

Since the UANs cannot use security protocols designed for WSNs and ASNs directly, [9], the authors developed a security protocol that can be used with UANs. The authors believed that they should consider countermeasures against security threats, but they did not provide specific recommendations.

The authors [10] investigated threats and attacks on UAN's security. An adversary can easily intercept sensor nodes and can tamper the information packets. UANs are vulnerable to malicious attacks due to its innate characteristics. A layered security system has several limitations against a blended attack, and the authors proposed a security mechanism and designed a layered security structure to overcome these limitations in UANs. The authors did not present any efficient algorithm to tackle the problem.

In [11], Dini and Duca addressed the issue of secure cooperation among underwater acoustic vehicles and proposed a cryptographic suite capable of reducing message overhead to the absolute minimum. Authentication, confidentiality, and integrity of messages, as well as key management, have all been provided by the cryptographic suite. In a block cipher, the ciphertext stealing (CTS) mode manipulates input data without restricting its size and generates ciphertext which is the same size as the initial plaintext. But the authors, however, limited their research to AUVs of the same broadcast domain without considering the system of multi-hop networks.

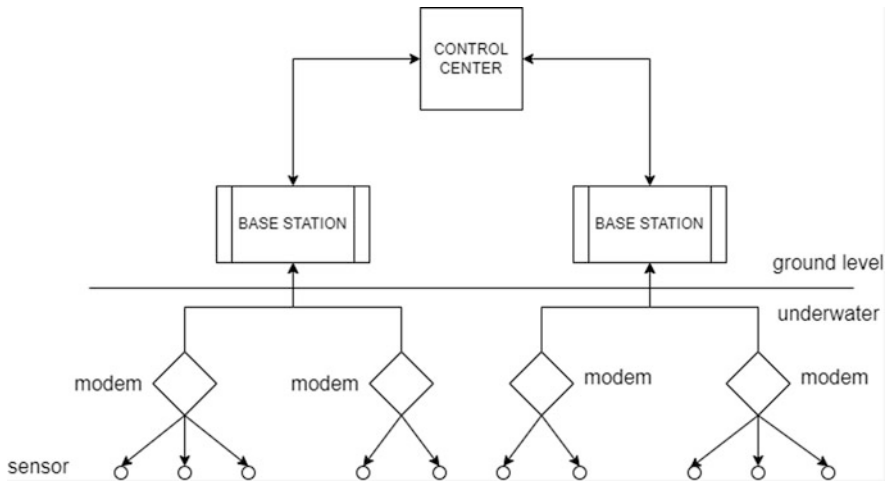
The authors of [10, 12] presented various security threats to UANs and essential requirements to combat them, but the encryption was not considered a security foundation. Kim, Ji-Eon et al. discussed a security algorithm [13] that could be used. They only looked at security research trends and UAN security issues, and they mentioned a block cipher algorithm. Still, they did not go over how to put the algorithm into practice.

Logan and Dorothy [14] proposed a general location-based encryption technique that enhances security by integrating position and time into encryption and decryption processes. They call this technique geo-encryption, but it only worked for terrestrial networks.

In this chapter, an encryption algorithm that uses the location of nodes in the key construction process is presented, which provides an additional layer of security. The aim is to make this algorithm efficient to suit the requirements of UANs. A limited number of rounds for this purpose and a block cipher technique with no ciphertext expansion are used. A strong structure in which compromising one node does not jeopardize the entire topology is proposed. The model stores only the most basic information about the topology in end nodes to accomplish this. As it is known that the location of nodes can change, a session-based approach to tackle this problem is adopted. The entire system resets every time a session ends, and nodes share their new locations with the base stations, which will be used for key generation algorithms. This also reduces the possibility of spamming attacks because the compromised node will use an entirely new key in each session.

