

Lecture Notes in Networks and Systems 632

Valentina Emilia Balas
Vijay Bhaskar Semwal
Anand Khandare *Editors*

Intelligent Computing and Networking

Proceedings of IC-ICN 2022

 Springer

Lecture Notes in Networks and Systems

Volume 632

Series Editor

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

Advisory Editors

Fernando Gomide, Department of Computer Engineering and Automation—DCA,
School of Electrical and Computer Engineering—FEEC, University of
Campinas—UNICAMP, São Paulo, Brazil

Okyay Kaynak, Department of Electrical and Electronic Engineering,
Bogazici University, Istanbul, Türkiye

Derong Liu, Department of Electrical and Computer Engineering, University of
Illinois at Chicago, Chicago, USA

Institute of Automation, Chinese Academy of Sciences, Beijing, China

Witold Pedrycz, Department of Electrical and Computer Engineering, University of
Alberta, Alberta, Canada

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

Marios M. Polycarpou, Department of Electrical and Computer Engineering,
KIOS Research Center for Intelligent Systems and Networks, University of Cyprus,
Nicosia, Cyprus

Imre J. Rudas, Óbuda University, Budapest, Hungary

Jun Wang, Department of Computer Science, City University of Hong Kong,
Kowloon, Hong Kong

The series “Lecture Notes in Networks and Systems” publishes the latest developments in Networks and Systems—quickly, informally and with high quality. Original research reported in proceedings and post-proceedings represents the core of LNNS.

Volumes published in LNNS embrace all aspects and subfields of, as well as new challenges in, Networks and Systems.

The series contains proceedings and edited volumes in systems and networks, spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution and exposure which enable both a wide and rapid dissemination of research output.

The series covers the theory, applications, and perspectives on the state of the art and future developments relevant to systems and networks, decision making, control, complex processes and related areas, as embedded in the fields of interdisciplinary and applied sciences, engineering, computer science, physics, economics, social, and life sciences, as well as the paradigms and methodologies behind them.

Indexed by SCOPUS, INSPEC, WTI Frankfurt eG, zbMATH, SCImago.

All books published in the series are submitted for consideration in Web of Science.

For proposals from Asia please contact Aninda Bose (aninda.bose@springer.com).

Valentina Emilia Balas · Vijay Bhaskar Semwal ·
Anand Khandare
Editors

Intelligent Computing and Networking

Proceedings of IC-ICN 2022

 Springer

Editors

Valentina Emilia Balas
Aurel Vlaicu University
Arad, Romania

Vijay Bhaskar Semwal
National Institute of Technology
Bhopal, Madhya Pradesh, India

Anand Khandare
Thakur college of Engineering
and Technology
Mumbai, India

ISSN 2367-3370

ISSN 2367-3389 (electronic)

Lecture Notes in Networks and Systems

ISBN 978-981-99-0070-1

ISBN 978-981-99-0071-8 (eBook)

<https://doi.org/10.1007/978-981-99-0071-8>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 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 Singapore Pte Ltd.

The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

“International Conference on Intelligent Computing and Networking (IC-ICN-2022)” is a platform for conducting conferences with the objective of strengthening the research culture by bringing together academicians, scientists, and researchers in the domain of intelligent computing and Networking.

The 14th annual event IC-ICN-2022 in the series of international conferences organized by Thakur College of Engineering and Technology (TCET) under the umbrella of MULTICON ever since the first event of International Conference and Workshop on Emerging Trends in Technology was conducted online and offline on 25th and 27th February 2022. The IC-ICN-2022 event is organized with the insightfulness of providing not only a great platform to think innovatively, but also bringing in sync the theory and applications in the field of Intelligent Computing and Networking for the students, faculty, scientists, researchers from the industry as well as the research scholars. This platform is an efficacious link for the students/authors/researchers to collaborate and enhance the network with peer universities and institutions in India and abroad in the respective domain. The basic aim is to hold the conference where the participants present their Research Papers, Technical Papers, Case Studies, Best and Innovative Practices, Engineering Concepts and Designs so that the applied study or research can be sop up into the real world. Technological development in the domain of intelligent computing and Networking is the need of the hour, which will simplify our life in an eco-friendly environment with better connectivity and security and this conference facilitates the pathway to the purpose.

Not just inculcating the research culture, IC-ICN 2022 has gained wide publicity through website, social media coverage as well as the vigorous promotion by the team of faculty members to the various colleges. The IC-ICN-2022 has an affiliation with Scopus Indexed journals for intelligent systems, leading publication house and conference proceeding with ISBN number.

Participants applauded TCET for making the event successful and appreciated its sound belief in building a strong relationship and bonding by taking care of each and every participant’s requirement throughout the event. The two days event comprises conferences and workshops with multiple tracks. During these two days, there were 200 presentations by national as well as international researchers and

industrial personnel. Also, the idea presentations with deliberation by the delegates were part of the event. We are grateful for the efforts of all the members of the organizing and editorial committee for supporting the event and extending their cooperation to make it a grand success.

Team-ICICN-2022

Arad, Romania
Bhopal, India
Mumbai, India

Valentina Emilia Balas
Vijay Bhaskar Semwal
Anand Khandare

Contents

Implementation of a PID Controller for Autonomous Vehicles with Traffic Light Detection in CARLA	1
Shivanshu Shrivastava, Anuja Somthankar, Vedant Pandya, and Megharani Patil	
Binary Classification for High Dimensional Data Using Supervised Non-parametric Ensemble Method	15
Nandan Kanvinde, Abhishek Gupta, Raunak Joshi, and Pinky Gerela	
Deep Linear Discriminant Analysis with Variation for Polycystic Ovary Syndrome Classification	25
Raunak Joshi, Abhishek Gupta, Himanshu Soni, and Ronald Laban	
Improved Helmet Detection Model Using YOLOv5	35
Premanand Ghadekar, Shreyas Mendhekar, Vallabh Niturkar, Sanika Salunke, Abhinav Shambharkar, and Kshitij Taley	
Stock Market Trend Prediction Along with Twitter Sentiment Analysis	45
Priyadarshan Dhabe, Ayush Chandak, Om Deshpande, Pratik Fandade, Naman Chandak, and Yash Oswal	
A Study on MQTT Protocol Architecture and Security Aspects Within IoT Paradigm	61
M. Nimavat Dhaval and G. Raiyani Ashwin	
Comparative Analysis of Different Block Chain Technology to Improve the Security in Social Network	73
Niki Modi	
Euphonia: Music Recommendation System Based on Facial Recognition and Emotion Detection	85
Eliganti Ramalakshmi, Huma Hussain, and Kritika Agarwal	

Improvement of Makespan and TCTime in Dynamic Job Ordering and Slot Utilization for MapReduce Workloads	95
Tanmayi Nagale	
Identification and Detection of Plant Disease Using Transfer Learning	111
Neelam Sunil Khasgiwala and R. R. Sedamkar	
Blockchain Based E-Voting System	123
Mahima Churi, Anmol Bajaj, Gurleen Pannu, and Megharani Patil	
An Intelligent Voice Assistant Engineered to Assist the Visually Impaired	143
Rishabh Chopda, Aayan Khan, Anuj Goenka, Dakshal Dhere, and Shiwani Gupta	
Analysis of Python Libraries for Artificial Intelligence	157
Anand Khandare, Nipun Agarwal, Amruta Bodhankar, Ankur Kulkarni, and Ishaan Mane	
Annual Rainfall Prediction of Maharashtra State Using Multiple Regression	179
Loukik S. Salvi and Ashish Jadhav	
Automated Healthcare System Using AI Based Chatbot	191
Akshay Mendon, Megharani Patil, Yash Gupta, Vatsal Kadakia, and Harsh Doshi	
Winner Prediction of Football Match Using Machine Learning	207
Shailja Jadon, Aman Jain, Prathamesh Bagal, Kunal Bhatt, and Manish Rana	
RaktaSeva—An App for Civilians and Blood Banks	219
Akash Singh, Vidhi Punjabi, Samiksha Bedekar, and Anand Khandare	
Prediction of Anemia Disease Using Machine Learning Algorithms	229
Aditya Dixit, Rahul Jha, Raunak Mishra, and Sangeeta Vhatkar	
Author Index	239

Editors and Contributors

About the Editors

Valentina Emilia Balas is currently Full Professor at “Aurel Vlaicu” University of Arad, Romania. She is author of more than 300 research papers. Her research interests are in Intelligent Systems, Fuzzy Control, Soft Computing. She is Editor-in-Chief to International Journal of Advanced Intelligence Paradigms (IJAIP) and to IJCSE. Dr. Balas is member of EUSFLAT, ACM and a SM IEEE, member in TC – EC and TC-FS (IEEE CIS), TC – SC (IEEE SMCS), Joint Secretary FIM.

Vijay Bhaskar Semwal is working as Assistant professor (CSE) at NIT Bhopal since 5 February 2019. Before joining NIT Bhopal, he was working at NIT Rourkela. He has also worked with IIIT Dharwad as Assistant Professor (CSE) for 2 years (2016–2018) and he has also worked as Assistant Professor (CSE) at NIT Jamshedpur. He has earned his doctorate degree in robotics from IIIT Allahabad (2017), M.Tech. in Information Technology from IIIT Allahabad (2010) and B.Tech. (IT) from College of Engineering Roorkee (2008). His areas of research are Bipedal Robotics, Gait Analysis and synthesis, Artificial Intelligence, Machine Learning and Theoretical Computer Science. He has published more than 15 SCI research papers. He has received early career research award by DST-SERB under Government of India.

Dr. Anand Khandare Associate Professor & Dy. HOD, Computer Engineering, Thakur college of engineering and Technology, Mumbai with 17 years of teaching experience. He completed Ph.D. in Computer Science and Engineering in the domain Data Clustering in Machine Learning from Sant Gadge Baba Amravati University. He has total 50+ publications in national, international conferences and journals. He has 1 copyright and 2 patents. He guided various research and funded projects. He is worked as volume editor in springer conference on Intelligent Computing and Networking 2021. He is also reviewer in various journal and conferences.

Contributors

Kritika Agarwal Department of Information Technology, Chaitanya Bharathi Institute of Technology, Hyderabad, India

Nipun Agarwal Computer Department, Thakur College of Engineering and Technology, Mumbai, India

Prathamesh Bagal Thakur College of Engineering & Technology, Mumbai, Maharashtra, India

Anmol Bajaj Thakur College of Engineering and Technology, Mumbai, India

Samiksha Bedekar Department of Computer Engineering, Thakur College of Engineering and Technology, Mumbai, India

Kunal Bhatt Thakur College of Engineering & Technology, Mumbai, Maharashtra, India

Amruta Bodhankar Computer Department, Thakur College of Engineering and Technology, Mumbai, India

Ayush Chandak Department of Information Technology, Vishwakarma Institute of Technology, Pune, Maharashtra, India

Naman Chandak Department of Information Technology, Vishwakarma Institute of Technology, Pune, Maharashtra, India

Rishabh Chopda Thakur College of Engineering & Technology, Mumbai, Maharashtra, India

Mahima Churi Thakur College of Engineering and Technology, Mumbai, India

Om Deshpande Department of Information Technology, Vishwakarma Institute of Technology, Pune, Maharashtra, India

Priyadarshan Dhabe Department of Information Technology, Vishwakarma Institute of Technology, Pune, Maharashtra, India

Dakshal Dhere Thakur College of Engineering & Technology, Mumbai, Maharashtra, India

Aditya Dixit IT Department, Thakur College of Engineering and Technology, Mumbai, India

Harsh Doshi Department of Computer Engineering, Thakur College of Engineering and Technology, Mumbai, India

Pratik Fandade Department of Information Technology, Vishwakarma Institute of Technology, Pune, Maharashtra, India

Pinky Gerela Thakur Institute Of Management Studies, Mumbai, India

Premanand Ghadekar Department of Information Technology, Vishwakarma Institute of Technology, Pune, India

Anuj Goenka Thakur College of Engineering & Technology, Mumbai, Maharashtra, India

Abhishek Gupta University of Mumbai, Mumbai, India

Shiwani Gupta Thakur College of Engineering & Technology, Mumbai, Maharashtra, India

Yash Gupta Department of Computer Engineering, Thakur College of Engineering and Technology, Mumbai, India

Huma Hussain Department of Information Technology, Chaitanya Bharathi Institute of Technology, Hyderabad, India

Ashish Jadhav Ramrao Adik Institute of Technology, Nerul D.Y. Patil Deemed to be University, Navi Mumbai, India

Shailja Jadon Thakur College of Engineering & Technology, Mumbai, Maharashtra, India

Aman Jain Thakur College of Engineering & Technology, Mumbai, Maharashtra, India

Rahul Jha IT Department, Thakur College of Engineering and Technology, Mumbai, India

Raunak Joshi University of Mumbai, Mumbai, India

Vatsal Kadakia Department of Computer Engineering, Thakur College of Engineering and Technology, Mumbai, India

Nandan Kanvinde Thakur Institute Of Management Studies, Mumbai, India

Aayan Khan Thakur College of Engineering & Technology, Mumbai, Maharashtra, India

Anand Khandare Department of Computer Engineering, Thakur College of Engineering and Technology, Mumbai, India

Neelam Sunil Khasgiwala Computer Engineering Department, Thakur College of Engineering & Technology, Mumbai, India

Ankur Kulkarni Computer Department, Thakur College of Engineering and Technology, Mumbai, India

Ronald Laban St. John College of Engineering and Management, Palghar, India

Ishaan Mane Computer Department, Thakur College of Engineering and Technology, Mumbai, India

Shreyas Mendhekar Department of Information Technology, Vishwakarma Institute of Technology, Pune, India

Akshay Mendon Department of Electronics and Telecommunication, Thakur College of Engineering and Technology, Mumbai, India

Raunak Mishra IT Department, Thakur College of Engineering and Technology, Mumbai, India

Niki Modi Princeton University, Princeton, NJ, USA;
Springer Heidelberg, Tiergartenstr. 17, Heidelberg, Germany

Tanmayi Nagale Thakur College of Engineering and Technology, Mumbai, Kandivali (E)Maharashtra, India

M. Nimavat Dhaval RK University, Rajkot, India

Vallabh Niturkar Department of Information Technology, Vishwakarma Institute of Technology, Pune, India

Yash Oswal Department of Information Technology, Vishwakarma Institute of Technology, Pune, Maharashtra, India

Vedant Pandya Department of Computer Engineering, Thakur College of Engineering & Technology, Mumbai, India

Gurleen Pannu Thakur College of Engineering and Technology, Mumbai, India

Megharani Patil Thakur College of Engineering and Technology, Mumbai, India

Vidhi Punjabi Department of Computer Engineering, Thakur College of Engineering and Technology, Mumbai, India

G. Raiyani Ashwin Nirma University, Ahmedabad, India

Eliganti Ramalakshmi Department of Information Technology, Chaitanya Bharathi Institute of Technology, Hyderabad, India

Manish Rana Thakur College of Engineering & Technology, Mumbai, Maharashtra, India

Sanika Salunke Department of Information Technology, Vishwakarma Institute of Technology, Pune, India

Loukik S. Salvi Ramrao Adik Institute of Technology, Nerul D.Y. Patil Deemed to be University, Navi Mumbai, India

R. R. Sedamkar Computer Engineering Department, Thakur College of Engineering & Technology, Mumbai, India

Abhinav Shambharkar Department of Information Technology, Vishwakarma Institute of Technology, Pune, India

Shivanshu Shrivastava Department of Computer Engineering, Thakur College of Engineering & Technology, Mumbai, India

Akash Singh Department of Computer Engineering, Thakur College of Engineering and Technology, Mumbai, India

Anuja Somthankar Department of Computer Engineering, Thakur College of Engineering & Technology, Mumbai, India

Himanshu Soni St. John College of Engineering and Management, Palghar, India

Kshitij Taley Department of Information Technology, Vishwakarma Institute of Technology, Pune, India

Sangeeta Vhatkar IT Department, Thakur College of Engineering and Technology, Mumbai, India

Implementation of a PID Controller for Autonomous Vehicles with Traffic Light Detection in CARLA



Shivanshu Shrivastava , Anuja Somthankar , Vedant Pandya ,
and Megharani Patil

Abstract In the last decade, self-driving cars have witnessed a meteoric rise in popularity due to exceptional research in the fields of Edge Computing and Artificial Intelligence. Nowadays, autonomous vehicles use elaborate mathematical models in tandem with sophisticated Deep Learning techniques to navigate safely. PID Controllers have been used ubiquitously by researchers for autonomous vehicles. Deep Learning techniques like YOLO allow autonomous vehicles to be able to detect a wide range of objects in their surroundings leading to better responses. In this paper, a PID controller has been implemented to navigate a vehicle in CARLA Simulator. A Custom Traffic Light detection model has also been integrated with the controller to respond to traffic lights in the path of the vehicle.

Keywords PID Controller · Object Detection · Deep Learning · Autonomous Vehicle · Traffic Light Detection · CARLA Simulator

1 Introduction

Vehicles that are equipped with advanced technologies that assist humans to control the vehicle or control the vehicle autonomously such that no human interaction is required are called Autonomous vehicles [1, 2]. The control decision that is generated by an Autonomous Vehicle are Speed or Throttle, Steering Angle, Brake, Lane Changing, and Parking concerning the requirements of the trajectory and the perception of the environment [3, 4]. As per the National Highway Traffic Safety Administration in America 2013, the automation of the vehicle can be classified into five levels [5].

1. Level 0 (No Automation)—There is no automation and the driver controls the vehicle using brakes, steering angle, and throttle.

S. Shrivastava (✉) · A. Somthankar · V. Pandya · M. Patil
Department of Computer Engineering, Thakur College of Engineering & Technology, Mumbai,
India
e-mail: shivanshu123shrivastav@gmail.com

2. Level 1 (Function Specific Automation)—It includes functions related to controls like brakes or stability or any other specific function
3. Level 2 (Combined Function Automation)—This includes a combination of multiple controls like lane maneuvering with cruise control.
4. Level 3 (Limited Self-Driving Automation)—In this level, the entire vehicle can navigate autonomously under the monitoring of the driver for occasional control.
5. Level 4 (Full Self-driving Automation)—This level of automation only expects the start and endpoint for the journey else everything will be controlled by the autonomous vehicle itself.

Nowadays, Level 2 automated vehicles are already present in the industry which can easily perform tasks like cruise control for maintaining speed and lane centering, with research still ongoing for improving their performance.

There are 4 common steps that are followed to achieve this autonomy, as stated in [6]:

1. Perception—Understanding the surroundings.
2. Planning—Taking decisions about the appropriate trajectories.
3. Control—Mapping the interactions in terms of force and energy in the real world.
4. Co-ordination—Sharing trajectories with other autonomous vehicles to make navigation safer.

Perception in AV via Camera is done by Object Detection which can be done by various Deep Learning Techniques. One of the techniques is CNN i.e., Convolution Neural Networks have been extensively applied to tasks of image classification and computer vision and they have proven very useful [7] CNN architecture progressively develops features via backpropagation where the last layer consists of the output [8]. CNN models are very difficult to train in both the parameters i.e., data and computational power, hence one uses pre-trained CNN architectures to design novel architecture using transfer learning [9]. The operational environment of AV is dynamic, as CNN is robust to both transitional invariance and rotational invariance, they become a suitable fit for the task at hand.

YOLO i.e. You Only Look Once algorithm [10] is one of the object detection model architectures that can be used in real-time with high accuracy. Various versions of YOLO are available and with every version, the speed and the accuracy are improved [11].

2 Related Work

Authors in [12] explain the automation of car in five basic steps, which are perception, localization, planning, control, and system management using technologies like LiDAR, Radar, and Vision. [13] grouped multiple deep learning autonomous driving systems and introduced deep learning methods for visual localization while comparing the two most popular sensors, LiDAR and Vision. In [14], a survey was

conducted which mapped the socio-economic standards of the respondents. The survey was made comparable with international studies. DeepPicar: A Low-cost Deep Neural Network-based Autonomous Car was developed by [15].

According to [6], Engineers continued to eliminate the Steady Error term by resetting the point to some hypothetical value as long as the error wasn't zero. Proportional Integral Controller was the term coined for the part of resetting the integrated error. As mentioned in [16], in 1940, TIC introduced a controller with a derivative action that reduced overshooting issues. It became known as PID pneumatic controller. PID controllers are not an entity that is only related to vehicles, they are used in various sectors of mechanical automation. PID controllers play a significant role in solving any control problem.

The You Only Look Once (YOLO) algorithm was presented in [17], the field exploded due to the algorithm giving much better results. To date, there have been 5 versions of YOLO, where YOLOv5 has the highest accuracy and speed, according to [10]. In YOLOv5, this selection process was integrated into the algorithm. The task of object detection is a vital one for autonomous vehicles. Detection of the traffic light, vehicles, and lanes must be done for the vehicle to run in a real environment. In [18], DeepTLR is proposed. It was the very first method to propose CNN for traffic light recognition. [19] describes a hierarchal DeepTLR for traffic light recognition. It is based on one end-to-end trainable CNN, that does not commit itself to a feature extraction network and uses tiling layers to augment final output granularity.

CARLA simulator is an open-world simulator concerning vehicle navigation, it has support for various Maps i.e., interaction environment and detailed control over that environment is also available [20].