### 3 Network Architecture

Using a set of AquaSeNT OFDM (Orthogonal Frequency Division Multiplexing) modems [15], the chapter considers an underwater acoustic network to protect all



**Fig. 1** A basic underwater acoustic network topology

assets in underwater environments. The network architecture of UANs is depicted in Fig. 1.

The network consists of control center, base station, modems, and many sensor nodes. Sensors are the end nodes of the topology. Several sensors are connected to a modem [16]. Sensors sense the surrounding state and study the environment. They send these data to the modem. The modem then ships data in packets to the base station. Every packet is divided into multiple packets by modem, each of which is 64 bits in size. The base station serves as an interface between the control center and the modem and is situated at the water surface. Maximum processing occurs at the base station. The control center is the main storage component of the entire network. It is situated on the ground and communicates only with the base station. It acts as a bridge between external networks and the UAN topology.

The modems used in the architecture have [17] limited computational and storage resources. They are very expensive and cannot be replaced frequently once they are deployed. In the proposed solution, each modem is an individual node assumed to be capable of encryption, decryption, and transmission of messages. This chapter assumes that each modem is embedded with a location tracking device capable of calculating absolute location or the relative location to the corresponding base station. To sum up, network can be considered as a hierarchical structure with a control center at the top level. It is connected to various base stations in Layer 2. Layer 3 comprises modems which in turn are connected to sensors.

## 4 Proposed Algorithm for UANs

The following section contains the complete procedure for key generation, encrypting, and decrypting messages for UANs.

### 4.1 *Communication Flow and Key Mapping*

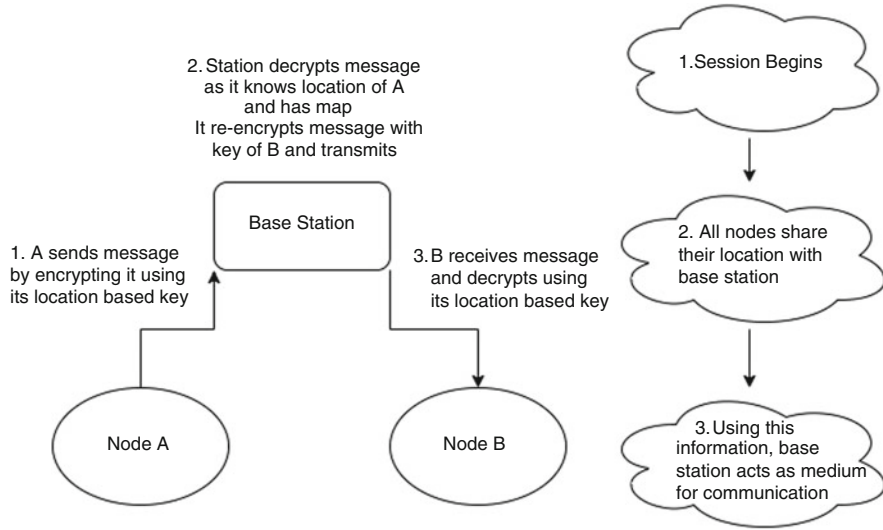
In this chapter, we want a location-based key generation system. This key should be known by both sender and receiver. To accomplish this, the chapter adopts a map-based approach. Both sender and receiver have a copy of the same map. This map stores the key value pairs of location and the secret key which will be used for encryption. A session-based approach to share location is used. Basically, at the beginning of each session, all sensor nodes transmit their current location to the base station. These locations can be used for getting the secret keys. All sensor nodes have unique maps and the base station has a copy of all the maps. All secret keys are 64 bits long. These maps are hard-coded in the sensor nodes. At the start of each session, locations are shared and the same location is used throughout the session. Communication is only allowed from sensor node to base station and vice versa. If two sensor nodes want to communicate with each other, the base station acts as an intermediate.

Also, it was mentioned that a successful attack on one of the nodes should not make the whole topology tumble. The full encryption depends on the location-based key generation map. If this map is leaked, all the messages in the topology become insecure as the secret key used for encryption can be easily found out. Such an attack is highly improbable but still a possibility. Unique maps corresponding to each modem node to get more protection are used. The base station assumed to be a powerful machine stores all the maps (corresponding to each end node) (Fig. 2).

For two end nodes to communicate [18], we assume the base station to act as an intermediate. Say node A wants to send a message to node B. This will happen as follows. Node A will first send the message to the base station. The base station then finds the secret key using the key generation map corresponding to node A. It then considers the new location key again using the node B location key generation map and xor this with the secret key, which is then conveyed to node B along with the encrypted message.

### 4.2 *Round Key Generation*

The proposed key expansion algorithm is to generate round keys for the encryption algorithm. The input is the 64-bit key generated from the key construction algorithm.



**Fig. 2** Communication flow

The key is divided into 16 blocks each of 4-bit size named  $b_1, b_2 \dots b_{16}$  as shown in Fig. 3. The intermediate round involves generation of 4 16-bit blocks  $B_1, B_2, B_3, B_4$ .  $B_i = b_i.b_i + 4.b_i + 8.b_i + 12.i = 1, 2, 3, 4$ , where  $.$  denotes the concatenation.

Cellular Automata [19] is used to generate a pseudo-random 16-bit blocks  $Q_1, Q_2, Q_3, Q_4$  using  $B_1, B_2, B_3, B_4$  as inputs. The output blocks  $Q_1, Q_2, Q_3, Q_4$  are concatenated to generate the required round key. This round key is used as an input for the next round key generation.

### 4.3 Encryption Algorithm

Our algorithm uses Feistel cipher structure [20] with additional layers for an added security. As the computational power of the sensor nodes is limited, the chapter presents this lightweight algorithm that maximizes security while compromising on resources.

The function  $F$  used in the algorithm is described in Figs. 4 and 5.

### 4.4 Decryption Algorithm

The decryption algorithm works in the exact opposite direction of encryption. All operations are the same but happen in the reverse order.