Papers like [21] and [7], performed well on the KTTI dataset, by introducing better algorithms like YOLO and Vote3Deep respectively. Most of the papers reviewed, consisted of CNN/RNN Deep Neural Networks, for performing Object Detection Tasks. Papers like [5] and [22], on the other hand, used Feedback based system and Adaboost Cascaded Classifier, for performing Object Detection. Papers like [2, 21], and [10], also made use of 3-D Camera Projections, to create a better perception of the environment. Many papers did not make use of "Off Device Storage/Cloud" for storing models with the exceptions of Paper [23] and [24]. Barring Paper [4, 6], and [22], all other papers have considered Hard-ware Integration while proposing an approach. Except for Papers [1, 6, 25], and [24], all other papers propose an AI/Perception Approach.

3 Methodology

In this paper, a combined PID Controller [26] and Traffic Light Detection Model were implemented for navigating a vehicle in CARLA Simulator. For implementing, the below steps were followed:

1. Implementation of PID Controller.

2. Traffic Light Detection.
3. Combined Framework.

4 Implementation of PID Controller

PID Controller is used to steering a vehicle along a given path with specified velocities. Any PID Controller needs a list of waypoints to follow and the speed of the vehicle at that waypoint. For this paper, we used the autopilot function of CARLA Simulator to get the list of waypoints and the instantaneous velocities. CARLA’s autopilot feature navigates a vehicle in the simulator automatically. After setting the car in autopilot, a program was used to note down the coordinates of the path followed by the vehicle and its velocity at that instant in a CSV file. Table 1 shows a snippet of the generated CSV for a path in Town 02 of CARLA Simulator. As seen in the snippet, the first two columns depict the X and Y Coordinates of the vehicle and the third column depicts the speed of the vehicle.

This CSV file is then used by our PID Controller to help navigate the vehicle in the simulator. In our implementation, we are using two PID Controllers [27]:

1. Lateral PID Controller.
2. Longitudinal PID Controller.

Lateral PID Controller

The lateral PID Controller is used to control the vehicle’s steering angle to follow a given set of waypoints. A PID controller minimizes the error between the current value and the target value. In the case of steering angle, the error is calculated using the following formula:

$$err = \cos^{-1} \frac{v \cdot w}{|v||w|} \quad (1)$$

Table 1 Trajectory CSV snippets

X	Y	Speed
89.31	-46.71	26.58
89.45	-46.57	26.53
89.59	-46.43	26.61

Table 2 Model training result

Model	Training		Testing	
	Precision	Recall	Precision	Recall
YOLOv5	0.976	0.931	0.945	0.913

where v is the forward vehicle vector and w is the target vector

Using the error calculated, the value of steer at any given instant is given by the following formula:

$$steer = K_p * err + K_d * \frac{\Delta err}{\Delta t} + K_i * \sum err \quad (2)$$

where K_p , K_d and K_i are the proportional, derivate, and integral constants.

The value of steer is then clipped to make sure it lies in the interval $[-1,1]$. The vehicle is then navigated using the calculated steering value. In this way, the car is steered along the given path following the list of waypoints.

Longitudinal PID Controller

The longitudinal PID Controller is used to control the speed of the vehicle. This controller tries to minimize the error between the current vehicle velocity and the target velocity. Figure X, shows the formula for calculation of error in case of speed:

$$err = v - w \quad (3)$$

Using the error calculated, the value of throttle at that instant is calculated using:

$$throttle = K_p * err + K_d * \frac{\Delta err}{\Delta t} + K_i * \sum err \quad (4)$$

The calculated throttle is then clipped in the interval of $[-1,1]$. In the case of a negative throttle, a brake is applied with the same magnitude. Using this, the speed of the vehicle is controlled to match the list of target speeds.

Combined PID Controller

To navigate a vehicle, we need to control its steering angle and control its speed using the throttle. For this task, two independent controllers viz. a Lateral PID Controller and a Longitudinal PID Controller can be used to determine the vehicle's steering angle and throttle value respectively. In this paper, a combined Longitudinal and Lateral PID Controller were developed and it was made to follow a trajectory taken by the vehicle in autopilot mode in CARLA simulator. Figure 1 presents the steps followed to navigate a vehicle using the controllers in CARLA Simulator:

5 Traffic Light Detection

In this paper, Traffic Light Detection was implemented using a custom dataset made up of images of traffic lights in CARLA simulator. This dataset is then used to train a YOLOv5 Object Detection model [28] and is then deployed on Town-02 in

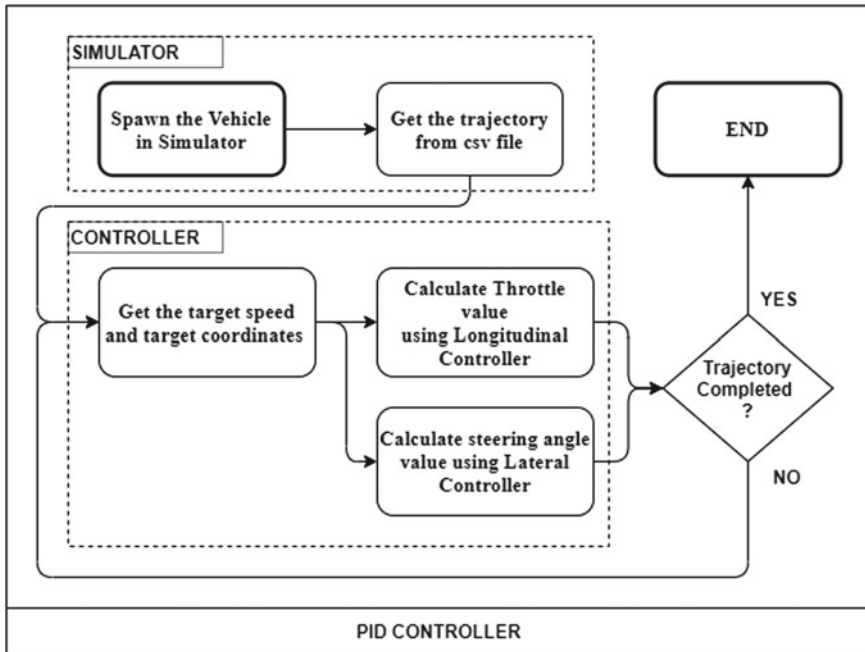


Fig. 1 Combined PID controller

CARLA Simulator. Figure 2, shows the steps followed for developing the Traffic Light Detection Model.

Capture images via manual control

In order to be able to detect traffic lights more accurately, images of traffic signals were captured directly from CARLA simulators. For this purpose, cameras were spawned on the dashboard of vehicles, and images were taken at regular intervals.

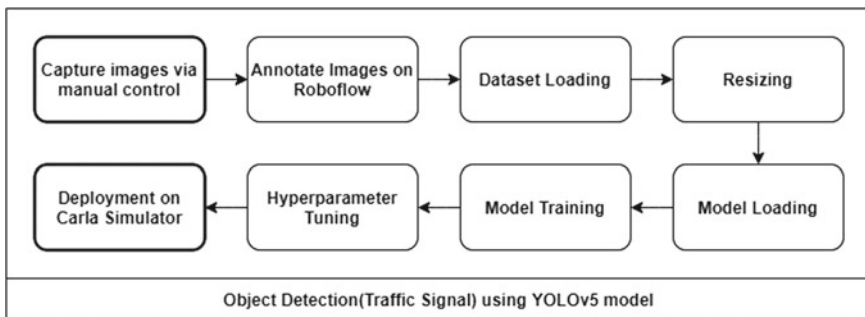


Fig. 2 Traffic light detection

The images captured were of RGB format and were uploaded to Roboflow [29] for annotation.

Annotate Images on Roboflow

After images were uploaded, the Roboflow API was used for manually annotating the images. The traffic lights were highlighted manually and the color of the traffic light was set as the class. For this paper, only red and green traffic lights are considered. Figure 3, shows the results of the annotations.

Dataset Loading and Images Resizing

The custom dataset consists of 146 images containing traffic lights of red and green color. Figure 4 shows the distribution of the dataset. It can be seen that out of 146 images, 67 images are of red color and 79 images are of green color. The dataset is then split into training and testing set of 116 and 30 images respectively. The images are then resized to 416×416 , which were then fed to the model.



Fig. 3 Annotation example

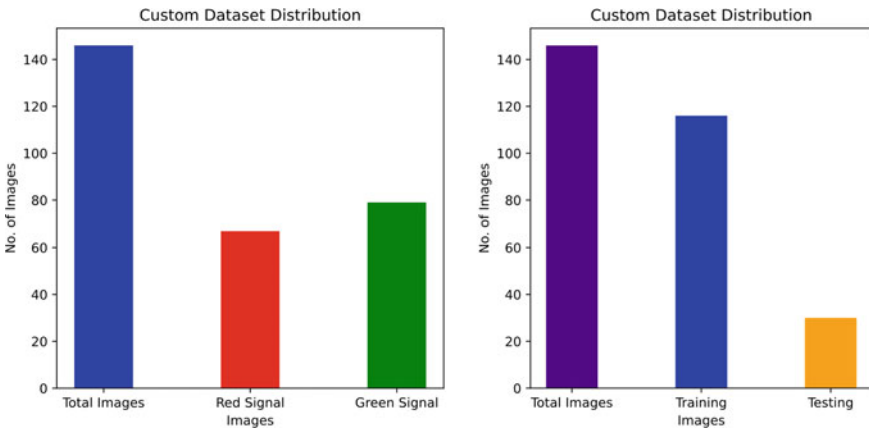


Fig. 4 Dataset distribution

Model Loading and Testing

For object detection, You Only Look Once (YOLOv5) model was used. Before starting the training, the model was loaded with pre-trained weights. The model was trained with a batch size of 16 for 100 epochs. For measuring the performance two metrics were used, viz: Precision and Recall.

Hyperparameter Tuning and Deployment

A callback function was implemented to use the best weights obtained in the last 10 epochs to increase the performance. After successful training, the model weights were saved. These weights were then used on a model deployed on CARLA simulator and its performance was evaluated.

6 Combined Framework

The PID controller and the traffic light detection model were combined in a single framework to navigate a vehicle along a given path while also responding to traffic lights in the path. Figure 5 shows the steps followed in the combined framework.

As seen in the flowchart, the vehicle is first spawned in the simulator. This is followed by reading the trajectory CSV file and also initializing the YOLOv5 model and the PID controller. A camera is also spawned on the dashboard of the vehicle. For identifying when to apply the brake, a boolean variable is initialized to false. The images from the dashboard camera are then regularly passed to the traffic light detection model. If the model detects a red light, the boolean is set to true and brakes are applied immediately. As soon as a green light or no light is detected in the picture, the boolean is set to false and the combined PID controller is used to navigate the car along the path. These steps are followed till the vehicle reaches the end of the trajectory.

7 Result and Discussion

7.1 PID Controller

Figure 6 shows the results of the combined PID controller. The blue graph represents the reference path and reference speed, whereas the orange graph represents the actual path followed by the vehicle and its speed. It can be observed that the PID controller was able to follow an almost identical path while also closely matching the target speed.

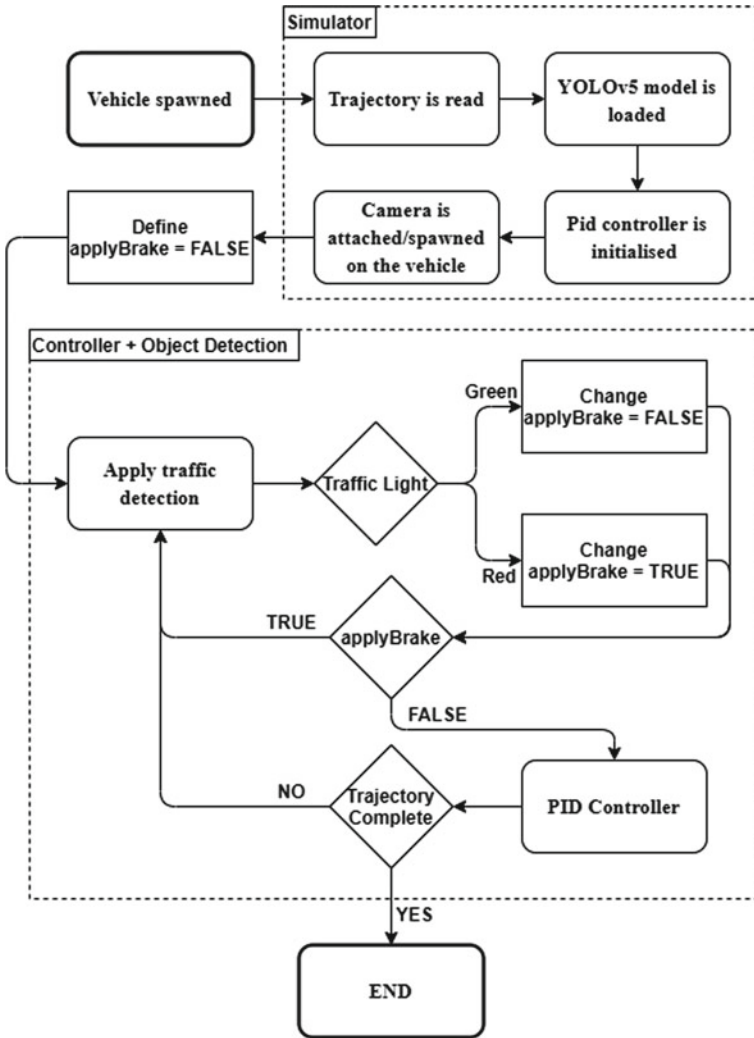


Fig. 5 Combined framework

8 Traffic Light Detection

The results of training of YOLOv5 model can be seen in Fig. 7. It can be observed that the training graph represented in blue color is very close to 1 for both precision and recall. The orange graph which represents the testing graph is initially increasing for both precision and recall. However, around 85–90 epochs, it started to plateau. Hence the model training was stopped after 100 epochs.

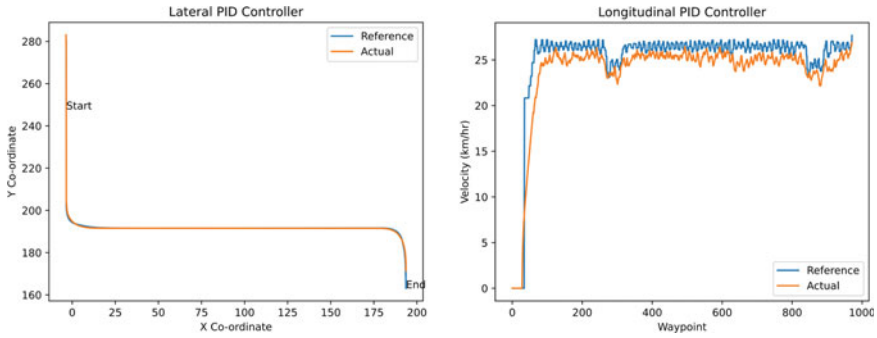


Fig. 6 Results of combined PID controller

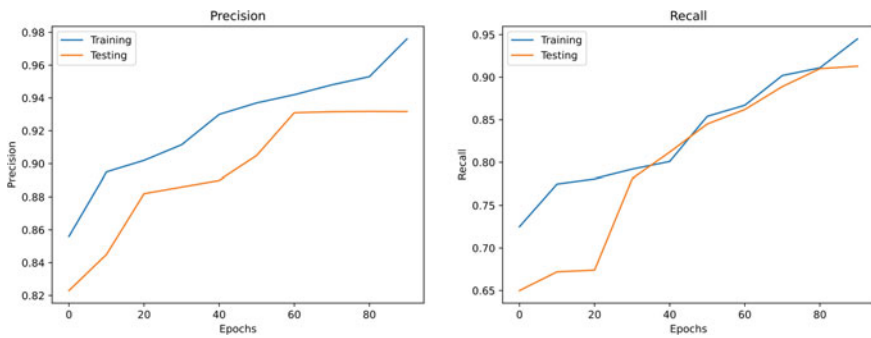


Fig. 7 YOLOv5 model training graph

9 Combined Framework

The combined framework was deployed in Town-02 in CARLA simulator with YOLOv5 and combined PID controller. Figure 8 shows the results of the combined framework. As seen in the graph, the vehicle was able to closely follow the list of waypoints. It can be observed that at the point a red light is detected, brakes are applied and the velocity of the vehicle is zero. As soon as the light turns green, it can be seen that the velocity of the vehicle starts to gradually increase to match the target velocity.

10 Conclusion

In this paper, a combined framework consisting of Longitudinal and Lateral PID controller to control speed and steering angle respectively, and a YOLOv5 model to detect traffic lights was deployed on CARLA simulator. The results from combined

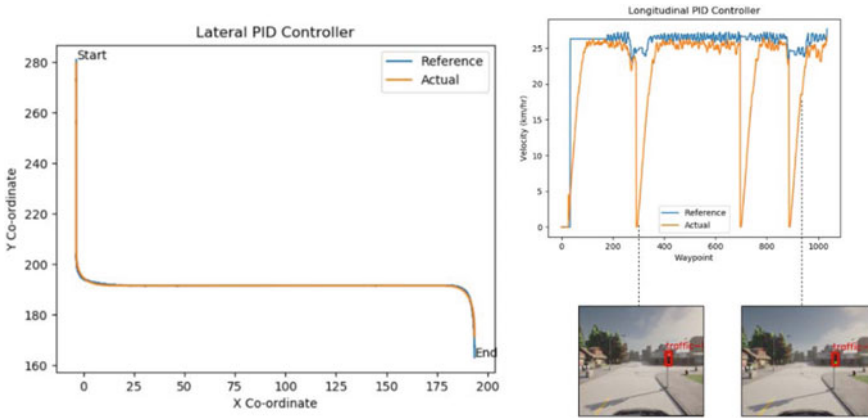


Fig. 8 PID controller + traffic light detection graph

Lateral and Longitudinal PID controller show that they can be used to navigate an autonomous vehicle in a simulator without traffic very closely to the actual trajectory. The traffic light detection model was trained with both precision and recall values reaching more 90%. This model was able to identify traffic lights of red and green color in the CARLA simulator with ease. The combine framework was able to safely maneuver around the map while stopping at the red lights. The vehicle was able to detect red lights from a distance and stopped immediately when it detected it to be red. Once a green light or no traffic light is detected, the vehicle was able to successfully continue following the path using PID controllers.

This paper uses a simplistic PID controller for controlling the vehicle. However, complex PID controllers can be used for improving the performance. Different algorithms can also be used for finding the optimal values of the constants for the PID controller. The traffic light detection model can be improved by using a more comprehensive dataset which can also include instances of other objects like cars, pedestrians, etc.

References

1. Anderson JM, Nidhi K, Stanley KD, Sorensen P, Samaras C, Oluwatola OA (2014) Autonomous vehicle technology: a guide for policymakers. Rand Corporation
2. Favarò FM, Nader N, Eurich SO, Tripp M, Varadaraju N (2017) Examining accident reports involving autonomous vehicles in California. PLoS ONE 12(9):e0184952
3. Millard-Ball A (2018) Pedestrians, autonomous vehicles, and cities. J Plan Educ Res 38(1):6–12
4. Palmeiro AR, van der Kint S, Vissers L, Farah H, de Winter JC, Hagenzieker M (2018) Interaction between pedestrians and automated vehicles: a Wizard of Oz experiment. Transport Res F: Traffic Psychol Behav 58:1005–1020
5. NHTSA Home Page. <https://www.nhtsa.gov/vehicle/2013>. Last accessed 06 Sept 2022

6. Thorpe C, Hebert MH, Kanade T, Shafer SA (1988) Vision and navigation for the Carnegie-Mellon Navlab. *IEEE Trans Pattern Anal Mach Intell* 10(3):362–373
7. Mahmoudi N, Ahadi SM, Rahmati M (2019) Multi-target tracking using CNN-based features: CNNMTT. *Multimedia Tools Appl* 78(6):7077–7096
8. Al-Qizwini M, Barjasteh I, Al-Qassab H, Radha H (2017) Deep learning algorithm for autonomous driving using googlenet. In: 2017 IEEE intelligent vehicles symposium (IV). IEEE, pp 89–96
9. Tao J, Wang H, Zhang X, Li X, Yang H (2017) An object detection system based on YOLO in traffic scene. In: 2017 6th international conference on computer science and network technology (ICCSNT). IEEE, pp 315–319
10. Thuan D (2021) Evolution of yolo algorithm and yolov5: the state-of-the-art object detection algorithm
11. Jo K, Kim J, Kim D, Jang C, Sunwoo M (2014) Development of autonomous car—part I: distributed system architecture and development process. *IEEE Trans Industr Electron* 61(12):7131–7140
12. Grigorescu S, Trasnea B, Cocias T, Macesanu G (2020) A survey of deep learning techniques for autonomous driving. *J Field Robot* 37(3):362–386
13. Jana A, Sarkar A, Kallakurchi JV, Kumar S (2019) Autonomous vehicle as a future mode of transport in India: analyzing the perception, opportunities and hurdles. In: Proceedings of the eastern Asia society for transportation studies, vol 12
14. Bechtel MG, McElhiney E, Kim M, Yun H (2018) Deeppicar: a low-cost deep neural network-based autonomous car. In: 2018 IEEE 24th international conference on embedded and real-time computing systems and applications (RTCSA). IEEE, pp 11–21
15. Pendleton SD, Andersen H, Du X, Shen X, Meghjani M, Eng YH, Rus D, Ang MH (2017) Perception, planning, control, and coordination for autonomous vehicles. *Machines* 5(1):6
16. Saeki M (2006) Fixed structure PID controller design for standard H_{∞} control problem. *Automatica* 42(1):93–100
17. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
18. Weber M, Wolf P, Zöllner JM (2016) DeepTLR: a single deep convolutional network for detection and classification of traffic lights. In: 2016 IEEE intelligent vehicles symposium (IV). IEEE, pp 342–348
19. Weber M, Huber M, Zöllner JM (2018) HDTLR: a CNN based hierarchical detector for traffic lights. In: 2018 21st international conference on intelligent transportation systems (ITSC). IEEE, pp 255–260
20. CARLA Homepage. <https://carla.org/>. Last accessed 06 Sept 2022
21. Li Y, Cui F, Xue X, Chan JCW (2018) Coarse-to-fine salient object detection based on deep convolutional neural networks. *Signal Process Image Commun* 64:21–32
22. Mayne DQ (2014) Model predictive control: recent developments and future promise. *Automatica* 50(12):2967–2986
23. Challenge DU (2007) Route network definition file (RNDF) and mission data file (MDF) formats. Tech. Rep; Defense Advanced Research Projects Agency: Arlington County, WV, USA
24. LaValle SM (2006) Planning algorithms. Cambridge University Press
25. Sivaraman S, Trivedi MM (2013) Looking at vehicles on the road: a survey of vision-based vehicle detection, tracking, and behavior analysis. *IEEE Trans Intell Transp Syst* 14(4):1773–1795
26. Zhao P, Chen J, Song Y, Tao X, Xu T, Mei T (2012) Design of a control system for an autonomous vehicle based on adaptive-pid. *Int J Adv Rob Syst* 9(2):44
27. Kusuma DH, Ali M, Sutantra N (2016) The comparison of optimization for active steering control on vehicle using PID controller based on artificial intelligence techniques. In: 2016 international seminar on application for technology of information and communication (ISEmantic). IEEE, pp 18–22

28. Wu TH, Wang TW, Liu YQ (2021) Real-time vehicle and distance detection based on improved yolo v5 network. In: 2021 3rd world symposium on artificial intelligence (WSAI). IEEE, pp 24–28
29. Lin Q, Ye G, Wang J, Liu H (2022) RoboFlow: a data-centric workflow management system for developing AI-enhanced Robots. In: Conference on robot learning. PMLR, pp 1789–1794

Binary Classification for High Dimensional Data Using Supervised Non-parametric Ensemble Method



Nandan Kanvinde, Abhishek Gupta, Raunak Joshi, and Pinky Gerela

Abstract High dimensional data for classification does create many difficulties for machine learning algorithms. The generalization can be done using ensemble learning methods such as bagging based supervised nonparametric random forest algorithm. In this paper we solve the problem of binary classification for high dimensional data using random forest for polycystic ovary syndrome dataset. We have performed the implementation and provided a detailed visualization of the data for general inference. The training accuracy that we have achieved is 95.6% and validation accuracy over 91.74% respectively.