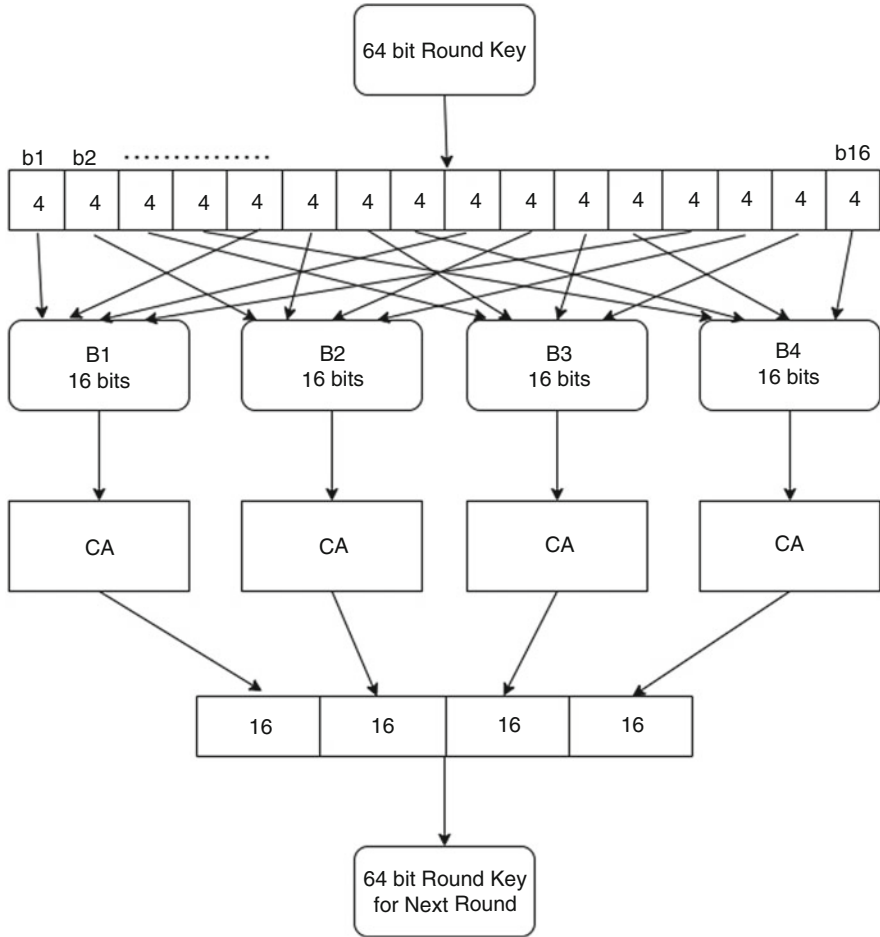


Fig. 3 Round key generation

### 5 Comparative Study and Performance Analysis

For any efficient adversary on attack encryption scheme, there are four attack models [21]: ciphertext-only attack, chosen-plaintext attack, chosen-ciphertext attack, and known-plaintext attack. If an adversary wishes to use a measure known as a brute-force attack or an exhaustive key search on ciphertext-only attack, the key space must be large enough. The security algorithm proposed in this chapter uses the 64-bit length of each block and adds up to a 10-round iteration.

The round key has a 64-bit value. The round key is mutative in every round. The adversary cannot decrypt the message even if the current key is intercepted. Ten-

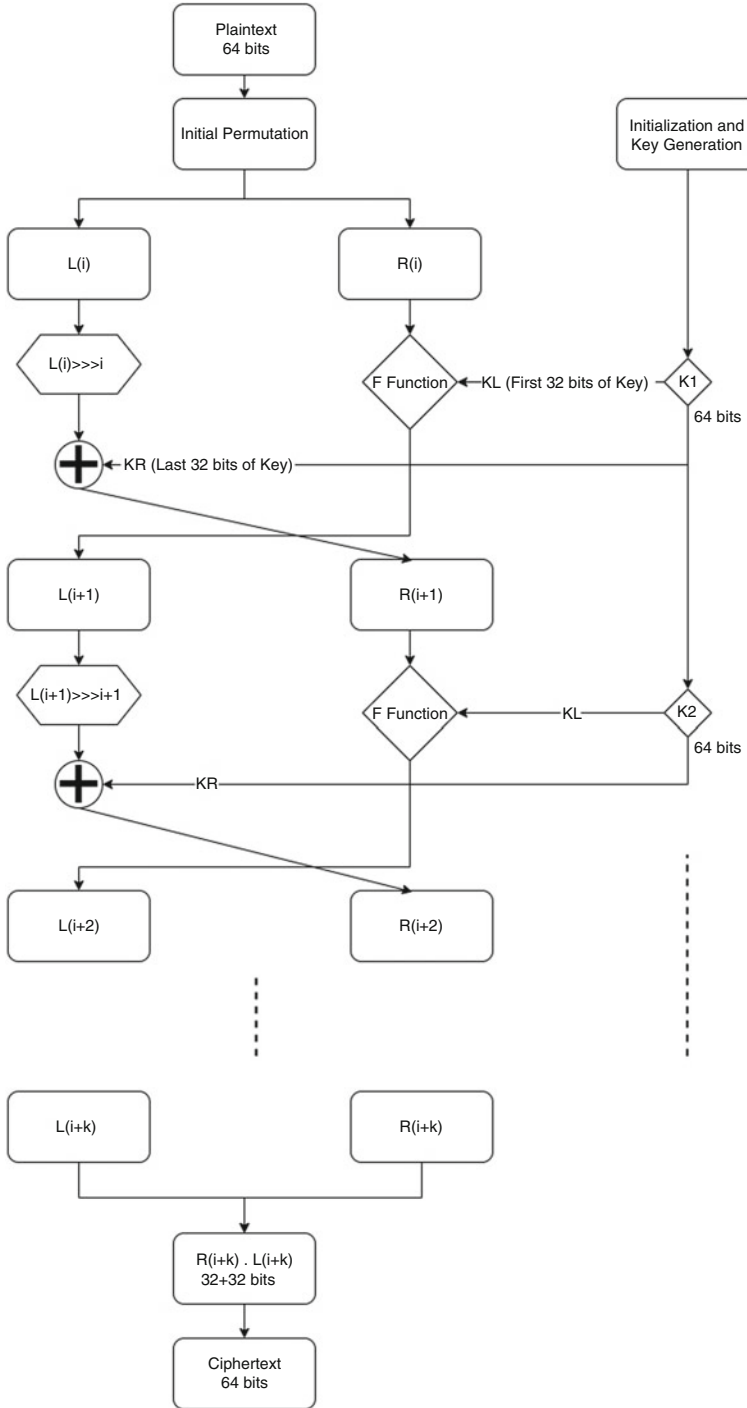
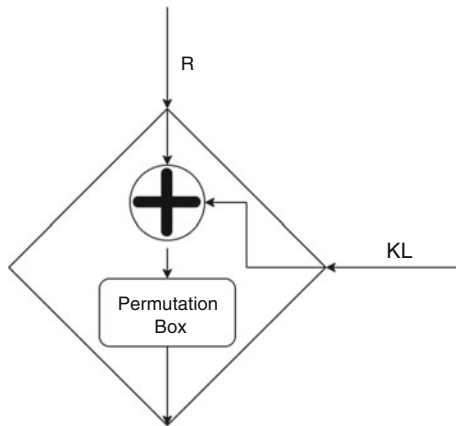