Keywords Bagging · Ensemble Methods · Random Forest

1 Introduction

Machine Learning [1] technique performs predictive analysis and works with different dimensions of data. The high dimensional data is problematic for some of the basic machine learning algorithms. Data can be particularly intended for classification [2] or regression [3] tasks, but high dimensions in the data are independent of the factor. In this paper we try to consider the classification task, especially binary classification [4] task. Considering the binary classification for high dimensional data we require learning procedure that is not quite basic. Considering a basic linear learning classification algorithm such as Logistic Regression [5], the binary classification is better because the algorithm is intended for it yet after training over high dimensional data with multiple categorical variable, the performance might degrade to a greater extent. The improved algorithms like K-Nearest Neighbors [6], Support Vector Machines [7] and CART [8] are good in context as compared to

N. Kanvinde · P. Gerela
Thakur Institute Of Management Studies, Career Development & Research, Mumbai 400101,
India

A. Gupta (✉) · R. Joshi
University of Mumbai, Mumbai 400032, India
e-mail: abhishek.gupta20001@gmail.com

Logistic Regression yet have some or the limitation considering the high dimensional data. This is where use of ensemble learning [9] procedures can prove to be a better solution which is practically an accumulation of weak-set of learners that yield a good result. The main divisions in the area of ensemble learning are bagging [10] and boosting [11]. This paper covers the bagging process in detail and boosting is out of context. Now the main task that further requires attention is the data used. Considering the points pertaining to our problem statement, binary classification dataset which high dimensional with categorical variables which will be handled using feature engineering techniques. We contemplated implementing the "Polycystic Ovary Syndrome"[12] diagnosis dataset. The primary task of the paper is proving that bagging ensemble learning method can prove to be much efficient with high dimensional data and provides a more generalized result as compared to some of the traditional learning procedures.

2 Implementation

Bootstrap Aggregation [13] is other terminology for bagging [10] ensemble learning methods. Random Forest [14] is the most commonly used bootstrap aggregation. The process of bootstrap aggregation states segregating the data into pieces with set of rules and later performing an aggregation. The random forest is a supervised non-parametric learning system. Hence it is ensemble learning, it considers the accumulation of weak learners with the bootstrap aggregation system. The basic structure of the random forest uses a set of decision trees [15] which are considered as Estimators in the random forest implementation. The estimators when used in large numbers increase the time complexity of the model for learning process. The decision trees by themselves are inefficient as they go into high variance problem when trained on high dimensions. The random forest tackles that problem effectively. The number of decision trees used are influenced by the depth parameter, which indicates the depth of tree starting from the first node. The depth of the random forest can be considered as logarithmic value of the number of estimators. The estimators in random forest for the last layer are considered as 2 times the number of estimators. The random forest is primarily used for generalization [16] of error using out of bag [17] score as a parameter. The samples which are not trained nor tested are used by the out of bag score for checking the efficiency of data. Considering the parameters used in the implementation using scikit-learn [18, 19] for random forest, the maximum depth of the forest used is 8, estimators which are also known as decision trees are 100, minimal sample split is 23 and minimum sample leaves are 2 respectively.

3 Results

3.1 Analysis of Data

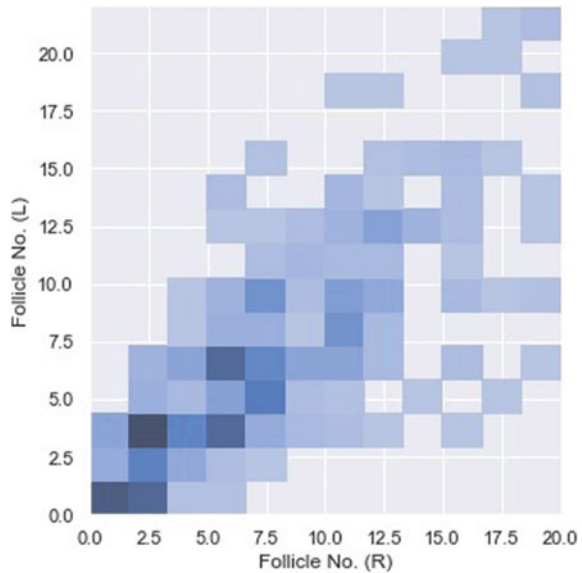
Many implementations of the PCOS is done using boosting methods [20], discriminant analysis [21], stacked generalizations [22] and deep learning [23], but analysis of the data is done most precisely in this paper. Analysis of the data in varied processes that can derive the inferences is important. Visualizations[24] are necessary because they can point out very subtle points in the dataset. The Fig. 1 visualizes the follicles with its correlation where the correlation is considered from left to right where darker the color, higher the correlation. The follicles that are affected by PCOS are given in Fig. 2 where the affected PCOS values can be depicted as orange in the figure whereas blue for not affected.

The physical activity has influence over the affected PCOS. The Fig. 3 depicts affects of PCOS when examined with physical activity and junk food consumption. The section where the utilization of junk food is good and regular physical activity is negative, PCOS does affect the highest.

The Fig. 4 is depiction of a Bi-variate KDE [25] Graph. It provides one the continuous PDF curve in specific dimensions for distributions. It is comparatively simple for interpreting than comparing along line scatters. This could be further applied for giving graphs with respect to the classes.

The Fig. 5 provides the Bi-variate KDE graph for follicles in accordance to the classes. The understandability is perceivable as one can certainly infer the affected range of PCOS under the distribution.

Fig. 1 Follicles correlation



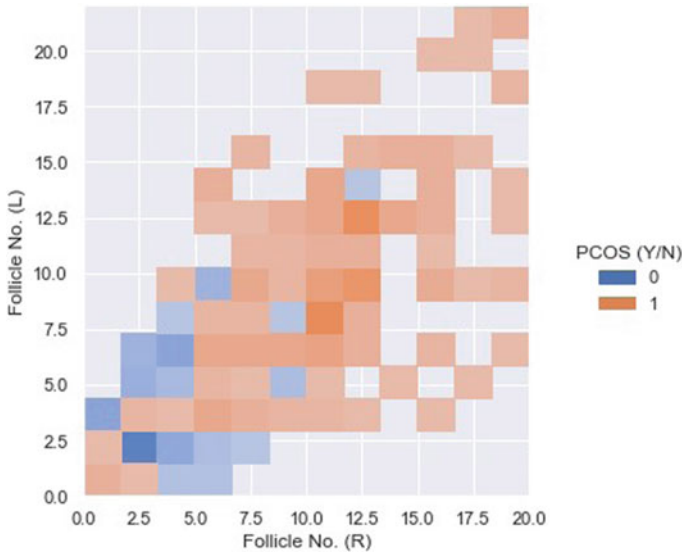


Fig. 2 Follicles affected correlation

3.2 Precision and Recall

The Precision and Recall [26] are basic metrics for all classification problems. Confusion Matrix [27] focuses on the rudimentary elements like true positives, false positives, true negatives and false negatives for p and r abbreviations for precision and recall (Table 1).

Precision and Recall have 2 different types known as macro and weighted average. The macro considers all the individual classes into consideration with the unweighted average (Table 2).

3.3 F-Score

F-Score [28] is the precision and recall harmonic average. Even F-Score has macro and weight averages. The Table 3 provides precise depiction of the F-score.

Same as different classes observations were available in Table 2, depiction for separate classes are present for F-score. The Table 4 does provide precise F-score with respect to all the single classes. Separate classes depict the results observations on individual levels. Combination does make a difference in long run.

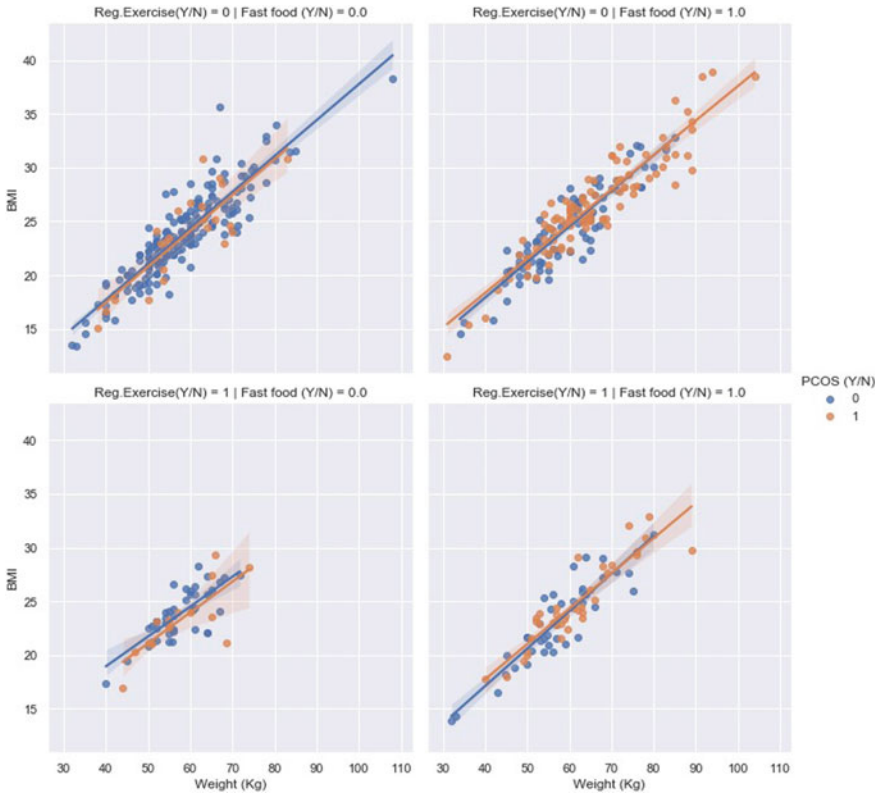


Fig. 3 PCOS exercise effects

3.4 Receiver Operating Characteristic

The Receiver Operating Characteristic [29] abbreviated as RoC Curve is graphical depiction for binary classification only that provides the algorithm, quality for surpassing specific threshold values. The area that falls above the threshold is considered as Area Under Curve [30]. Values closer to 100% are considered to be good and closer to 0% are considered weakly performing. The basic elements that form the RoC Curve are true positive rate [31] and false positive rate which are basically the sensitivity and specificity. The Fig. 6 shows the random forest performed with RoC curve. The solid scarlet color line that bypasses through middle is threshold which once crossed indicates a good metric for consideration of RoC. The random forest in this case performs good and gives the score 98% area under curve.

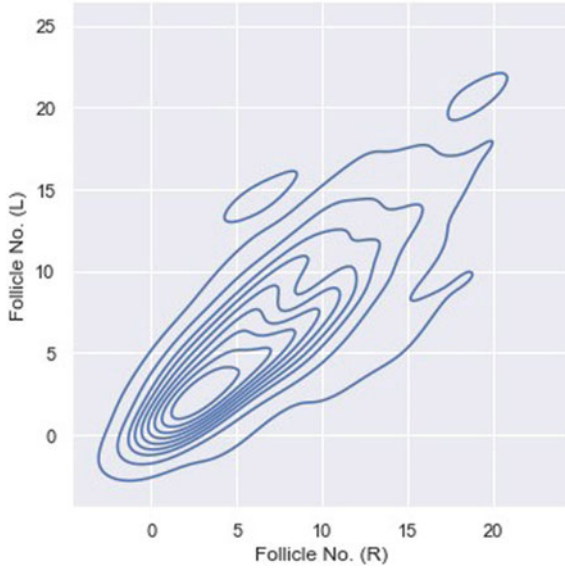


Fig. 4 Follicles KDE

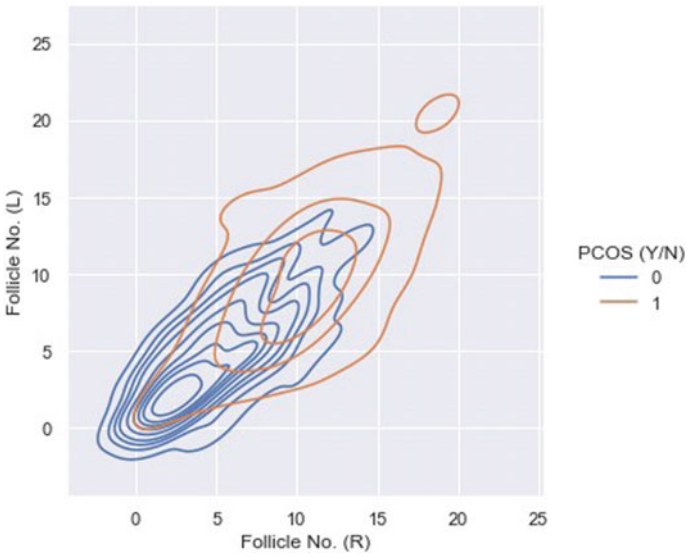


Fig. 5 Labeled Follicle KDE

Table 1 Macro and weighted averages for P and R

Types	Precision	Recall
Macro	0.93	0.87
Weighted	0.91	0.91

Table 2 Individual labels for precision and recall

Types	Precision	Recall
Class 0	0.89	0.99
Class 1	0.96	0.75

Table 3 F-score for macro and weight average

Metric	Macro	Weight
F-score	0.89	0.90

Table 4 Individual labels F-score

Metric	Class 0	Class 1
F-score	0.94	0.84

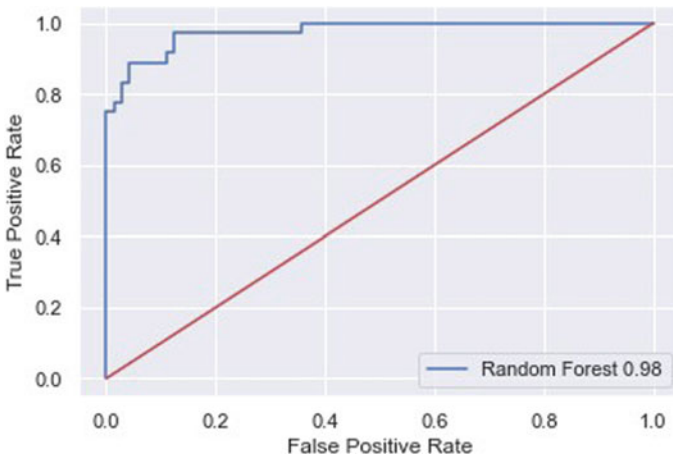


Fig. 6 RoC curve and AuC

4 Conclusion

If we consider the binary classification problem with high dimensional data the use of bagging ensemble method proves to be effective at generalizing the model and also covering the drawbacks of traditional algorithms. This paper covers a perfect implementation of the random forest for polycystic ovary syndrome dataset. The paper gives a detailed visualization based analysis of the data along with varied metrics that cover the reach of the algorithm. This paper will definitely enlighten many researchers on subtle topics and their implementations which we are proud to be a part of it.

References

1. Ray S (2019) A quick review of machine learning algorithms. In: 2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon). pp 35–39. <https://doi.org/10.1109/COMITCon.2019.8862451>
2. Cormack RM (1971) A review of classification. *J Roy Stat Soc Ser A (General)* 134(3):321–367. <http://www.jstor.org/stable/2344237>
3. Maulud D, Abdulazeez AM (2020) A review on linear regression comprehensive in machine learning. *J Appl Sci Technol Trends* 1(4):140–147. <https://doi.org/10.38094/jastt1457>. <https://jastt.org/index.php/jasttpath/article/view/57>
4. Kumari R, Srivastava SK (2017) Machine learning: a review on binary classification. *Int J Comput Appl* 160:11–15
5. Cramer JS (2003) The origins of logistic regression. SSRN Electron J
6. Guo G, Wang H, Bell DA, Bi Y, Greer KRC (2003) Knn model-based approach in classification. OTM
7. Hearst M, Dumais S, Osuna E, Platt J, Scholkopf B (1998) Support vector machines. *IEEE Intell Syst Appl* 13(4):18–28. <https://doi.org/10.1109/5254.708428>
8. Bittencourt H, Clarke R (2003) Use of classification and regression trees (cart) to classify remotely-sensed digital images. In: IGARSS 2003. 2003 IEEE international geoscience and remote sensing symposium. Proceedings (IEEE Cat. No.03CH37477), vol 6, pp 3751–3753. <https://doi.org/10.1109/IGARSS.2003.1295258>
9. Opitz D, Maclin R (1999) Popular ensemble methods: an empirical study. *J Artif Intell Res* 11:169–198. <https://doi.org/10.1613/jair.614>
10. Breiman L (2004) Bagging predictors. *Mach Learn* 24:123–140
11. Freund Y, Schapire RE (1999) A short introduction to boosting
12. Allahbadia GN, Merchant R (2011) Polycystic ovary syndrome and impact on health. *Middle East Fertility Soc J* 16(1):19–37
13. Lee TH, Ullah A, Wang R (2019) Bootstrap aggregating and random forest
14. Breiman L (2004) Random forests. *Mach Learn* 45:5–32
15. Quinlan JR (2004) Induction of decision trees. *Mach Learn* 1:81–106
16. Athey S, Tibshirani J, Wager S (2019) Generalized random forests. *Ann Stat* 47(2):1148–1178
17. Janitza S, Hornung R (2018) On the overestimation of random forest’s out-of-bag error. *Plos One* 13(8):1–31. <https://doi.org/10.1371/journal.pone.0201904>
18. 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 (2011) Scikit-learn: machine learning in python. *J Mach Learn Res* 12:2825–2830
19. Buitinck L, Louppe G, Blondel M, Pedregosa F, Mueller A, Grisel O, Niculae V, Prettenhofer P, Gramfort A, Grobler J, Layton R, VanderPlas J, Joly A, Holt B, Varoquaux G (2013) API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD workshop: languages for data mining and machine learning, pp 108–122
20. Gupta AM, Shetty SS, Joshi RM, Laban RM (2021) Succinct differentiation of disparate boosting ensemble learning methods for prognostication of polycystic ovary syndrome diagnosis. In: 2021 international conference on advances in computing, communication, and control (ICAC3). IEEE, pp 1–5
21. Gupta A, Soni H, Joshi R, Laban RM (2022) Discriminant analysis in contrasting dimensions for polycystic ovary syndrome prognostication. arXiv preprint [arXiv:2201.03029](https://arxiv.org/abs/2201.03029)
22. Nair S, Gupta A, Joshi R, Chitre V (2022) Combining varied learners for binary classification using stacked generalization. arXiv preprint [arXiv:2202.08910](https://arxiv.org/abs/2202.08910)
23. Gupta A, Nair S, Joshi R, Chitre V (2022) Residual-concatenate neural network with deep regularization layers for binary classification. arXiv preprint [arXiv:2205.12775](https://arxiv.org/abs/2205.12775)
24. Sadiku M, Shadare AE, Musa SM, Akujuobi CM. Data visualization
25. Chen YC (2017) A tutorial on kernel density estimation and recent advances
26. Powers DMW (2020) Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation

27. Ting KM (2017) Confusion matrix. Springer US, Boston, MA, pp 260–260
28. Sokolova M, Japkowicz N, Szpakowicz S (2006) Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In: Australian conference on artificial intelligence
29. Fawcett T (2006) An introduction to roc analysis. *Pattern Recogn Lett* 27:861–874
30. Bradley AP (1997) The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recogn* 30:1145–1159
31. Wang H, Zheng H (2013) True positive rate. Springer New York, New York, NY, pp 2302–2303

Deep Linear Discriminant Analysis with Variation for Polycystic Ovary Syndrome Classification



Raunak Joshi, Abhishek Gupta, Himanshu Soni, and Ronald Laban

Abstract The polycystic ovary syndrome diagnosis is a problem that can be leveraged using prognostication based learning procedures. Many implementations of PCOS can be seen with Machine Learning but the algorithms have certain limitations in utilizing the processing power graphical processing units. The simple machine learning algorithms can be improved with advanced frameworks using Deep Learning. The Linear Discriminant Analysis is a linear dimensionality reduction algorithm for classification that can be boosted in terms of performance using deep learning with Deep LDA, a transformed version of the traditional LDA. In this result oriented paper we present the Deep LDA implementation with a variation for prognostication of PCOS.

Keywords Deep Learning · Deep LDA · Linear Discriminant Analysis

1 Introduction

The use of medical research data for various statistical tasks has been done from a prolonged period of time. The data after having consistent number of records can be utilized for deriving inference using prognostication methods. The methods that fall under the area of inferential statistics [1] which are extended with applied areas of statistics can be used, one of which can be used is Machine Learning [2]. Task of prognostication based on data can be done by learning patterns from the data using machine learning. The dataset that we have used in this paper is pertaining to polycystic ovary syndrome [3] diagnosis, which falls under the classification [4] category. Classification has 2 sub-divisions, viz. Binary and Multi-Class where the data used in this paper falls under binary classification [5] precisely. The methods using Machine Learning have already been performed on polycystic ovary syndrome

R. Joshi · A. Gupta (✉)
University of Mumbai, Mumbai 400032, India
e-mail: abhishek.gupta20001@gmail.com

H. Soni · R. Laban
St. John College of Engineering and Management, Palghar 401404, India

abbreviated as PCOS using logistic regression [6], bagging ensemble methods [7], discriminant analysis [8], stacked generalization [9], boosting ensemble methods [10] and deep neural networks [11]. The use of deep learning is evident and we want to focus on the more variations that can be brought into the current state-of-the-art system. Deep Learning [12, 13] provides more depth of learning as compared to machine learning and is used when the amount of parameters in dimensions are high. The PCOS dataset has over 41 dimensions which are enough for instating the use of deep learning. The variation we wanted to perform was related to some machine learning algorithm that can be leveraged with power of deep learning. The implementation of any machine learning algorithm using a library like scikit-learn [14, 15] meets limitations in terms of utilization with GPU processing power. For the same reason, using a mature framework like Tensorflow [16] can definitely bring change to the working. This is where we decided to work with parametric learning method which works with simple and definite procedures. The parametric learning method we focused on using was discriminant analysis [17, 18] which has variations in it where we focused on Linear Discriminant Analysis [19]. This actually accounted for an idea that training the linear discriminant analysis with deep learning style will yield us Deep Linear Discriminant Analysis [20, 21] which has been already been discovered and we decided to proceed with our implementation using it. The variations that we brought in the network will be explained in further sections of this paper.

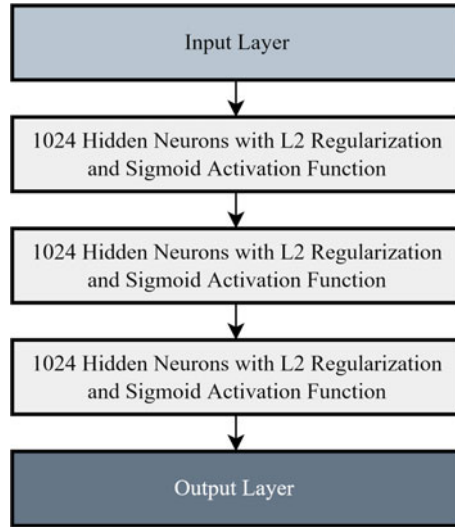
2 Methodology

This section of the paper gives detailed insights about the implementation and approach we have taken to solve the problem. The model considering the implementation with respect to Deep LDA [20, 21] revolves around the idea of the convolutional neural networks [22–25]. The modification can be done to work with numerical values. This has been implemented by various developers and names of the developers are given in the acknowledgement section of the paper. The Deep LDA is basically an implementation of latent representations in linearly separable method. The Deep LDA is an extensive implementation of the traditional Linear Discriminant Analysis which was intended for dimensionality reduction based classification methods. The implementation consists of 2 phases, first phase consists of linear discriminator as the deep neural network and second phase consists of support vector machine for detailed classification.

2.1 First Phase

The Fig. 1 gives depiction of first phase of the implementation. The input layer takes 41 dimension of features from the data. This is passed on to one dense layer that has

Fig. 1 First phase with LDA implementation

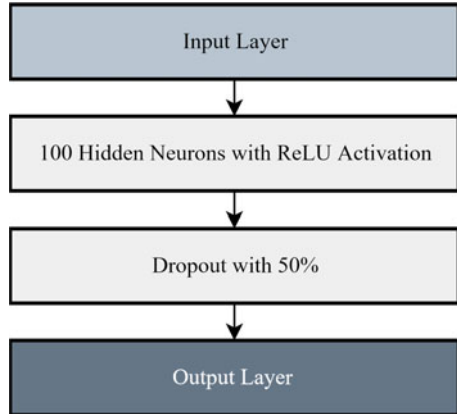


1024 hidden neurons. The L2 regularization [26] is applied as a kernel regularization for the layer. The activation function is used sigmoid [27]. The rectified linear unit abbreviated as ReLU [28] is the activation function commonly used for deep learning methods but using sigmoid ensures linear based system such as developed for linear discriminant analysis. The parameters learned from the first hidden layer are 43,008. Similar type of hidden layers are repeated twice, where second hidden layer learns 1,049,600 parameters and third hidden layer also learns same amount of parameters. The output layer consists of 1 hidden neuron with sigmoid activation function and learns 1025 parameters. The network learns total of 2,143,233 parameters where all the parameters are trainable. The loss function used is binary cross-entropy that differs from the original implementation of deep LDA paper. The loss optimizer used is Adam [29] optimizer with learning rate of $1 * 10^{-5}$ that roughly denotes 0.00001. The implementation is done using Keras [30] over Tensorflow [16] back-end trained for 100 epochs with 64 as batch size.

2.2 Second Phase

The Fig. 2 depicts the second phase implementation. This is done using Support Vector Machine [31] implementation with neural network inclination. The input layer is connected to hidden layer with 100 hidden neurons with ReLU [28] activation function. This layer learns 200 parameters. This layer is connected to dropout [32] layer with 50% threshold and does not learn any parameters. The output layer has 1 hidden neuron and has sigmoid activation function for binary classification and learns 101 parameters. The total parameters learned are 301 and the network uses

Fig. 2 Second phase with SVM implementation



binary cross-entropy. The loss optimizer used is Adam [29] optimizer with $1 * 10^{-5}$ that approximately denotes 0.00001 learning rate. The network is trained with 100 epochs and 64 as batch size. 2.3 Complete Network (Fig. 3).

The complete network is accumulation of first and second phase where the output of the first phase is the input for second phase. The output of the first phase is 1-dimensional array from 41-dimensional output. This is given as an input to the second phase of the network and final prediction which is 1-dimensional is achieved. The both phases are trained independently and the output is retained from first phase and given as second phase. The results of the network will be given in succeeding section of the paper.

3 Results

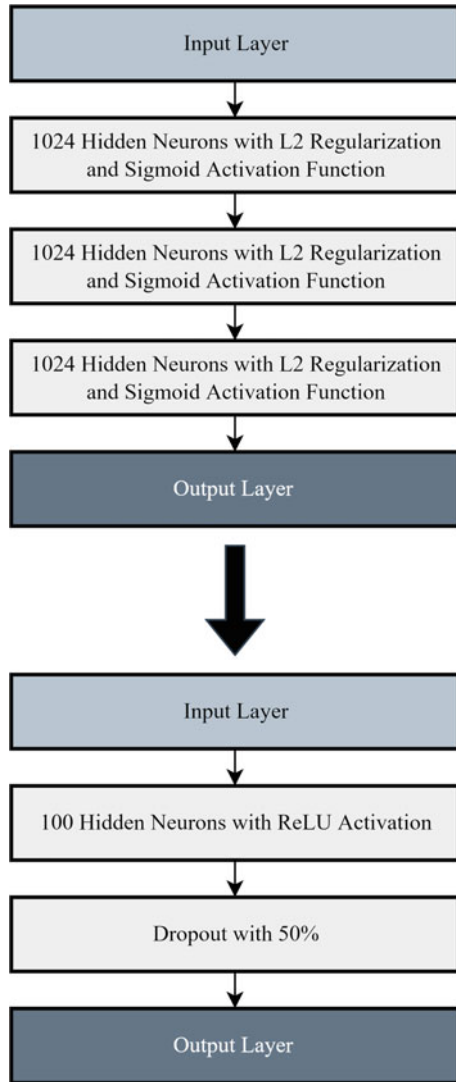
3.1 Accuracy and Loss for First Phase

The training and validation accuracy graph can be seen from Fig. 4 and the infractions between the training and validation accuracy seem a bit wider but the values for training are 98.35% and validation 90.909% respectively. Considering the loss, the validation loss has some infraction that can be seen with different metrics the training loss is 6.79% whereas the validation loss is 38.05% respectively.

3.2 Accuracy and Loss for Second Phase

The second phase contains support vector machine and not necessarily the graph depiction is right measure but we have included the graph. The inference can be

Fig. 3 Complete network



drawn as there are no significant changes in the accuracy or loss. All the metrics are learnt from the trained parameters of the first phase and it does not make much sense to make mappings out of it. The training accuracy for the support vector machine phase is generated as 98.354% and validation accuracy is obtained as 90.909% which is similar to the first phase training and validation accuracy. The training loss is 6.79% and validation loss is 38.052% which is again similar to the first phase. The better inference can be generated from different metrics intended for classification and not just the graph depictions of the training and validation accuracy as well as loss (Fig. 5).

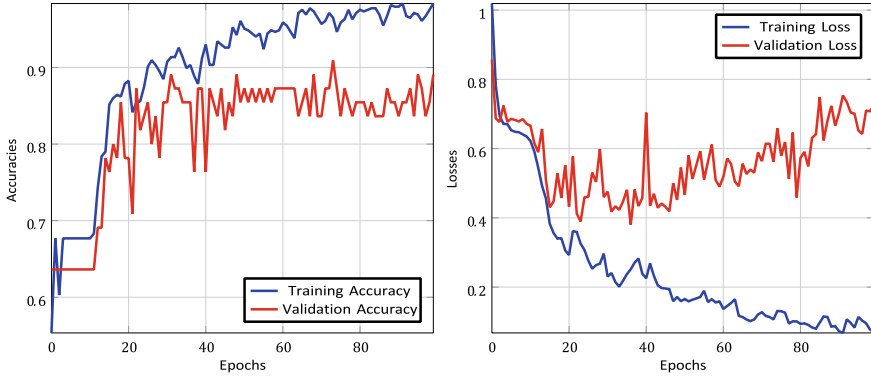


Fig. 4 Accuracy and loss for the first phase

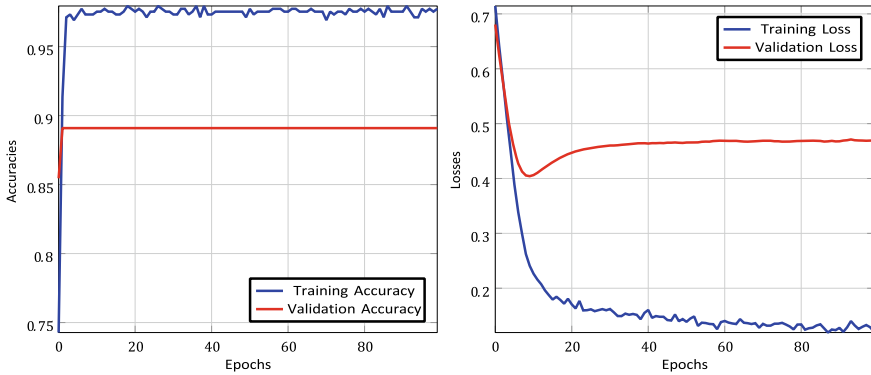


Fig. 5 Accuracy and loss for the second phase

3.3 Precision

The precision [33] is dependent on total number of samples that are predicted to be positive among all the set of samples. This is a popular metric for prognostication algorithms and requires basic elements of a confusion matrix [34], viz. true positives, true negatives, false positives and false negatives. The precision score obtained in 88.88% which is very close to 1 as expected.

3.4 Recall

The recall [33] is a metrics which states all the positive elements from the every single predicted element. The recall also utilizes every single element from the confusion

matrix same as precision. The recall generated for the model is 80% which is again a very good score and gives inference that model has performed adequately. The recall is not only metric that gives the final inference and more precise metric can be obtained.

3.5 *F-Score*

This is a subtle metric that gives the overall flow of how efficiently does the model perform. The building blocks of F-score [33, 35] are precision and recall. The F-Score we got for the model is 84.21%, which is adequately good and proves that model performs better on average.

4 Conclusion

The idea of entire paper revolves around an experimentation that can be performed for polycystic ovary syndrome diagnosis problem. We proved a point that simple machine learning algorithms can be leveraged using deep learning for efficient performance based inclination. The Deep Linear Discriminant Analysis idea was proven in this paper. We also introduced some of our personal variations in implementation and they turned out to be effective from the results section of the paper. The paper can definitely sum up many lost ideas into practical implementation and enforce many young researchers for better development perspective.

Acknowledgements We would genuinely like to thank Mr. Prasoon Kottarathil for making the polycystic ovary syndrome dataset available through Kaggle platform. We would also like to deeply thank for the contributions of VahidooX - <https://github.com/VahidooX/DeepLDA> and Thomas Chaton <https://github.com/tchaton/DeepLDA> for providing us the basic implementations of Deep LDA based from the original paper.

References

1. Marshall G, Jonker L (2011) An introduction to inferential statistics: a review and practical guide. Radiography 17
2. Sarker IH (2021) Machine learning: algorithms, real-world applications and research directions. SN Comput Sci 2(3):1–21
3. Ndefo UA, Eaton A, Green MR (2013) Polycystic ovary syndrome: a review of treatment options with a focus on pharmacological approaches. Pharmacy Therap 38(6):336
4. Cormack RM (1971) A review of classification. J Roy Stat Soc Ser A (Gen) 134(3):321–367. <http://www.jstor.org/stable/2344237>

5. Kumari R, Srivastava SK (2017) Machine learning: a review on binary classification. *Int J Comput Appl* 160(7):11–15. <https://doi.org/10.5120/ijca2017913083>, <http://www.ijcaonline.org/archives/volume160/number7/27084-2017913083>
6. Chauhan P, Patil P, Rane N, Raundale P, Kanakia H (2021) Comparative analysis of machine learning algorithms for prediction of pcos. In: 2021 international conference on communication information and computing technology (ICCICT), pp 1–7. <https://doi.org/10.1109/ICCICT50803.2021.9510128>
7. Kanvinde N, Gupta A, Joshi R (2022) Binary classification for high dimensional data using supervised non-parametric ensemble method. arXiv preprint [arXiv:2202.07779](https://arxiv.org/abs/2202.07779)
8. Gupta A, Soni H, Joshi R, Laban RM (2022) Discriminant analysis in contrasting dimensions for polycystic ovary syndrome prognostication. arXiv preprint [arXiv:2201.03029](https://arxiv.org/abs/2201.03029)
9. Nair S, Gupta A, Joshi R, Chitre V (2022) Combining varied learners for binary classification using stacked generalization. arXiv preprint [arXiv:2202.08910](https://arxiv.org/abs/2202.08910)
10. Gupta AM, Shetty SS, Joshi RM, Laban RM (2021) Succinct differentiation of disparate boosting ensemble learning methods for prognostication of polycystic ovary syndrome diagnosis. In: 2021 international conference on advances in computing, communication, and control (ICAC3). IEEE, pp 1–5
11. Gupta A, Nair S, Joshi R, Chitre V (2022) Residual-concatenate neural network with deep regularization layers for binary classification. arXiv preprint [arXiv:2205.12775](https://arxiv.org/abs/2205.12775)
12. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444
13. Goodfellow I, Bengio Y, Courville A (2016) Deep learning
14. 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 (2011) Scikit-learn: machine learning in Python. *J Mach Learn Res* 12:2825–2830
15. Buitinck L, Louppe G, Blondel M, Pedregosa F, Mueller A, Grisel O, Niculae V, Prettenhofer P, Gramfort A, Grobler J, Layton R, VanderPlas J, Joly A, Holt B, Varoquaux G (2013) API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD workshop: languages for data mining and machine learning, pp 108–122
16. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X (2015) TensorFlow: large-scale machine learning on heterogeneous systems. <https://www.tensorflow.org/>. Software available from tensorflow.org
17. King LJ (1970) Discriminant analysis: a review of recent theoretical contributions and applications. *Econ Geogr* 46:367–378. <http://www.jstor.org/stable/143150>
18. Das Gupta S (1980) Discriminant analysis. In: Fienberg SE, Hinkley DV, Fisher RA (eds) *An appreciation*. Springer New York, New York, NY, pp 161–170
19. Tharwat A, Gaber T, Ibrahim A, Hassanien AE (2017) Linear discriminant analysis: a detailed tutorial. *AI Commun* 30:169–190
20. Dorfer M, Kelz R, Widmer G (2015) Deep linear discriminant analysis. arXiv preprint [arXiv:1511.04707](https://arxiv.org/abs/1511.04707)
21. Tian Q, Arbel T, Clark JJ (2017) Deep lda-pruned nets for efficient facial gender classification. In: 2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW), pp 512–521. <https://doi.org/10.1109/CVPRW.2017.78>
22. LeCun Y, Bengio Y et al. Convolutional networks for images, speech, and time series
23. Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, Santamaría J, Fadhel MA, Al-Amidie M, Farhan L (2021) Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. *J Big Data* 8(1):1–74
24. Gupta A, Joshi R, Laban R (2022) Detection of tool based edited images from error level analysis and convolutional neural network. arXiv preprint [arXiv:2204.09075](https://arxiv.org/abs/2204.09075)
25. Joshi RM, Shah D (2022) Refactoring faces under bounding box using instance segmentation algorithms in deep learning for replacement of editing tools. In: *Intelligent computing and networking*. Springer, pp 236–247

26. Cortes C, Mohri M, Rostamizadeh A (2012) L2 regularization for learning kernels. arXiv preprint [arXiv:1205.2653](https://arxiv.org/abs/1205.2653)
27. Minai AA, Williams RD (1993) Original contribution: on the derivatives of the sigmoid. *Neural Netw* 6(6):845–853. [https://doi.org/10.1016/S0893-6080\(05\)80129-7](https://doi.org/10.1016/S0893-6080(05)80129-7)
28. Agarap AF (2018) Deep learning using rectified linear units (relu). arXiv preprint [arXiv:1803.08375](https://arxiv.org/abs/1803.08375)
29. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
30. Chollet F et al (2015) Keras. <https://github.com/fchollet/keras>
31. Hearst M, Dumais S, Osuna E, Platt J, Scholkopf B (1998) Support vector machines. *IEEE Intell Syst Their Appl* 13(4):18–28. <https://doi.org/10.1109/5254.708428>
32. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15(56):1929–1958. <http://jmlr.org/papers/v15/srivastava14a.html>
33. Powers DM (2020) Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. Xiv preprint [arXiv:2010.16061](https://arxiv.org/abs/2010.16061)
34. Ting KM (2017) Confusion matrix. *Encyclop Mach Learn Data Mining* 260
35. Goutte C, Gaussier E (2005) A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In: *European conference on information retrieval*. Springer, pp 345–359

Improved Helmet Detection Model Using YOLOv5



Premanand Ghadekar, Shreyas Mendhekar, Vallabh Niturkar,
Sanika Salunke, Abhinav Shambharkar, and Kshitij Taley

Abstract This report is about detecting motorbike riders without a helmet and also the pillion rider with the use of YOLO object detection algorithm. We introduced the updated approach for helmet detection. This approach is an upgradation of YOLO object detection algorithm which detects not only the rider's helmet but also the helmet of the pillion rider. Primary objective is Detection of helmet of rider and pillion rider in that targeted image and Increase the accuracy of the YOLOv5 algorithm by adding one layer for detection of small details in an image. In this proposed model a new layer has been added for detection of smaller objects having smaller features. This has been done by changing the configuration of YOLOv5 architecture. The helmet detection using this proposed model has been carried out for a dataset containing images with maximum 3 people, with no helmets, 1 helmet each or 2 wearing it.