Fig. 4 F function

Fig. 5 Encryption algorithm




---

**Algorithm 1** Encryption Algorithm

---

**Input:** plaintext, Roundkey, P, initial\_Permutation  
 $plaintext' \leftarrow initial\_Permutationonplaintext$   
 $L \leftarrow plaintext'[0 : 32]$   
 $R \leftarrow plaintext'[32 : 64]$   
**for all**  $i \in range(1, 11)$  **do**  
     $KL \leftarrow RoundKey_i[0 : 32]$   
     $KR \leftarrow RoundKey_i[32 : 64]$   
     $R' \leftarrow R \oplus KL_i$   
     $L_{i+1} \leftarrow PonR'$   
     $L' \leftarrow L_i \gg i$   
     $R_{i+1} \leftarrow L' \oplus KR_i$   
**end for**  
 $ciphertext \leftarrow R_{10} + L_{10}$

---



---

**Algorithm 2** Decryption Algorithm

---

**Input:** ciphertext, inverse\_IP\_matrix, RoundKey, inverse\_P  
 $L \leftarrow ciphertext[32 : 64]$   
 $R \leftarrow ciphertext[0 : 32]$   
**for all**  $i \in range(0, 10)$  **do**  
     $KL \leftarrow RoundKey_{10-i}[0 : 32]$   
     $KR \leftarrow RoundKey_{10-i}[32 : 64]$   
     $R' \leftarrow inverse\_PonL_{10-i}$   
     $R_{9-i} \leftarrow R' \oplus KL_{10-i}$   
     $L' \leftarrow R_{10-i} \oplus KR_{10-i}$   
     $L_{9-i} \leftarrow L' \ll 10 - i$   
**end for**  
 $plaintext' \leftarrow L_1 + R_1$   
 $plaintext \leftarrow inverse\_IP\_matrixonplaintext'$

---



**Table 1** Comparison of various algorithms

Algorithm	— Key size	— Block size	— Number of rounds
Blowfish	32-448	64	16
AES-128	128	128	10
DES	64	64	16
PRESENT	80	64	31
Our algorithm	64	64	10

round encrypted operation is carried out. If an adversary illegally breaks one of the round encryption.