Keywords YOLO · Helmet · YOLOv5 · Darknet · PyTorch · CSP · PA-Net · YAML · Head · Backbone · Feature maps · Anchor box · Bounding box · GPU · Custom data

P. Ghadekar (✉) · S. Mendhekar · V. Niturkar · S. Salunke · A. Shambharkar · K. Taley
Department of Information Technology, Vishwakarma Institute of Technology, Pune, India
e-mail: premanand.ghadekar@vit.edu

S. Mendhekar
e-mail: shreyas.mendhekar19@vit.edu

V. Niturkar
e-mail: vallabh.niturkar19@vit.edu

S. Salunke
e-mail: sanika.salunke19@vit.edu

A. Shambharkar
e-mail: abhinav.shambharkar19@vit.edu

K. Taley
e-mail: kshitij.taley19@vit.edu

1 Introduction

In India, two-wheeler is one of the mostly used mode of transport for shorter or some-time longer distance travelling. The reason behind this is two-wheeler have comparatively low maintenance, easy handling and less price. But, there is one concern about the this vehicle is the physical security issue [1]. If we compare the damage due to two-wheeler accidents and the other vehicle accidents then we will get that damage in two-wheeler is more insane and deadliest. In last 15 years accidents were increased extensively, in that specifically two-Wheeler accidents are comparatively more than other transport vehicles such as bus, car and also the majority about 90% of the rider didn't wear the helmet so that they were seriously injured or at some serious cases death occurred [2], so that use of helmet for the rider and also the pillion rider is must.

The new innovation for helmet detection and road safety is the smart helmet i.e. if a rider doesn't wear the helmet then that bike or two-wheeler won't start, but this implementation in real world is not that cost effective and the optimum solution [3]. The special feature of smart helmet is it can call contacts from emergency contact list whenever that helmet senses the considerable impact at the time of unfortunate accident. The system nowadays used to detect the helmet is manual; it means the traffic police need to capture the image for proof, but this the age of AI, machine learning and deep learning so that, in India some major metropolitan cities the high resolution cctv cameras are used to detect illegal activities, speed violations and also helmet detection purpose [4, 5]. The algorithm available for specifically helmet detection are not that much optimized and not able to detect the helmet of pillion rider [6]. The proposed model in report is focused on YOLOv4 (object detection model), other version of YOLO are good but not accurate as YOLOv4 [7]. This YOLO is actually a deep learning algorithm in which there are many convolution layers, addition of some more layer to detect the smaller details [8, 9].

2 Literature Survey

Wen et al. [10] proposed a model in that they used the technique like Image Descriptors and Classifiers for helmet detection purpose. Specifically For vehicle classification, wavelet transform. The HOG i.e. Histogram of oriented gradient and CHT i.e. Circular Hough Transform is mainly used for feature extraction of input images.

The accuracy is high i.e. 97.66% for the vehicle classification and for helmet detection it is about 91% [11].

Hu et al. [12] proposed a model, which mainly focuses on the detection of mask, helmet and number plate detection and this paper is published in 2021. They used the YOLO algorithm Version 3 and canny edge detection technique.

Now, the accuracy of the model is a very important constraint. Their accuracy through YOLOv3 [13] is about 95% for the vehicle classification which is decent,

but the accuracy is 90% for the helmet detection which is less as we compare to other models and the mask detection accuracy is about 99% [14]. But, one more concern is that it requires large datasets for training and testing purposes, it is very much time consuming. The detection of helmet and mask of pillion rider is not done in this paper [15].

Hu and Li [16] proposed a model which uses Yolov4's deep neural network architecture for the helmet recognition and also the accuracy is pretty much good i.e. 95%, this paper mainly focuses on the helmet detection of construction site workers.

They used a data set for the practically visible light which is sensitive to the human eye. For this they commit some changes in the properties of the images like noise, intensity, brightness to understand situations in real time data. They used different image enhancement techniques like first the input image undergoes some process like (scaled, flipped and clipped) to increase the accuracy of the proposed model. They also mentioned that when dealing with image occlusion, target overlap etc. [17]. They are using random erase and grid mask and other techniques which are the same as the avoidance of unnecessary ROI in the process of feature extraction and then selection of most appropriate rational part in the image area [18].

Raju et al. [18] proposed the model of object detection using YOLO v1. In which they modified the yolov1 algorithm. They mainly focused on improvement in 3 areas like loss function, inception structure and spatial pyramid pooling layer [19].

For the analysis of modified models, they prefer pascal VOC dataset [20], but the average accuracy was about 65% which is better than the basic model of yolov1.

Krishna and Reddy [21] proposed the model for automatic detection of helmets on a real time video [22]. They approached the problem in the simplest manner i.e. they converted the image to grayscale, then subtracting the background data and targeted on ROI (region of interest). For classification purpose they prefer the SVM because of the robustness of that classification algorithm. The accuracy they got was about 93% which was impressive with the basic model.

3 Architecture of YOLOv5

The crucial and very useful contribution of YOLOv5 is to translate the analysis done by Darknet [23] (which is the most important architectural part of the framework as it contains configuration files) to the PyTorch framework. The darknet framework is developed entirely in C and provides first-class fine-grained control over the network's activities. YOLOv5 composes the model configuration in .yaml, as contradictory to .cfg files in Darknet. The main difference between these two formats is that the .yaml file is built to enumerate the varied layers that the network comprises and then multiply those by the total number of layers in each block [24]. This updated yaml format sense like the following:

YOLO architecture is based on 3 items:

Backbone—It is CNN (Convolutional Neural Network) that gathers and generates image options at numerous granularities.

Neck—This is a sequence of layers to combine and mix image options to pass them forward to prediction.

Head—Collects the options from the neck and takes bounding boxes and sophistication predicting steps.

4 Proposed System

As we have mentioned the architecture for the YOLOv5l above, In the proposed model for adding a new layer, the head and the backbone parameters in the Yaml configuration file for YOLOv5l are changed. Previously the head consisted of three feature maps $52 * 52 * 255$, $26 * 26 * 255$, $13 * 13 * 255$. In the proposed technique a $104 * 104 * 255$ feature maps have been added for the detection of extra small features. After experimenting with the proposed model this model can also detect whether the pillion rider is wearing a helmet or not with the help of an extra added layer. So the model also works for 3 people sitting on the vehicle and detects whether they are wearing helmets or not. The proposed model is efficient and convenient for any OpenCv developer for specific reasons. Figure 1 shows improved architecture of YOLOv5.

5 Results and Discussion

5.1 Experimental Environment

Experimental environment used for this project: Intel(R) Core i5-9300H CPU 8 GB RAM, GPU NVIDIA GeForce GTX 1650, 64 bit operating system.

5.2 Dataset

Experimental dataset contains images of riders wearing helmets and riders who don't wear helmets and their annotations into YOLOV5 format. This proposed model used data which was gathered from Kaggle, Google and used some random images for the experimentation part. Experimental dataset contains 555 images for training and their annotations and 63 validation images and their annotations. Training and testing data is given in Table 1.

As there is no sufficient data available for bike rider helmet detection, this proposed model uses some images for bicycle riders also for more accuracy and as there is so much similarity between bike helmets and bicycle helmets. Figure 2 shows training dataset.

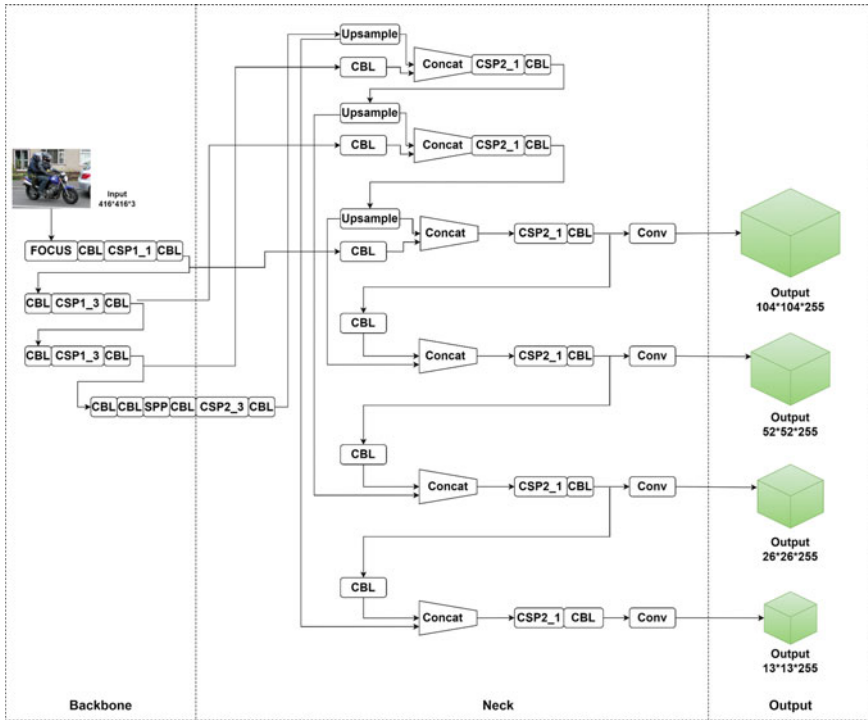


Fig. 1 Improved architecture of YOLOV5

Table 1 Training and testing data

Number of training images	555
Number of testing images	63

5.3 Results

This proposed model mainly focuses on detection of rear person helmets. For detection of the rear person’s helmet one detection head is added to the architecture of YOLOV5 and it gives results as follows. This proposed model gives better results for helmet detection and non-helmet detection also. Detection of helmet results are given in Fig. 3.

This proposed model gives precision and recall graphs as shown in Fig. 4. These results/graphs show precision and recall values for 100 epochs. After each epoch it will calculate value and Fig. 4 is a graph for that precision and recall values. In the Fig. 5, the accuracy and mAP values are shown. The proposed model gives accuracy about 89% for helmet detection as mAP value is about 0.89. This proposed model has given 100 epochs for training. Precision and recall values are also taken under consideration. Figure 4 shows precision and recall graph over 100 epochs.



Fig. 2 Training dataset

mAP value which gives accuracy measure shown in Fig. 5. In Fig. 6 Box loss and Object loss are shown. It helps much in the proposed model for YOLOV5. Results are based on detection of the pillion rider helmet. Accuracy for helmet detection is about 89% and accuracy for non helmet detection is about 75% as this model contains two classes helmet and non-helmet.

6 Conclusion

The proposed technique focuses on producing a ROI region of interest detector model. The output that we get by using YOLOv5 modified algorithm is better as compared with different object detection techniques [19, 25]. Pytorch procedure for the training is extensively useful for the enhancement in the performance of YOLOv5, at the same time the yolov5 and yolov4 are similar architecture wise. Proposed model gives accuracy about 89% for helmet detection of rider and pillion riders as mAP value is about 0.89 and for non-helmet rider detection is about 75% for riders as well as pillion riders.

The helmet is a very important safety guard for the workers, two-wheeler drivers and also for cyclist. Due to the lack of awareness and not taking seriously about wearing the helmet become the reason of many accidents. To help the traffic department by digitally monitoring the helmet we proposed this improved model YOLOv5.

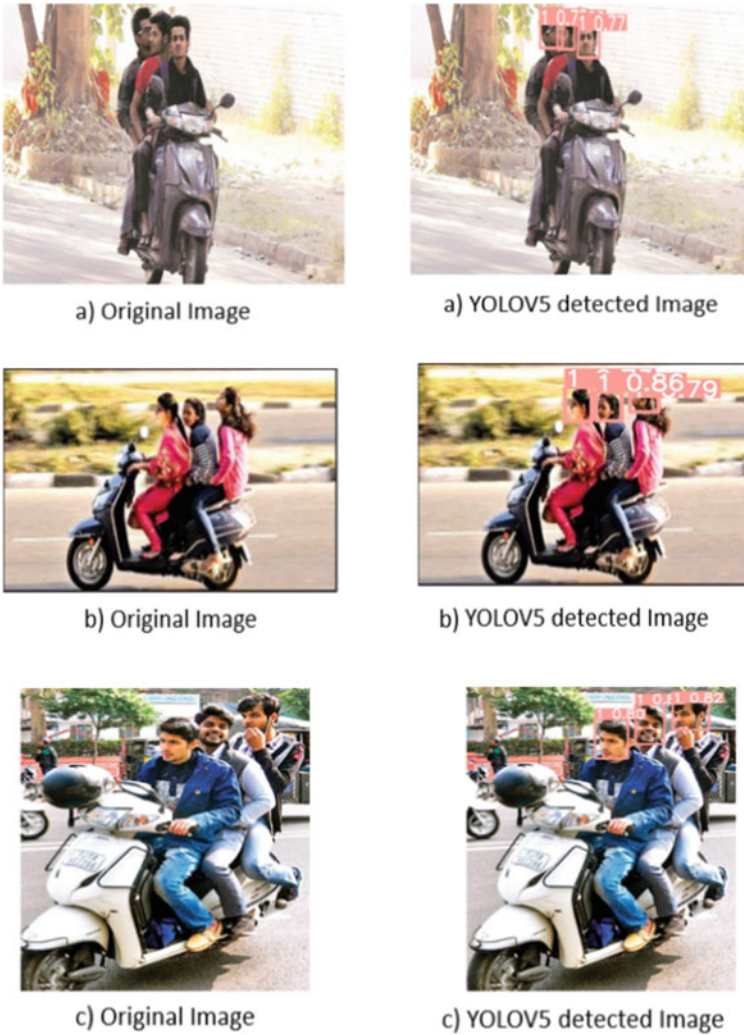


Fig. 3 Detection of helmet results

After that the gathered data set was annotated and the YOLOv5 model was trained and tested using different parameters. The experiment result displays that yolo v5 have better detection speed which is about 110 FPS for the real time helmet detection [26].

The YOLOv5 is currently a new and evolving technology in object detection techniques. There is scope for improvement in accuracy and some minor modification in the architecture part of YOLOv4 which is similar to YOLOv5 architecture wise [27]. We are planning to use optimized and updated dataset for more accurate

Fig. 4 Precision and recall graphs over 100 epochs

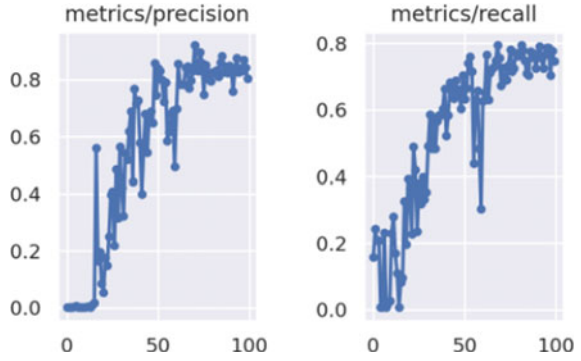


Fig. 5 mAP value which gives accuracy measure

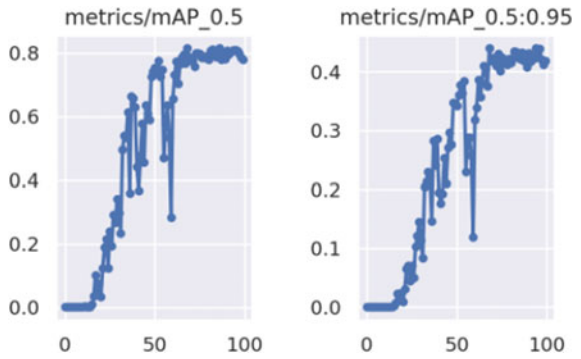
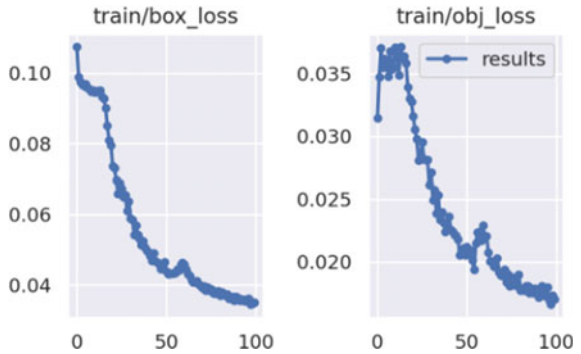


Fig. 6 Box loss and obj loss



detection. This algorithm can be optimized to use in complex object detection. Optimized version of YOLOv5 [28] is very useful in CCTV cameras (upto 60 fps camera resolution) for precise detection of riders with or without helmet.

References

1. Naik C, Holla HV, Meleet M (2021) Motorcycle traffic rule violation detection and license plate recognition using YOLO. *Int J Adv Res Comput Commun Eng* 10(8). Reza ZN (2019) Real-time automated weld quality analysis from ultrasonic B-scan using deep learning. Doctoral dissertation (University of Windsor (Canada))
2. Megalingam RK, Babu DHA, Sriram G (2021) Concurrent detection and identification of multiple objects using YOLO algorithm. In: *Symposium on image, signal processing and artificial vision (STSIVA)*
3. Bochovski A, Wang C-Y, Liao H-YM (2020) YOLOv4: optimal speed and accuracy of object detection. *arXiv: Computer Vision and Pattern Recognition*
4. Cao R, Li H, Yang B, Feng A, Yang J, Mu J (2020) Helmet wear detection based on neural network algorithm. In: *2020 international conference on applied physics and computing (ICAPC 2020)*. <https://doi.org/10.1088/1742-6596/1650/3/032190>
5. Fathima S, Chandana U (2019) Bike authentication by helmet using faster R-CNN using machine learning. *Int J Res* 8(9)
6. Bochovski A, Wang C-Y, Liao M (2021) Scaled YOLOv4: scaling cross stage partial network. In: *Computer vision foundation CVPR conference 2021*
7. Huang YQ, Zheng JC, Sun SD, Yang CF, Liu J (2020) Optimized YOLOv3 algorithm and its application in traffic flow detection. *Appl Sci*
8. Silva R, Aires K, Veras R (2014) Helmet detection on motorcyclists using image descriptors and classifiers. In: *Conference: 2014 27th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*. <https://doi.org/10.1109/SIBGRAPI.2014.28>
9. Gangadeep S (2019) Study of object detection methods and applications on digital images. *IJS DR* 4(5)
10. Wen P, Tong M, Deng Z, Qin Q (2020) Improved helmet wearing method based on YOLOv3. In: *Artificial intelligence and security*
11. Fuchuan, GongXin W (2019) Research on safety helmet wearing YOLO-V3 detection technology improvement in mine environment. *J Phys Conf Ser*
12. Hu J, Gao X, Wu H, Gao S (2019) Detection of workers without the helmets in videos based on YOLO V3. In: *2019 12th international conference on image and signal processing, biomedical engineering and informatics (CISP-BMEI)*. <https://doi.org/10.1109/CISP-BMEI48845.2019.8966045>
13. Dhyanjith G, Manohar N, Raj AV (2021) Helmet detection using YOLO V3 and single shot detector. In: *2021 6th international conference on communication and electronics systems (ICCES)*. <https://doi.org/10.1109/ICCES51350.2021.9489194>
14. Maliye S, Oza J, Rane J, Pathak N (2021) Mask and helmet detection in two-wheelers using YOLOv3 and Canny edge detection. *Int Res J Eng Technol (IRJET)* 8(4)
15. Tang M, Le QV (2019) EfficientNet: re-thinking model scaling for convolutional neural networks. In: *Proceedings of IEEE conference on machine learning*
16. Hu L, Li Y (2021) Micro-YOLO: exploring efficient methods to compress CNN based object detection models, *ICAART*
17. Zheng W, Chang J (2021) Helmet detection based on an enhanced YOLO method. Part of the lecture notes in electrical engineering book series (LNEE, vol 653)
18. Raju S, Paul SP, Sajeer S, Johnny A (2020) Detection of Helmetless riders using faster R-CNN. *Int J Innov Sci Res Technol* 5(5)
19. Wang J, Zhu G, Wu S, Luo C (2021) Worker's Helmet recognition and identity recognition based on deep learning. *Open J Model Simul* 9(2)
20. Tang M, Pang R, Le QV (2020) EfficientDet: scalable and efficient object detection. In: *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)*
21. Krishna NM, Reddy RY (2021) Object detection and tracking using YOLO. In: *International conference on inventive research in computing applications*
22. Shrivastav S, Divekar AV, Anilkumar C, Naik I, Kulkarni V (2021) Comparative analysis of deep learning image detection algorithms. *J Big Data* 8 Article 66

23. Dixit KGS, Chadaga MG (2019) Evaluation and evolution of object detection techniques YOLO and R-CNN. IJRTE 8(2S3)
24. Mukul M, Tiwary B, Bahria S, Nartam P, Suryavanshi V (2021) Real-time Helmet detection of bike riders. Int J Res Appl Sci Eng Technol (IJRASET) 9(VI)
25. Rohith CA, Nair SA, Nair P, Alphonsa S, John NP (2019) An efficient Helmet detection For MVD using deep learning. In: International conference on trends in electronics and informatics (ICEI)
26. Viola P, Jones M (2017) Robust real time object detection. Int J Comput Vision 4
27. Nepal U, Eslamiat H (2022) Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs. Sensors 22:464
28. Thakur N, Nagrath P, Jain R, Saini D (2021) Object detection in deep surveillance

Stock Market Trend Prediction Along with Twitter Sentiment Analysis



Priyadarshan Dhabe, Ayush Chandak, Om Deshpande, Pratik Fandade, Naman Chandak, and Yash Oswal

Abstract The Stock Market Prediction and Analysis has always been one of the most challenging tasks (Polamuri and Mohan in A survey on stock market prediction using machine learning techniques, 2019; Parmar et al. in First international conference on secure cyber computing and communication (ICSCCC), pp. 574–576, 2018). The variety of influences and unpredictability beats even the heavyweights to ground when it comes to successfully analyzing Stock Price data. In the proposed System, we have designed and successfully built a Machine Learning model using Long-Short Term Memory (LSTM) algorithm which helps for prediction of stock price data. We have done experimentations for better training, accuracy and results, on used data. The proposed system is also deployed on a web application which helps eliminate/reduce the difficulty of its use for the users. The model also works on the real-time data as we are using Yahoo finance API for getting updated data for model training and prediction. Lastly, The Indian stock market prices are also heavily driven by public sentiments which have for providing a better public opinion upon a particular stock. To help our users tackle this, we have added twitter sentiment analysis as a feature which provides us results in term of percentages of positive and negative sentiments within the tweets in the public domain at present about a particular stock, achieving a better opinion on a particular stock for the users.

P. Dhabe (✉) · A. Chandak · O. Deshpande · P. Fandade · N. Chandak · Y. Oswal
Department of Information Technology, Vishwakarma Institute of Technology, Pune,
Maharashtra, India
e-mail: priyadarshan.dhabe@vit.edu

A. Chandak
e-mail: ayush.chandak18@vit.edu

O. Deshpande
e-mail: om.deshpande18@vit.edu

P. Fandade
e-mail: pratik.fandade18@vit.edu

N. Chandak
e-mail: naman.chandak18@vit.edu

Y. Oswal
e-mail: yash.oswal18@vit.edu

The resulting model successfully gives us a prediction graphs as an output when given a particular stock on the proposed web application. We obtained least error in prediction, for Asian Paints data for the split of 80:20, using 75 epochs.

Keywords Stock market prediction · LSTM · Yahoo finance · Sentiments · Twitter

1 Introduction

For business analysts and researchers, forecasting the stock marketplace rate is usually a task. Stock market costs estimation isn't most effective, a thrilling however additionally tough vicinity of studies [1, 2]. Predicting the stock market with complete accuracy may be very tough as external entities such as social, mental, political, and financial have a top notch and large impact on it. The characteristic of the data related to the stock market is commonly time version and nonlinear. Prediction of inventory marketplace performs a crucial position in stock enterprise. If traders lack enough information and knowledge, then their funding can go through the greatest loss. Traders must expect the destiny stock fee of agencies a good way to attain excessive earnings. Diverse prediction techniques were developed to do predictions on the stock market as it should be. This model considers a company's former equity share value and uses the RNN method called as LSTM. The data set was obtained directly from Yahoo Finance. The proposed approach uses a share's historical data and makes predictions on a given attribute. Attributes of a share can be day high, day low, opening price, prior day opening and close price, day of trading, total trade quantity and turnover are all characteristics of shares. The said model apply time series analysis to foretell the share price over a given time span. Long Short-Term Memory (LSTM) is one of many forms of Recurrent Neural Networks (RNN) that can capture input from previous stages and use it to predict the future. Given the effect of social media on our daily lives, understanding public sentiment of a stock market company through various social media platforms has become a need in today's world. It's crucial to track public opinion while making a decision with respect to them and deciding the next step. For doing the same, social networking websites are a good place to start. Twitter is popular platforms for candid public sentiment analysis on a variety of topics. With the use of Tweepy, TextBlob, and Data Frame by pandas this study intends to examine public mood via Tweets from Twitter of any particular Stock in recent time. Then Later indicate the positivity, negativity, and neutrality of a tweet based on the Polarity score and visualize the data to gain a clearer picture of the attitude that dominates. This research project seeks to analyze social media data, with a focus on Twitter, in order to compute sentiment scores and depict them, with the goal of explaining people's social media sentiment of any particular stock or a listed company.

2 Literature Survey

In the work [3], researchers used KNN and nonlinear regression for stock price prediction. They used data of 6 companies from Jordanian stock exchange to help all the stakeholders. Their experience says that KNN is robust, reasonable and coherent method in this case. It also gives small error ratio. Mehtab and Sen [4] have shown that how LSTM can be employed in practice for foretelling stock prices of NIFTY 50 changes on the National Stock Exchange (NSE) of India. It is among the most recent ideas in this work domain. The authors created three forecasting models using daily stock prices. The models' prediction accuracies were then assessed due to their ability to predict the perturbation patterns of NIFTY index's nearer value over a one-week time horizon. Authors used NIFTY, fifty index values in the span, Jan 2018 to Jun 2019, for testing [4]. The work proposed in [5], uses Convolutional Neural Networks (CNN) and multivariate time series data for stock prediction. The authors' suggested prediction model combines a CNN's learning capacity with prediction validation to achieve a high degree of accuracy in expecting future index values of NIFTY and the trend in movement. The authors offer three distinct CNN designs, each with a different number of variables utilized in predicting, count of systems sub-models and input data size for model training. The CNN-based multivariate foretelling prototype was considerably worthy in predicting the weekly movement of NIFTY index values, according to the experimental data. LSTM networks have also been proposed for stock price prediction. Use of LSTM [9] in stock price foretelling is described in [5]. In the study [6], the comprehensive procedure of evolving a stock price foretelling model utilizing the ARIMA model is given. A stock price prediction algorithm is integrated with available data from New York Stock Exchange (NYSE) and the Nigerian Stock Exchange (NSE). The results showed that the ARIMA model has a lot of potential for largely short-term prediction and might compete well with conventional stock price prediction strategies. This can guide stock market investors for taking profitable investment decisions. ARIMA models may compete quite well with developing forecasting approaches in short-span foretelling based on the findings obtained [6]. This paper [7] has proposed a hybrid model that combines the benefits of a CNN and a LSTM approach they talk about the different works related to pattern reading and prediction, providing us the comprehensive view about the prediction techniques. We also studied the way they combined the models which is interesting and inspirational for our approach towards our model.

The main outcome of this study is, suggested CNN-LSTM model beat 17 baseline time series forecasting algorithms for test as well as foretelling data, along with least average values of RMSE, MAPE and RRMSE. Finally, while individual CNN and LSTM models predict verified COVID-19 occurrences time series well and efficiently, combining these two models in projected CNN-LSTM encoder decoder structure greatly increases performance of forecasting. In addition, the suggested model exhibited acceptable predication demonstrated that the suggested model produced acceptable predicting results with limited of Date 2022-01-14 Words 744 Characters 5121 Page 1 of 2 data was available. This proposed technique has helped achieve

improved accuracy for the COVID-19 cases prediction, and we may apply it to our stock market prediction model as well, but we will have to careful at the same time honoring the exclusivity of both the applications [7].

3 Dataset

3.1 Yahoo Finance

Collection of data is most crucial task in the research. The dataset is collected from Yahoo finance which affords monetary news, statistics and statement together with stock charges, press releases, monetary reports, and unique content. This dataset is perfect as you can view historic price, dividend, and cut up information for most quotes in Yahoo Finance to forecast the destiny of an organization or advantage marketplace perception (Fig. 1).

3.2 Twitter Sentiment Analysis

Twitter is a large dataset, for working with sentiment evaluation using twitter statistics [8, 10]. The statistics extraction is critical. Twitter gives access to tweets using their APIs. The data set accumulated from Twitter API to apply tweets sentiment evaluation for the statistics is collected as positive, negative or neutral tweets. This is done with the help of polarity analyzer this is a technique of identifying attitudes in textual content statistics about a subject of interest. Its miles scored the use of polarity values that variety from 1 to -1. Values toward 1 suggest more positivity, while values closer to -1 indicate extra negativity [8] (Fig. 2).

Fig. 1 Price of stock (dataset) snap

	High	Low	Open	Close	Volume
0	7.619643	7.520000	7.611786	7.528071	352410800.0
1	7.660714	7.585000	7.622500	7.643214	493729600.0
2	7.699643	7.616071	7.664286	7.656429	601904800.0
3	7.686786	7.526786	7.656429	7.534643	552160000.0
4	7.571429	7.466071	7.562500	7.520714	477131200.0
...
2009	43.855000	43.625000	43.669998	43.752499	65397600.0
2010	42.867500	42.419998	42.700001	42.642502	132742000.0
2011	42.695000	42.427502	42.525002	42.650002	85992800.0
2012	42.962502	42.819999	42.750000	42.770000	65920800.0
2013	42.647499	42.305000	42.630001	42.307499	103999600.0

Fig. 2 A sample tweet



3.3 Standardization

Standardization is the data transformation with respect to mean value and scaling it dividing their variance. Post standardization mean and the variance becomes 0 and 1, respectively. Standardization helps to improve model performance too.

The standardization (Z) formula is as given in (1) for N samples.

$$Z = \frac{x - \mu}{\sigma} \tag{1}$$

where mean and variance are computed as in (2) and (3), respectively.

$$\mu = \frac{1}{N} \sum_{i=1}^N (x_i) \tag{2}$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} \tag{3}$$

4 Work Flow Diagram

Following Fig. 3 indicates the flow diagram of trend prediction and Fig. 4 shows planned work flow of twitter analysis.

Fig. 3 Trend prediction work flow diagram

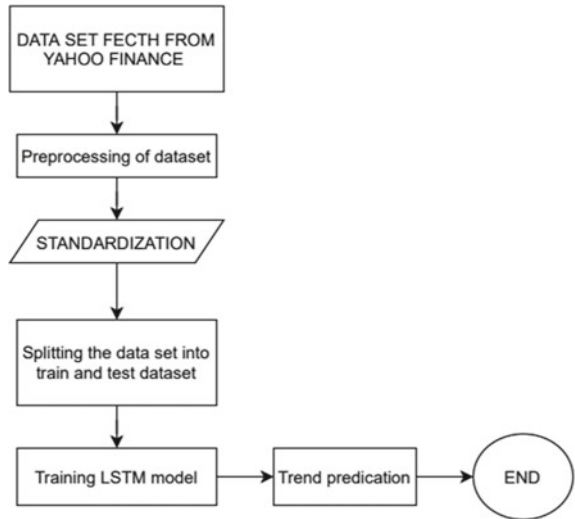
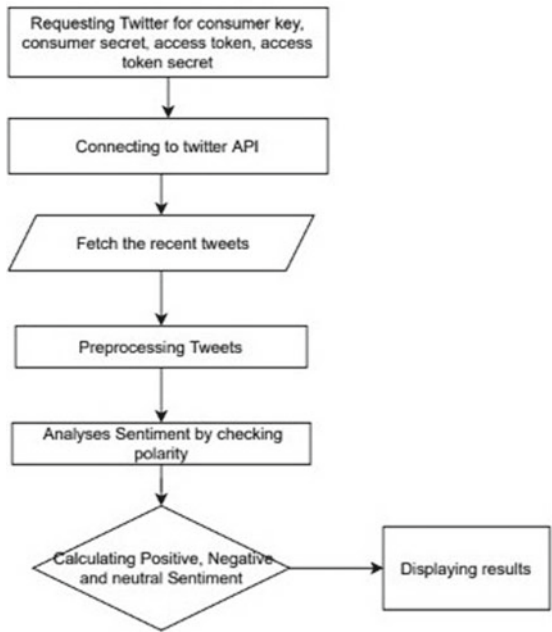


Fig. 4 Twitter analysis workflow diagram



5 Methodology

5.1 Stock Trend Predication

Data Splitting

The dataset is divided in to disjoint training and testing sets. The model learns with supervised training set and reason for the test set. Test dataset is used to evaluate the accuracy of our model’s predictions. We used, both, 70–30% and 80–20% data split for the experimentation.

Long Short-Term Memory (LSTM)

Given the problem statement, we understand that prediction in the stock market take place using pattern creation and iterations of those patterns historically and serving the exact purpose we have the “Real-Time Recurrent Learning” (RTRL), but when looked throughout, it has a major issue which has to be addressed.

The Conventional RTRL when dealing with errors which go backwards in time either leads to oscillations of weights or learning to bridge up a gap takes a lot of time lag, which hints us to the solution and the used model i.e., LSTM.

The LSTM is a kind of recurrent network in conjuncture with the gradient-based learning. The original developed model promised to keep short-term memory as long as 1000 consecutive inputs. One of the important components which contribute majorly to the Architecture is Memory Cells and Gate Units. To eliminate the possibility of perturbation by irrelevant inputs, multiple input gate units and output gate units have been introduced expanding the constant error carrousel, result is a more complex unit know as a memory cell. Given below in Fig. 5, is the figure which represents the memory cell.

The Memory Cell has 3 doors to be generalised, entrance, door with a view and an info door. This cell collects data at appropriate set timings which serves as the long short-term memory of our model [9]. A significant thing to understand and decrypt is that overhead door has both the responsibility to loads and capacity to start up the state cell. Also, Memory from the past cell can be allowed to pass as it is, rather than expanding and decreasing exponentially at each layer of network, and loads can have their ideal quality as quick as possible. This also solves issue-as a value put in cell is not adjusted every time, the inclination will not be hampered towards our Indian trading entities i.e. NSE and BSE (Fig. 6).

Fig. 5 A memory cell of LSTM, referred from the context of “LSTM by Sepp Hoch Reiter & Jurgen Schmid Huber”

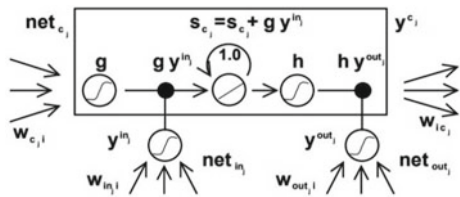


Fig. 6 LSTM model

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 100, 50)	10400
dropout (Dropout)	(None, 100, 50)	0
lstm_1 (LSTM)	(None, 100, 60)	26640
dropout_1 (Dropout)	(None, 100, 60)	0
lstm_2 (LSTM)	(None, 100, 80)	45120
dropout_2 (Dropout)	(None, 100, 80)	0
lstm_3 (LSTM)	(None, 120)	96480
dropout_3 (Dropout)	(None, 120)	0
dense (Dense)	(None, 1)	121

Total params: 178,761
 Trainable params: 178,761
 Non-trainable params: 0

5.2 Twitter Sentiment Analysis

Tweepy will be used to extract data in order to perform sentiment analysis. Tweepy is a Python module that allows you to access the Twitter API, which allows you to extract and access data on a range of topics. To use the Twitter API, you must first create a developer account and get it accepted by twitter through an application procedure. Once we obtain the access to it and have authorisation for a developer account, we generate consumer tokens and access tokens and define them as variables. We also utilise OAuthHandler and set_access_token to check the access tokens and authenticate the account. The same is demonstrated in Fig. 7.

The tweets search function is then used to scrape tweets, and we further search for tweets using a certain hashtag, language, and time. We must also specify the number of tweets to be extracted. When tweets are successfully extracted, they are stored in a Data Frame and labelled appropriately.

Fig. 7 LSTM model implementation

```

# keys and tokens from the Twitter Dev Console
consumer_key='Py0nJPLCN0F7Z3MNHtPCKgAx1'
consumer_secret='uMp5Ls2FzNiwPsImGMyPZARwzaCxIcsPGtClYsyIvC2xqnJdFD'
access_token='3553689916-HEuAPbMQxVOCpsQqHmyZgv8GJKKIiH0MmLxH5uyh'
access_token_secret='aIrPqZVLzV1sabwRhNzHUqCnlyEKPEAbso0SxzHjgltUm'

# attempt authentication
try:
    # create OAuthHandler object
    self.auth = OAuthHandler(consumer_key, consumer_secret)
    # set access token and secret
    self.auth.set_access_token(access_token, access_token_secret)
    # create tweepy API object to fetch tweets
    self.api = tweepy.API(self.auth)
except:
    print("Error: Authentication Failed")
  
```

We begin cleaning the tweets after the data has been scraped and placed in a data frame. Manipulation of any type of textual data should be approached with caution, as changing incorrect data might lead to biased analysis and, ultimately, false results. We then do data purification to remove various symbols and usernames from the tweets in order to ensure more accurate sentiment analysis computations.

Following data cleansing, we employ Text Blob’s sentiment function to compute Subjectivity and Polarity scores, which are critical for classifying the extracted tweets.

- Polarity is a float value between -1 and 1 that indicates whether a text is positive or negative. In a nutshell, the Polarity score aids in the analysis of a text’s emotion or attitude.
- Subjectivity is a float value between 0 and 1 that determines whether a text is more subjective or objective. A subjective sentence is a piece of literature with a tone that leans more towards an opinionated expression. An objective sentence is a type of textual material with a tone that favours factual expressions. The Polarity score is also used to categorise tweets into positive (> 0), neutral ($=0$), and negative (< 0) categories. By categorising tweets into the supplied tags, we visualise our final results in order to evaluate trends from our textual data [10].

6 Experimental Results

After training the model, result of our testing has shown different results with number of epochs and train-test split. The data is of stock ASIAN PAINTS from “1/1/2010” to “1/12/2021” with total of 2939. The model is trained and test with 2 combinations of split i.e., 70:30 (train: test) and 80:20 (train: test).

To evaluate model, we use Root Mean Square Error (RMSE), which is a de-facto way to measure error in predicting data. It is defined in (4) for n values.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}} \tag{4}$$

where \hat{y}_i represents the predicted values and y_i are test values for $i = 1, 2, \dots, n$ (Fig. 8).

In Table 1 the RSME of predicate value is given with respect to epochs. First with split of 70% (2057) train and 30% (882) test we can observe that with increasing no. of epochs, the RSME first decreases than increases. This is because of overfitting model.

When a model is trained with a large amount of data, it begins to learn from the noise and inaccuracies in the data set. The model then fails to predicate the input due to too many details and noise. This is known as Overfitting of model (Fig. 9).

In Table 2 the RSME of predicate value is given with respect to epochs. First with split of 80% (2352 samples) train and 20% (587 samples) test we can observe

Fig. 8 Epochs versus RSME

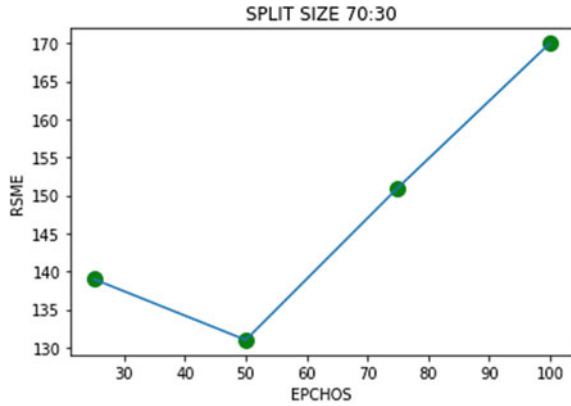
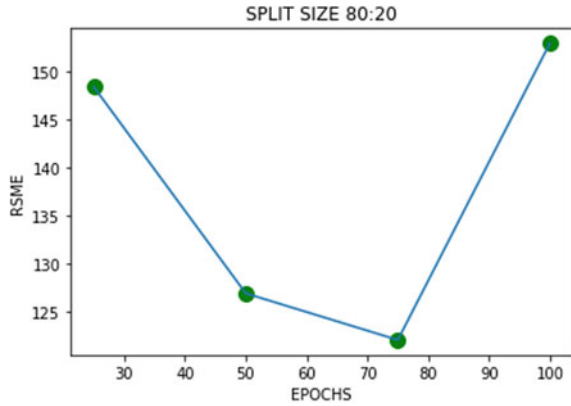


Table 1 Epochs versus RSME versus time

Split 70:30 (samples 2057:882)		
Epochs	RSME	Time taken to run epochs
100	170.10	2030
75	151.5	2325
50	131.43	1120
25	139.02	375

Fig. 9 Epochs versus RSME



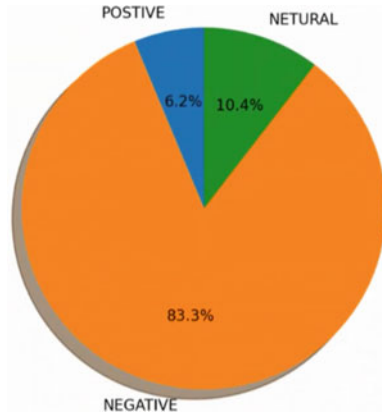
that with increasing no. of epochs, the RSME decreases drastically. In Epoch-75 we achieved lowest RSME value (122.42).

With the use of numerous charts imported from Matplotlib, various trends and conclusions are presented based on the sentiment computations that have been calculated. The retrieved texts are tokenized, and the words with the highest usage frequency are presented in the form of a word cloud.

Table 2 Epochs versus RSME versus time

Split 80:20 (samples 2352:587)		
Epochs	RSME	Time taken to run epochs
100	152.89	2360
75	122.42	2550
50	126.85	1050
25	148.35	540

Fig. 10 Visual representation of classification of tweets



As previously stated, the Subjectivity score is a number between 0 and 1 that defines if a tweet is more opinionated or factual, and the Polarity score is a number between -1 and 1 that classifies a tag as positive, negative, or zero.

Figure 10 depicts the distribution of various tweets depending on the count of tweets based on emotion tags assigned by Polarity scores. The visualization pie-chart reveals a nearly equal distribution of neutral and negative tweets, with a significant number of positive tweets. We may make a major conclusion from this, namely that while there appeared to large negative tweets as well as positive tweets, indicating people’s interest.

The trend predicated form test data set with respect to time and different epochs and split are plot below. Figures 11, 12, 13 and 14 are for the split of 70–30% and Figs. 15, 16, 17 and 18 are for splits of 80–20%.