Each round's kind of combination is about  $2^{64}$ , and if the encryption algorithm has 10 rounds, it can be broken in  $2^{64*10}$  time. A powerful computer can search at a rate of 10 power 11 per second. Then, it will take at least  $4.562441e+181$  years to complete the exhaustive search (Table 1).

The blowfish algorithm is a significantly faster encryption method [22] with no effective cryptanalysis technique found to date. Yet it is not suitable for the UAN topology. It makes use of 4 substitution boxes with 512 entries of 32 bits each. Thus, a total of 65,536 bits are required to store the s-boxes alone. Then, in each encryption step, s-box lookups are required multiple times which makes it computationally infeasible for a UAN end node. The proposed algorithm on the other hand completely eliminates the usage of s-boxes, thus saving memory space as well as reducing the computational need of the algorithm. This makes it better than the blowfish algorithm for the UAN topology.

The AES is even more computationally expensive than blowfish. Moreover, it encrypts plaintext in block sizes of 128 bits [23], which is more than what is required by the UAN sensor nodes. These factors make it unsuitable for the UANs. Similar to blowfish, the DES algorithm makes use of 8 s-boxes which are expensive to store in each end node.

PRESENT was developed as an improvement over AES [24] because AES was not suitable for extremely constrained environments. Though PRESENT claims to be a lightweight network, it is based on an SP network consisting of many S-boxes. Also, in each round, one s-box is applied 16 times in parallel. This requires heavy computational capabilities. PRESENT uses a total of 31 rounds for the encryption process and likewise for decryption. This also proves to be time-consuming. The proposed algorithm goes for only 10 rounds and avoids the use of the s-box which helps when dealing with very less memory. Also, the key size of PRESENT is 80 bits which then needs to be mapped to a 64-bit block size. The proposed algorithm does not have any such issues because the generated key size itself is 64 bits and can be directly used.

So finally, the proposed algorithm is based on all of these algorithms and tries to improve on their drawbacks. It took inspiration from AES to use 10 rounds and uses a reduced key and block size to increase encryption. Blowfish was not suitable for UAN topologies and proved that very less processing should occur at the end nodes.

PRESENT was time-consuming because of its *s*-boxes. So the improvement on the algorithm is to eliminate the usage of the *s*-box.

The main takeaway from the proposed algorithm is the usage of the location to encrypt data. The reason for choosing location is because the algorithm required some criteria to base our encryption while maintaining the overhead information involved. Also, the criteria that we use should be dynamic to benefit the session-based approach. Relative location proved handy in all these dimensions. It also makes the passive mobility of nodes look like an advantage in the session scenario.

## 6 Conclusion and Future Work

The chapter presented a novel approach for secure communication in UANs topology. UANs are subjected to various memory and computational constraints. Even the environment in which sensor nodes of a UAN topology are placed is very harsh and keeps on changing. The location of sensor nodes as well as the speed of sound keeps changing. The chapter first gives some protocols for communication, that is all communication between end nodes is done via the base station which is a powerful machine with ample amount of memory and computation resources. The chapter treats the base station as the heart of all communication. The changing location of sensor nodes which presents difficulty in communication is taken advantage of by adding a location parameter in the key forming process. A lightweight yet secure encryption algorithm is presented. A theoretical analysis of the algorithm proves it to be better than the existing approaches as well as within the computation limits of the UAN sensor nodes.

Future plan is to simulate the algorithm in the same environment as the UAN topology. The presented algorithm will be compared against the existing approaches. The communication suite will also be tested against various cyber attacks such as DDOS attacks. The results will be analyzed thoroughly on the basis of which some improvements on the existing approach will be made.