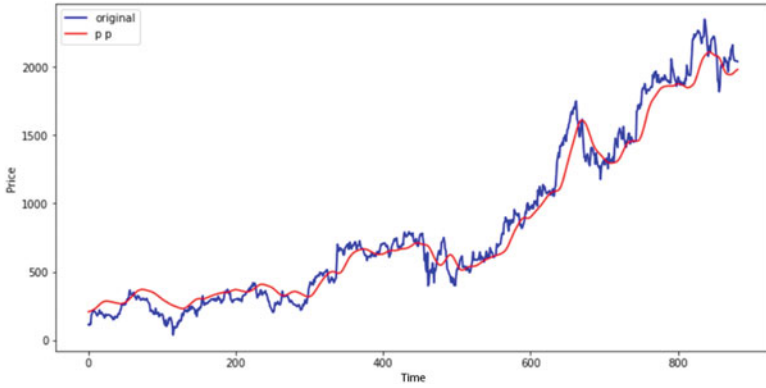


Fig. 11 Price versus time for 25 Epochs (split 70–30)

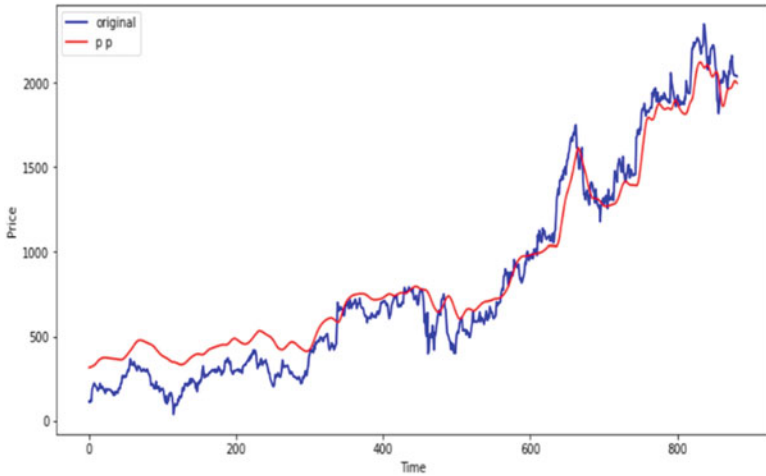


Fig. 12 Price versus time for 50 epochs (split 70–30)

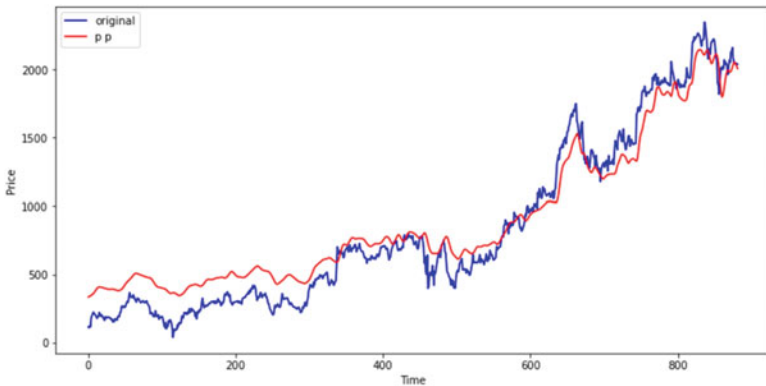


Fig. 13 Price versus time for 75 epochs (split 70–30)

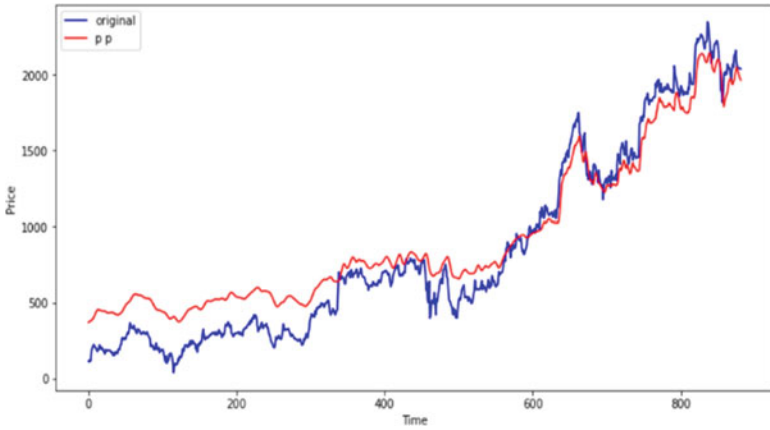


Fig. 14 Price versus time for 100 epochs (split 70–30)

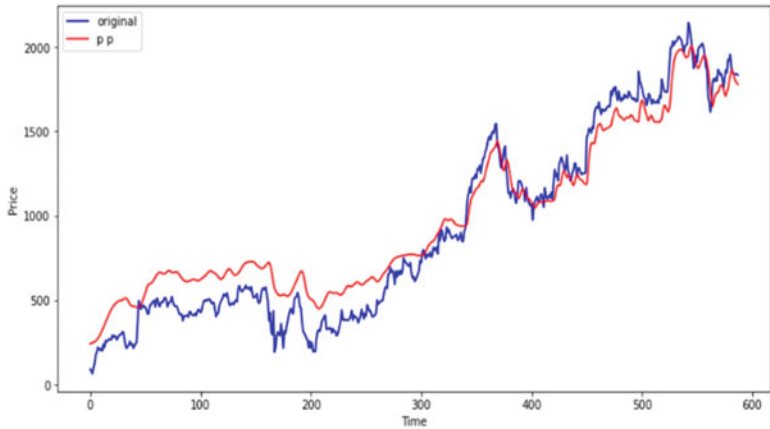


Fig. 15 Price versus time for 25 epochs (split 80–20)

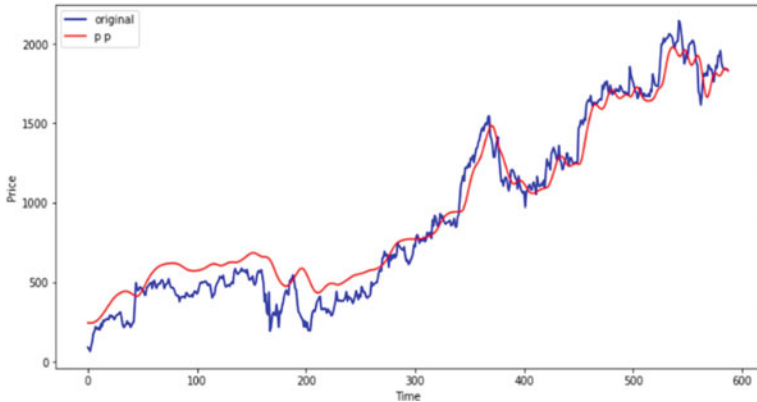


Fig. 16 Price versus time for 50 epochs (splits 80–20)

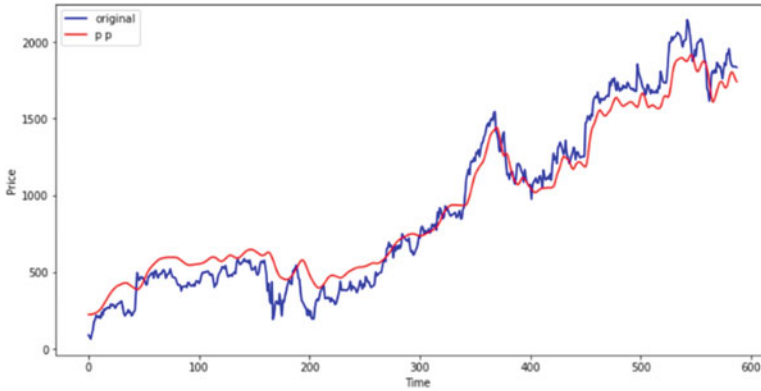


Fig. 17 Price versus time for 75 epochs (split 80–20)

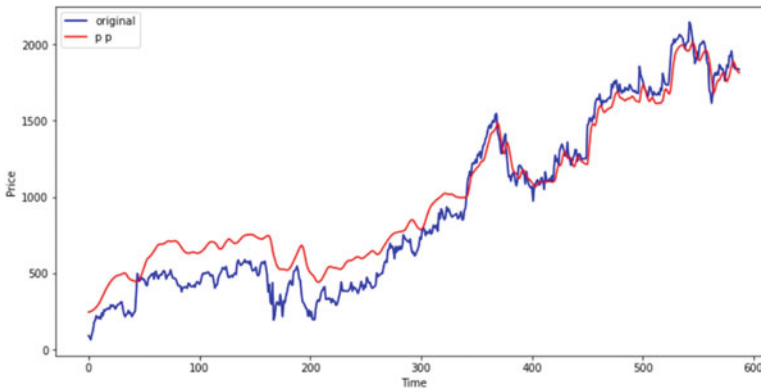


Fig. 18 Price versus time for 100 epochs (split 80–20)

7 Conclusion

This paper proposes LSTM model built to forecast future trend of STOCK and Sentiment Analysis model for determining the sentiment of asset trough twitter data feed. Model achieved the lowest RMSE in 75 Epochs with 80:20 (test: train) split of data set. Hence, we conclude that 75 epochs with 80:20 split predicate best trend predication of price with lowest error rate.

This can assist investors in gaining a significant financial benefit while maintaining a stable stock market environment. Investor can avoid huge draw down in finical market by using models in significant manner. In future work multiverse time series can be done by adding important feature in data set which will further improve the model.

Acknowledgements Our student's team would like to thank the Director, VIT, Pune, Prof. Rajesh Jalnekar, HOD IT Prof. Ghadekar and Associate Professor Dr. Priyadarshan Dhabe, for their continuous support, guidance and inspiration for the research work.

References

1. Polamuri SR, Mohan AK (2019) A survey on stock market prediction using machine learning techniques. ICDSMLA 2019, pp 923–931
2. Parmar et al (2018) Stock market prediction using machine learning. In: 2018 first international conference on secure cyber computing and communication (ICSCCC), pp 574–576. <https://doi.org/10.1109/ICSCCC.2018.8703332>
3. Stock price prediction using K-nearest neighbor (kNN) algorithm—scientific figure on ResearchGate. https://www.researchgate.net/figure/1-The-results-after-applying-kNN-algorithm-for-the-AIEI_tbl3_262456253. Accessed 14 Jan 2022 (L1)
4. Mehtab S, Sen J (2019) A robust predictive model for stock price prediction using deep learning and natural language processing: presentation. <https://doi.org/10.13140/RG.2.2.32046.66883> (L2)
5. Mehtab S, Sen J (2020) Stock price prediction using convolutional neural networks on a multivariate timeseries (L3)
6. Adebisi A, Adewumi A, Ayo C (2014) Stock price prediction using the ARIMA model. In: Proceedings—UKSim-AMSS 16th international conference on computer modelling and simulation, UKSim 2014. <https://doi.org/10.1109/UKSim.2014.67> (L4)
7. Lu J, Zhang Q, Yang Z, Tu M (2019) A hybrid model based on convolutional neural network and long short-term memory for short-term load forecasting 1–5. <https://doi.org/10.1109/PESGM40551.2019.8973549> (L5)
8. Balachander PSJB (2020) Sentimental analysis of Twitter data using Tweepy and Textblob. IJAST 29(3):6537–6544
9. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9:1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
10. Sarlan A, Nadam C, Basri S (2014) Twitter sentiment analysis 212–216. <https://doi.org/10.1109/ICIMU.2014.7066632>

A Study on MQTT Protocol Architecture and Security Aspects Within IoT Paradigm



M. Nimavat Dhaval and G. Raiyani Ashwin

Abstract In the Internet of Things (IoT) paradigm, boundless solutions have been designed and implemented to do effective and secure communication among its smart objects and its network. The outcome of effective and secure communication always relies on which IoT protocol has been used at the application layer. Generally IoT devices communicate using various IoT push protocols such as XMPP (Extensible Messaging and Presence Protocol), MQTT (Message Queuing Telemetry Transport), AMQP (Advanced Message Queuing Protocol) among which MQTT protocol is widely used protocol within IoT platform because it requires nominal resources as it's lightweight and efficient, it also support bi-directional communication among smart objects and cloud and MQTT also guarantees and support reliable message delivery through 3 Quality of Service (QoS) levels. This research paper focuses on key concepts on MQTT protocol architecture, basic security fundamentals such as identity, authentication, authorization and MQTT advance security fundamentals which includes X.509 client certification authentication, OAuth 2.0 and payload encryption.

Keywords IoT · MQTT · Security

1 Introduction to MQTT Protocol

1.1 MQTT Protocol Architecture

MQTT is an open source and light-weight protocol for communication between smart objects and its network over (Transmission Control Protocol) TCP. As MQTT is light weight, its energy consumption is very less compared to other push protocols,

M. Nimavat Dhaval
RK University, Rajkot, India
e-mail: dhaval.nimavat@rku.ac.in; dhaval.nimavat@gmail.com

G. Raiyani Ashwin (✉)
Nirma University, Ahmedabad, India
e-mail: ashwin.raiyani@rku.ac.in; ashwin.rkcet@gmail.com

and also supports low bandwidth communication. MQTT Protocol also ensures up to 3 levels of Quality of Service (QoS), Which is capable of managing how much data is transmitted and which types of handshakes are needed. MQTT is a widely used protocol in the IoT paradigm [1–3] (See Fig. 1).

MQTT uses the following fundamental concepts to transmit or receive data over network and connected smart objects.

- **Topics.** It identifies the category of data or message to be sent. It is about identifying and applying labels to the type of communication that takes place among publisher and subscriber. Such as if a temperature sensor senses change in environment, the topic or label might be allotted as “WEATHER_CHANGE_UPDATE”, and the sensed data will be transmitted as label “WEATHER_CHANGE_UPDATE” [2, 4, 5].
- **Publishers.** It is defined as a smart object which is connected to sensors and responsible for transmitting messages over TCP. Ex. Arduino or Raspberry Pie where different sensors are connected to it [2, 4, 5].
- **Subscribers.** It identifies the receiver who is interested to receive data or messages for specific one or more interested topics. Ex. Clients who subscribe to the “WEATHER_CHANGE_UPDATE” topic, will receive notification whenever changes will be detected [2, 4].
- **Broker.** It refers to a server which is responsible to connect publisher and subscribers as per the topics which subscriber had subscribed to. It is also defined as an agent [2, 4].

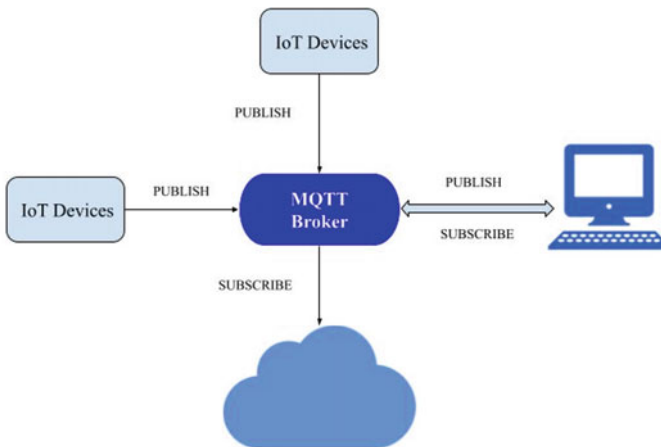


Fig. 1 MQTT architecture

1.2 Why MQTT?

Majority within the IoT environment, MQTT protocol is widely used because of certain reasons, as MQTT protocol occupies and consumes less energy from connected IoT devices which makes it lightweight and efficient protocols. It also support bi-directional communication between smart objects and it’s network over TCP.

MQTT can be subsidize millions of connected smart objects over unreliable networks as well. About security aspects, it also ensure identity, authentication and authorization using advance security mechanism which includes which includes X509 client certification authentication, OAuth 2.0 and payload encryption [2, 3].

As far as server utilization is concern, hereby we can observe the statics that depicts mean response time taken or utilize by various protocols such as CoAP, MQTT, AMQP and XMPP [6, 7].

Figure 2 shows MQTT and XMPP protocols utilize very nominal server processing. It’s due to skipping handshakes while connecting to the clients, which results into less mean response time [8, 9].

Figure 3, depicts the comparison between CoAP and MQTT. As far as performance is concern, CoAP depends upon UDP for communication to preform well at server utilization but required more mean response time compare to MQTT. On other side by using MQTT protocol, we can decrease server utilization within minimum mean response time [7, 8, 10].

Figure 4, depicts the comparison between XMPP and MQTT, by using this approach, XMPP protocol use less server utilization as it directly communicate

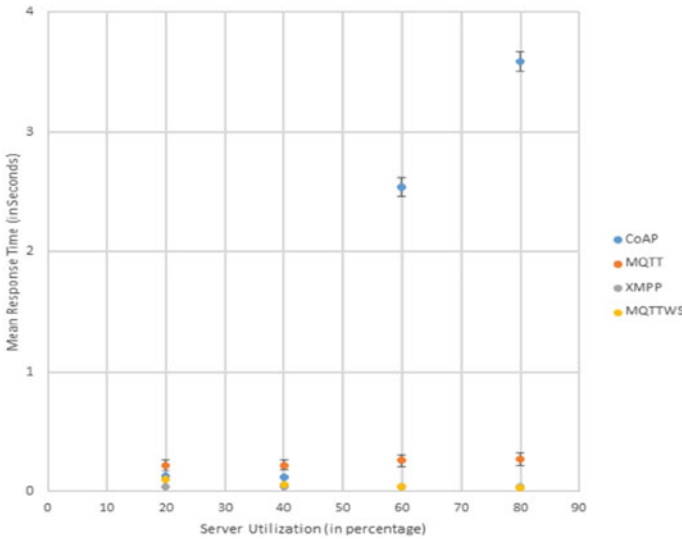


Fig. 2 Mean response time—IoT protocols

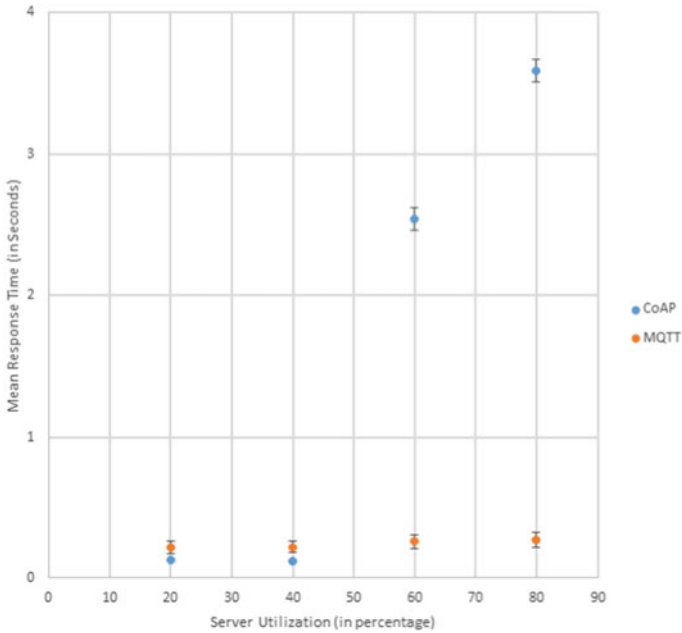


Fig. 3 Mean response time—MQTT versus CoAP

to client without handshaking. While MQTT protocol use marginal server utilization more compare to XMPP as it perform handshaking while communicating with clients. Thus, MQTT mean response time increase due to performing handshaking while managing each client communication [8].

2 Security Aspects Within MQTT

Mainly three aspects within MQTT protocol impacts on such as identity, authentication, and authorization.

2.1 Identity

Subscriber needs to establish a connection with a broker to communicate for requested topics. MQTT protocol ensures each subscriber by identifying subscriber id while requesting for a connection. Most of the subscriber has a unique client identifier, which recognized as universal unique identifier (UUID) or a MAC address of the subscriber used to connect with publisher.

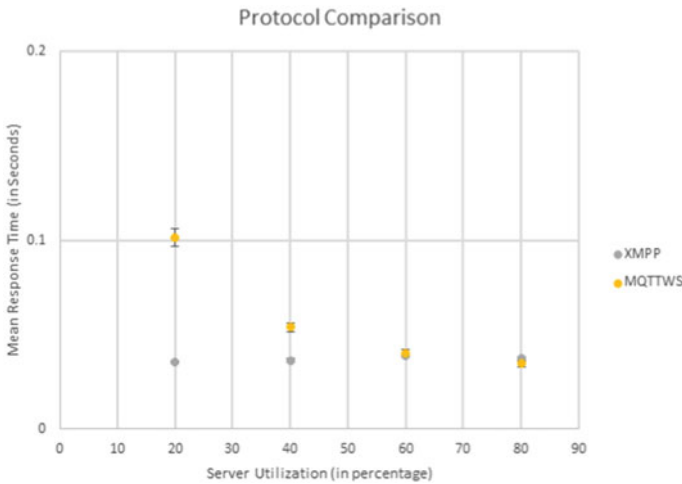


Fig. 4 Mean response time—MQTT versus XMPP

As brokers get requests from subscribers, it ensures that the subscriber is authorized to receive data using identifying valid subscriber id, username, and password [1, 11, 12].

2.2 Authentication

For secure authentication along with username and password, MQTT ensures authentication along with a X.509 certificate. Certificate X.509 is a digital certificate which works on public key infrastructure to ensure that public key must belong to each subscriber [13].

X.509 certifies the identity of each subscriber while performing a handshake process between publisher and subscriber. During the handshaking process, the subscriber provides a digital certificate to broker with identity and public key. Brokers need to verify submitted subscriber certificates to the certification authority for further verification. After confirmation, the broker identifies the subscriber as genuine and validated subscriber within subscriber username and public key. MQTT brokers must support the utilization of X.509 certification to ensure secure communication [13, 14].

For subscriber authentication using X.509 digital certificates, certificates must be created at TLS (Transport Layer Security) using advanced encryption methods. TLS encryption ensures a secure communication channel for TCP/IP for resisted subscribers.

2.3 Authorization

Once connection gets established with a broker, connected objects can publish and subscribe to one or more topics. Topics play a key role to authorize the subscriber, else clients would be able to subscribe and publish to any topics available within a broker.