Link to Implementation: <https://github.com/raj1701/location-based-encryption-strategy.git>

## References

1. E. H. Callaway, *Wireless Sensor Networks, Architectures and Protocols*, <https://www.routledge.com/Wireless-Sensor-Networks-Architectures-and-Protocols/Callaway-Jr/p/book/9780849318238> (2003).
2. Ramson et al. "Applications of wireless sensor networks — A survey." 2017 International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT) (2017): 325–329.
3. Jiang et al. "UNDERWATER ACOUSTIC NETWORKS-ISSUES AND SOLUTIONS." (2008).

4. Yisa et al. "Security challenges of Internet of Underwater Things: A systematic literature review, *Transactions on Emerging Telecommunications Technologies* 32 (2021): n. pag.
5. Jiang et al. "On Reliable Data Transfer in Underwater Acoustic Networks: A Survey From Networking Perspective." *IEEE Communications Surveys Tutorials* 20 (2018): 1036–1055.
6. Zuba et al. "Vulnerabilities of underwater acoustic networks to denial-of-service jamming attacks." *Secur. Commun. Networks* 8 (2015): (2015) 2635–2645.
7. Yang et al. "Challenges, Threats, Security Issues and New Trends of Underwater Wireless Sensor Networks." *Sensors (Based, Switzerland)* 18 (2018): n. pag.
8. Jiang Shengming. "On Securing Underwater Acoustic Networks: A Survey." *IEEE Communications Surveys Tutorials* 21 (2019): 729–752.
9. Paar et al. "New Designs in Lightweight Symmetric Encryption." (2008).
10. Cong et al. "Security in Underwater Sensor Network." *2010 International Conference on Communications and Mobile Computing I* (2010): 162–168.
11. Dini and Duca. "A cryptographic suite for underwater cooperative applications." *2011 IEEE Symposium on Computers and Communications (ISCC)* (2011): 870–875.
12. Dong et al. "Security Considerations of Underwater Acoustic Networks." (2010).
13. Kim et al. "Security in Underwater Acoustic Sensor Network: Focus on Suitable Encryption Mechanisms." *AsiaSim* (2012).
14. Scott et al. "A Location Based Encryption Technique and Some of Its Applications." (2003).
15. Zhou et al. "OFDM for Underwater Acoustic Communications." (2014).
16. Akyildiz et al. "Underwater acoustic sensor networks: research challenges." *Ad Hoc Networks* 3 (2005): 257–279.
17. Sendra et al. "Underwater Acoustic Modems." *IEEE Sensors Journal* 16 (2016): 4063–4071.
18. Saeed et al. "Underwater Optical Wireless Communications, Networking, and Localization: A Survey." *Ad Hoc Networks* 94 (2019): n. pag.
19. Dhingra et al. "Comparison of LFSR and CA for BIST." (2005).
20. Knudsen et al. "Practically Secure Feistel Cyphers." *FSE* (1993).
21. Kendhi et al. "A Survey Report on Various Cryptanalysis Techniques." (2013).
22. Qingxin et al. "Analysis of Blowfish cryptography." *Journal of Computer Applications* (2007): n. pag.
23. Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data
24. Bogdanov et al. "PRESENT: An Ultra-Lightweight Block Cipher." *CHES* (2007).

# Index

## A

Adithya Rajesh, C., 445–454  
Agarwal, R., 9–19  
Agrawal, C., 323–339  
Akarte, M., 33–42  
Alamin-Ul-Islam, Md., 455–466  
Anand, A., 341–352  
Anand, S.K., 69–83  
Anantwar, V., 57–66  
Aney, Y., 57–66  
Antony, R.T., 377–390  
Ashwanth, S., 261–283  
Ashwin, P., 405–421

## B

Babu, S.R., 175–188  
Badole, I., 191–200  
Bandi, Y., 499–509  
Banerjee, B., 489–497  
Bansod, U., 323–339  
Bhadoriya, S.S., 499–509  
Bhatlawande, S., 57–66  
Bhola, A.K., 45–54

## C

Chaithanya Shyam, D., 445–454  
Chandavarkar, B.R., 69–83, 139–173,  
233–245, 341–352, 357–366, 367–375,  
377–390, 393–402, 405–421, 425–433,  
445–454, 469–486, 511–521  
Chheda, S., 191–200  
Chitre, O., 191–200

## D

Date, H., 285–309  
Deshmukh, S., 499–509  
Deshpande, H., 129–136  
Dhody, I.S., 139–173  
Divya, C., 99–109  
Dixit, A.K., 33–42

## G

Gadagkar, A.V., 69–83  
Gaikwad, A., 57–66  
Garg, T., 9–19  
Goyal, V., 9–19  
Guduru, S., 341–352  
Gupta, R.K., 435–443

## H

Harsha, T.S., 511–521

## I

Islam, A., 45–54

## J

Jain, A., 139–173  
Jena, S., 393–402  
Jetawat, A.K. Dr., 247–258  
Johora, F.T., 455–466  
Joshi, P., 367–375  
Joshitha Reddy, D., 377–390

**K**

Kalantri, R., 1–7  
 Kalbande, D., 191–200  
 Kamediya, S., 367–375  
 Kapadia, S., 221–229  
 Karande, A., 113–126  
 Karani, R., 221–229  
 Katasani, V.S., 511–521  
 Ketkar, M.M., 357–366  
 Kirkire, S., 499–509  
 Kowsalya, V., 99–109  
 Kumar, A., 313–320  
 Kumar, P., 435–443  
 Kumar, P.V., 69–83  
 Kumar, Rahul, 405–421  
 Kumar, Ritik, 367–375  
 Kumar, Saurav, 33–42  
 Kumar, Sudhir, 435–443

**L**

Lal, A., 233–245  
 Lobo, J., 1–7

**M**

Manyatha, A.P., 261–283  
 Meena, A., 393–402  
 Misra, R., 313–320, 435–443  
 Mohania, M., 9–19  
 Mohiuddin, K., 45–54  
 More, N., 489–497

**N**

Nagar, L., 233–245  
 Naik, G., 129–136  
 Nath, D., 323–339  
 Naveen, B., 405–421  
 Nikam, V.B., 489–497  
 Nisar, M., 129–136

**P**

Pai, P.G., 357–366  
 Pai, R., 203–218  
 Parekh, V., 221–229  
 Partani, R., 421  
 Pooja, G., 425–433  
 Poojitha, S.L., 377–390  
 Pradhan, S., 1–7  
 Praladhka, U., 221–229  
 Prateek, L.A., 469–486  
 Pranav, D.V., 445–454  
 Purohit, M., 203–218

**R**

Rahman, M.M., 455–466  
 Rajguru, S., 1–7  
 Rana, A., 357–366  
 Rao, B., 113–126  
 Rasool, M.A., 45–54

**S**

Sahoo, D.K., 1–7  
 Saji, R., 69–83  
 Sandya, J.K., 261–283  
 Sangle, R., 247–258  
 Shah, B., 129–136  
 Shah, R., 469–486  
 Sharmila, V.C., 175–188, 261–283  
 Shilaskar, S., 57–66  
 Shruthi, M., 175–188  
 Siddamsetti, S.G., 425–433  
 Simon, S., 285–309  
 Singh, I., 139–173  
 Singh, S., 233–245  
 Singh, S.K., 21–31  
 Singh, T.N., 313–320  
 Singh, V., 313–320  
 Singhal, M., 341–352  
 Siroya, D., 129–136  
 Sithartha, M.A., 137  
 Srivastava, R., 21–31  
 Sudhama, K.K., 425–433  
 Sundarrajan, S., 393–402

**T**

Tony, A., 469–486

**U**

Unadkat, U., 221–229

**V**

Vinayakray-Jani, P., 203–218  
 Vishwakarma, M., 87–97  
 Vyas, R., 499–509

**Y**

Yadav, S., 323–339  
 Yesmin, F., 455–466