Authorization deals with RBAC (Role Based Access Control) and ACL (Access Control List). RBAC ensures the level of abstractions among subscribers and topics-based subscription [15]. ACL encompasses a list of subscribers with a list of permissions. Permissions lead to the secure policy on which topics, client can publish or subscribe [16].

MQTT broker configured topics along with permissions. Meanwhile brokers ensure the valid topics, types of operations and level of Quality of Services (QoS). If any subscriber performs unauthorized activity, the broker performs certain actions such as invalidating the digital certificate of the subscriber, so the subscriber would be unable to publish or subscribe to the given topic. For permissions, access token provided to each subscriber. By validating a token, IoT can be prevented from unauthorized access to publish or subscribe data that may have an inauspicious effect on connecting smart objects of the IoT paradigm.

Alternative to ensure authorization with a broker is to provide a third-party source to confirm authorization connected with a subscriber based on access token and unique identity. For secure communication, additional identification is provided with a username field. Access tokens can be distributed using several ways, but a widely used way is OAuth2.0 [14, 16, 17].

3 X.509 Client Certificate Authentication Within MQTT

MQTT brokers ensure the identity of each subscriber during TCL handshaking, and it can abort the handshake if the identity of the client certificate fails. Basically, the process of authenticating the secure communication between clients, publishers and subscribers.

Client certificate offers benefits such as verification of identity of subscriber, authentication of subscriber (at transport layer) and invalidate clients before MQTT CONNECT message sent. These X.509 client certificates must be implemented within MQTT protocol [12, 13].

These additional security approaches at the cost of provisioning certificates and revocation mechanisms that must be identified in communication policy.

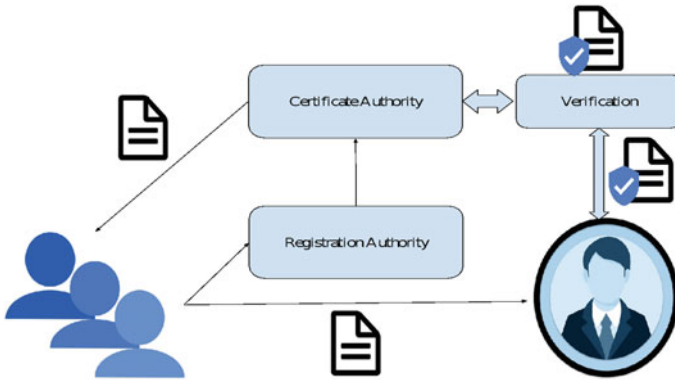


Fig. 5 Public key infrastructure

3.1 Certificate Provisioning

Public Key Infrastructure (PKI) would be the key element to ensure provisioning certificates to subscribers. Without PKI, it becomes challenging to maintain digital certificates for each client (See Fig. 5) [18].

3.2 Certification Revocation List

To ensure malicious clients from utilizing fake client certificates, brokers must identify invalid certificates and prevent clients that request to establish connection. To implement this, CRLs are used. CRL is a list of all invalidate certificate repositories (See Fig. 6) [19]

Fig. 6 Certification revocation list



```

@Override public CRList<iPermission> getPermissionsForClient(final ClientData sub) {
    final CRList<iPermission> permission = new ArrayList<iPermission>();
    final X509Certificate mycertificate = (X509Certificate) sub.getCertificate().get();
    if ("DhavalNimavat".equals(certificate.getIssuerDN().getName())) {
        log.info("Broker is DhavalNimavat, you can utilize all topics along with all
        permissions");
        // # --> identifies all permission will be given to client
        permission.add(new Permission("#", iPermission.ALLOWED_QOS.ALL));
    }
    return permission;
}

```

Fig. 7 Code depicts how permission can be added to certificate

3.3 X.509 Certificate for Authentication and Authorization

Using this digital certificate, we can ensure authentication and authorization both. The below code of fragment depicts how permission can be added based on certificate (See Fig. 7) [20].

To manage challenges of certificate lifecycle management, within MQTT protocol X.509 client certificates would be more useful. To ensure the secure communication, another layer of security is much needed within MQTT such as application-based authentication and client certificate authentication [12, 13, 21].

3.4 OAuth2.0 Within MQTT

In IoT, to add more layers of security, OAuth 2.0 would play a key role within MQTT protocol. OAuth 2.0 is all about an authorization framework which enables subscribers to access resources that are subscribed by a client without disclosing their unencrypted credentials. For example, currently, various mobile applications are accessing Google profiles for sign up and sign in process. Client does not need to provide Google password to each application, rather than login via Google and authorized applications will access Google information on client behalf [17, 22].

Key elements of OAuth 2.0 are client, resource owner, resource server and authorization server. Client can be defined as an application that requests access to resources from publishers. Resource owner is referred to as a publisher who owns or is responsible to publish data over TCL. Resource server supplies the resources that are required to be protected and can only be accessed by a valid user. Authorization server maintains and grants access to available resources. The authorization server and resource server must subsidize one another to ensure security [23, 24] (See Fig. 8).

OAuth2.0 commonly works with JSON (JavaScript Object Notation) Web Token (JWT), identifies and communicates by base64 encoding standards. JWT token consists of header, pay and signature. Header contains details about which cryptographic algorithm is used for encryption and to generate signatures. Payload contains information such as publisher, publish date, expiry time, subscriber and topics.

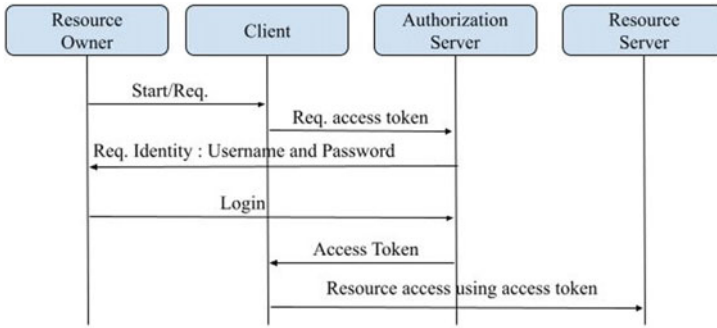


Fig. 8 OAuth 2.0—authorization process

Payload information can be customized or modified as per publisher need. Each JWT token is signed by the authorization server, so the resource server would be able to ensure the publisher is trusted [23, 25].

Future JWT tokens can be classified as refresh tokens and access tokens. Access token controls the permission on specific resources. But as it is a short-living token, it expires within one day. Where refresh tokens have longer life compared to access token, which request for fresh access token from the server as previous token gets expired.

OAuth 2.0 is an open source protocol which enables secure authorization using standard cryptography methods. It is also known as a framework that empowers third party applications to access assigned resources within the IoT environment.

Additionally, MQTT broker also need to validate using access tokens, to authorize the clients. Client Authorization leads to two different scenarios such as authorization within topic subscription and third party authorization [26, 27].

4 Payload Encryption

Payload encryption adds another layer of security within a trust-less IoT environment. Payload encryption is also defined as a cryptography method used to encrypt topic specific data at application level. This method supports end- to-end encryption between publisher, subscriber and MQTT broker. During transmission, metadata remains intact while only the payload message is being encrypted. Generally MQTT payload encryption is applicable to MQTT publisher packets only, which ensures that no client side mechanism is required at the broker end to decrypt data [24].

- Payload encryptions generally focus on
- PUBLISH topics
- CONNECT username and password
- SUBSCRIBER topics

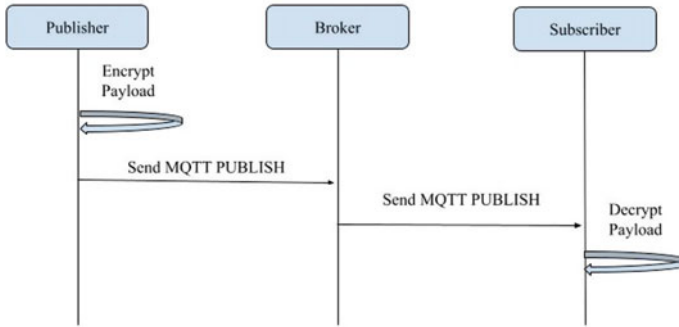


Fig. 9 End-to-end encryption process

- UNSUBSCRIBED topics

4.1 E2E (End-To-End)Encryption

MQTT broker works with unencrypted metadata packets for quality service handling and routing, only application or topic based data gets encrypted and even broker can not decrypt the encrypted topic based packet data.

So even an unauthorized client that gets access to an MQTT packet can not decrypt the information without having a public/private key. E2E encryption is not dependent on the MQTT broker, the publisher can apply encryption to any topic as per need [24] (See Fig. 9).

4.2 Client to Broker Encryption

In client to broker encryption paradigm encrypted payload of packet is decrypted by broker before publishing, as a result all subscribers would receive unencrypted information. In this method, the broker decrypts the message on the fly (while transmitting the message) (See Fig. 10).

Payload information ensures end-to-end encryption and adds another layer of security for specific topics within push protocol architecture [24].

5 Conclusion

In growing IoT fields, security always remains challenging for effective and secure communication among their smart objects and its network. To ensure effective

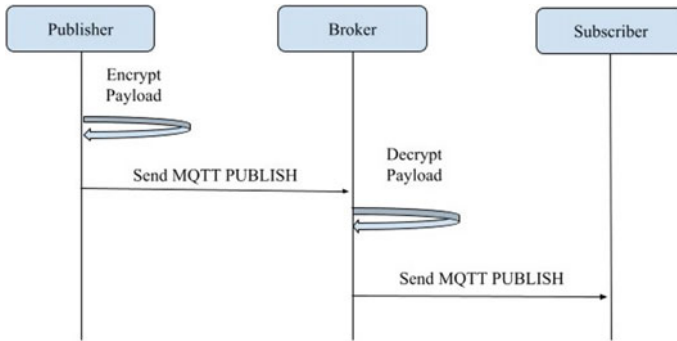


Fig. 10 Client to broker encryption process

communication and to achieve security within IoT architecture, it always depends upon protocols used for transmitting and receiving data over TCL. Various IoT push protocols such as XMPP, MQTT, AMQP ensure the communication at TCL among which MQTT protocol is widely used protocol within IoT platform because it requires nominal resources as it's lightweight and efficient. This survey provides a contribution to how MQTT protocol can be strengthened by approaching advanced security concepts such as client certification authentication using X509, authorization by using OAuth 2.0 paradigm and lastly payload encryption methods.

References

1. Kawaguchi R, Bandai M (2020) Edge based MQTT broker architecture for geographical IoT applications. In: 2020 International conference on information networking (ICOIN), IEEE
2. Gruener S, Koziolok H, Rückert J (2021) Towards resilient IoT messaging: an experience report analyzing MQTT brokers. In: 2021 IEEE 18th international conference on software architecture (ICSA), IEEE
3. Bellavista P, Foschini L, Ghiselli N, Reale A (2019) MQTT-based middleware for container support in fog computing environments. In: 2019 IEEE symposium on computers and communications (ISCC), IEEE
4. Park, C-S, Nam H-M (2020) Security architecture and protocols for secure MQTT-SN. IEEE 8
5. Hunkeler U, Truong HL, Stanford-Clark A (2008) MQTT-S—a publish/subscribe protocol for Wireless Sensor Networks. In: 2008 3rd international conference on communication systems software and middleware and workshops (COMSWARE '08), pp 791–798. <https://doi.org/10.1109/COMSWA.2008.4554519>
6. Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M (2015) Internet of things: a survey on enabling technologies, protocols, and applications. *IEEE Commun Surv & Tutor* 17(4):2347–2376, Fourthquarter 2015. <https://doi.org/10.1109/COMST.2015.2444095>
7. Naik N (2017) Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. *IEEE Int Syst Eng Symp (ISSE) 2017*:1–7. <https://doi.org/10.1109/SysEng.2017.8088251>

8. Kayal P, Perros H (2017) A comparison of IoT application layer protocols through a smart parking implementation. In: 2017 20th conference on innovations in clouds, internet and networks (ICIN), pp 331–336. <https://doi.org/10.1109/ICIN.2017.7899436>
9. Karagiannis V, Chatzimisios P, Vazquez-Gallego F, Alonso-Zarate J (2015) A survey on application layer protocols for the internet of things. *Trans IoT Cloud Comput* 3:11–17
10. Sadio O, Ngom I, Lishou C (2019) Lightweight security scheme for MQTT/MQTT-SN protocol. In: 2019 sixth international conference on internet of things: systems, management and security (IOTSMS), pp 119–123. <https://doi.org/10.1109/IOTSMS48152.2019.8939177>
11. Shi H, Niu L, Sun J (2020) Construction of industrial internet of things based on MQTT and OPC UA protocols. In: 2020 IEEE international conference on artificial intelligence and computer applications (ICAICA), IEEE
12. Longo E, Redondi AEC, Cesana M, Arcia-Moret A, Manzon P (2020) MQTT-ST: a spanning tree protocol for distributed MQTT broker. In: ICC 2020—2020 IEEE international conference on communications (ICC), IEEE
13. Vukasović M, Veselinović B, Stanisavljević Ž (2017) A development of a configurable system for handling X509 certificates. In: 2017 25th telecommunication forum (TELFOR), IEEE
14. Oh S-R, Kim Y-G (2019) Interoperable OAuth 2.0 Framework. In: 2019 international conference on platform technology and service (PlatCon), IEEE
15. Eason G, Noble B, Sneddon IN (1955) On certain integrals of Lipschitz-Hankel type involving products of Bessel functions. *Phil Trans Roy Soc London A247*, 529–551
16. Li X, Xu J, Zhang Z, Lan X, Wang Y (2020) Modular security analysis of OAuth 2.0 in the three-party setting. In: 2020 IEEE European symposium on security and privacy (EuroS&P), IEEE
17. Nicole R (in press) Title of paper with only first word capitalized. *J Name Stand Abbrev*
18. Laurent-Maknavičius, M (2007) A PKI approach targeting the provision of a minimum security level within internet, *Fourth European Conference on Universal Multiservice Networks (ECUMN'07)*, Toulouse, France, pp. 433–438, <https://doi.org/10.1109/ECUMN.2007.3>
19. Rigazzi G, Tassi A, Piechocki RJ, Tryfonas T and Nix A (2017) Optimized certificate revocation list distribution for secure V2X communications, IEEE 86th vehicular technology conference (VTC-Fall), Toronto, ON, Canada, pp. 1–7. <https://doi.org/10.1109/VTCFall.2017.8288287>
20. Alrawais A, Alhothaily A, Cheng X (2015) X.509 check: a tool to check the safety and security of digital certificates. In: 2015 international conference on identification, information, and knowledge in the internet of things (IIKI), pp 130–133. <https://doi.org/10.1109/IIKI>
21. Triartono Z, Negara RM, Sussi (2019) Implementation of role-based access control on OAuth 2.0 as authentication and authorization system. In: 2019 6th international conference on electrical engineering, computer science and informatics (EECSI), pp 259–263. <https://doi.org/10.23919/EECSI48112.2019.8977061>
22. Clerk Maxwell J (1892) A treatise on electricity and magnetism, 3rd ed, vol 2. Clarendon, Oxford, 68–73
23. Jacobs IS, Bean CP (1963) Fine particles, thin films and exchange anisotropy. In: Rado GT, Suhl H (eds) *Magnetism*, vol III. Academic, New York, pp 271–350
24. Manimegalai R, Priyadharshini A (2018) Privacy preserving public auditing with data storage security in cloud computing: an overview. *Int J Comput Sci Eng* 6(5), 532–534
25. Oh S, Kim Y (2019) Interoperable OAuth 2.0 Framework. In: 2019 international conference on platform technology and service (PlatCon), pp 1–5. <https://doi.org/10.1109/PlatCon.2019.8668962>
26. Vachhani SK, Nimavat D, Kalyani FK, A comparative analysis of different algorithms used in IOT based smart car parking systems. In: 2018 international research journal of engineering and technology (IRJET), 3244–3248
27. Collina M, Corazza GE, Vanelli-Coralli A (2012) Introducing the QUEST broker: scaling the IoT by bridging MQTT and REST. In: 2012 IEEE 23rd international symposium on personal, indoor and mobile radio communications—(PIMRC), 2012, pp 36–41. <https://doi.org/10.1109/PIMRC.2012.6362813>

Comparative Analysis of Different Block Chain Technology to Improve the Security in Social Network



Niki Modi

Abstract Social networking sites have given users unprecedented opportunities for the generation and dissemination of content. Block chain Technology as defined the decentralized system for distributed registers which are used to record data transactions on multiple computers. So a variety of social networking sites exist for different purposes, to afford users a range of anonymous and non-anonymous options for self-expression, and the ability to be a part of a virtual community. Sometimes a misinformation, propagated by users and group can create chaos or in some cases, might leads to cases of riots. Therefore, a robust and new system is required to check the information authenticity within the network, to stop the propagation of misinformation. In this paper, propose of block chain based framework is for sharing the information securely at the peer level. In the block chain model, a chain is created by combining blocks of information. I analyze real data by exploiting one of the most well-known DApps sites (decentralized applications), and also compare current technologies in order to get better algorithm or tool to secure our information. such as Facebook.

Keywords Decentralized applications · Block chain · Facebook · Dissemination · Networking · Authenticity

1 Introduction

Sharing is a fundamental human experience, and the rise and spread of social networking sites (SNSs) has served to open unprecedented avenues for achieving this experience. Recent statistics show that 30% of all time spent online is on social media, with teens leading the pack by spending an estimated nine hours online each day. Content on social media has often been satirized for the breadth of revealed content, ranging from everyday life occurrences such as meals, fitness, personal thoughts and family stories, workplace milestones and challenges, to broader calls

N. Modi (✉)

Princeton University, Princeton, NJ 08544, USA

e-mail: lncs@springer.com

Springer Heidelberg, Tiergartenstr. 17, 69121 Heidelberg, Germany

for prayers, funds, recommendations, and sharing of news [1]. An incorrect information may be termed as rumor related to public interest. Nowadays social network, like Facebook, Twitter is very common for communication among people to do collaborative action [2, 3].

Therefore, the relation between the dynamics of information and the structure of the underlying network is crucial in many real cases, e.g., the spreading of worms in a computer network (e.g., ransom virus on technological networks), [4] viruses in a human population (e.g., zika virus spreading in human), information propagation in the online social network.

The block chain technology has developed for the financial transaction of bit coin with trusted and secured contract between two communicating parties at the peer level. Also new platforms were born to implement various types of business models, called previously decentralized online social networks (DOSNs) [5] because they were based on P2P networks. There are another method to solve social network issue there is emergence of distributed OSNs (DOSNs) can solve this privacy issue, yet they bring inefficiencies in providing the main functionalities, such as access control and data availability. There is secure hash algorithm. So based on each technique will compare and analyze the better way to secure our information while using social network.

2 Literature Review

Block chains have shown tremendous potential to transform the user experience in governance, finance, and health informatics [2]. Therefore, intensive research is required for information dynamics in which, correct information should be propagated as well as misinformation or false information should be blocked to stop the chaos [6].

Therefore, a new technique or method is required to solve the problem of information dynamics considering verification and authentication of information, near to the initial period of the starting of the information, in the social network so that immediate action should be taken to remove the unverified information. In addition, each user in a social network creates trust with its neighbors at peer level before sharing the information. Trust is a measure of confidence in social networks and it provides the information about the neighbors with whom, what type of information one share/accept with others [7].

Block chain is based on a peer-to-peer architecture, guaranteeing that data is duplicated and distributed to all the nodes of the network. This way, information is made practically unassailable, being no longer in the hands of a single operator but being duplicated and spread among all the participants to the network. The records are stored in a linear chain. Pointers and linked list data structures are used in block chain for the block representation. Blocks are arranged in sequence and lined with each other, using a linked list. Pointers are used to point the location of the next block [8].

A block is a collection of data that stores transaction details, such as the timestamp and link to the previous block, which is generated by a secure hash algorithm. Many existing works on DOSNs have focused on solving the problems of storage, access control, and providing services [9].

3 Proposed Methodology

First, here they describe the network model as a decentralized network and evaluate the trust value by considering network parameter like the degree of a node of a given network. Trust in the network may be defined as the agreement to believe that some node (user in a social network) is good and honest and will not harm you, or that something is safe and reliable. Based on this trust, we define the credibility score of each node according to the message type. Credibility may be called as the fact that someone can be believed or trusted.

The aim of introducing trust among the nodes is to find the suitable user to validate or invalidate the information in order to propagate the information. Each user, will keep track of a trust value with each of its neighboring users, Trust may be considered in two ways, one is private trust (or local trust) between two communicating nodes and, the other is the public trust (or global trust), in which a source node broadcasts the information in the network about the type of information.

Another Methodology is for the BEV-SNS model. User activities in SNSs are stored in the block chain, along with queries for the data that are generated by system application programming interfaces (APIs). The block chain stores information about user content, preferences for sharing rewards for sharing content, and records about data access. This framework for content generation and by enabling secure transaction processing and record keeping as given in below Figs. 1 and 2. There are two components to this framework: the user data, and the enhanced block chain-based digital ledgers that contain algorithms for selecting sharing and reward-generation mechanisms.

4 Security Process Using SNS Model Dapps, & BCOSN's Architecture

Trust, the inherent architecture of block chains ensures that the user can tweak the parameters of sharing and rewards in BEV-SNS frameworks for a secure, trusted, and rewarding networking experience. To illustrate this framework, we provided examples of its use with SNSs that lie along the spectrum of anonymity and showed that the framework could be scaled for future use in a variety of collocated spaces.

We elaborated the data collected in order to better understand the status of current DApps and compared them with the data collected in 2019 from the same site, Fig. 3

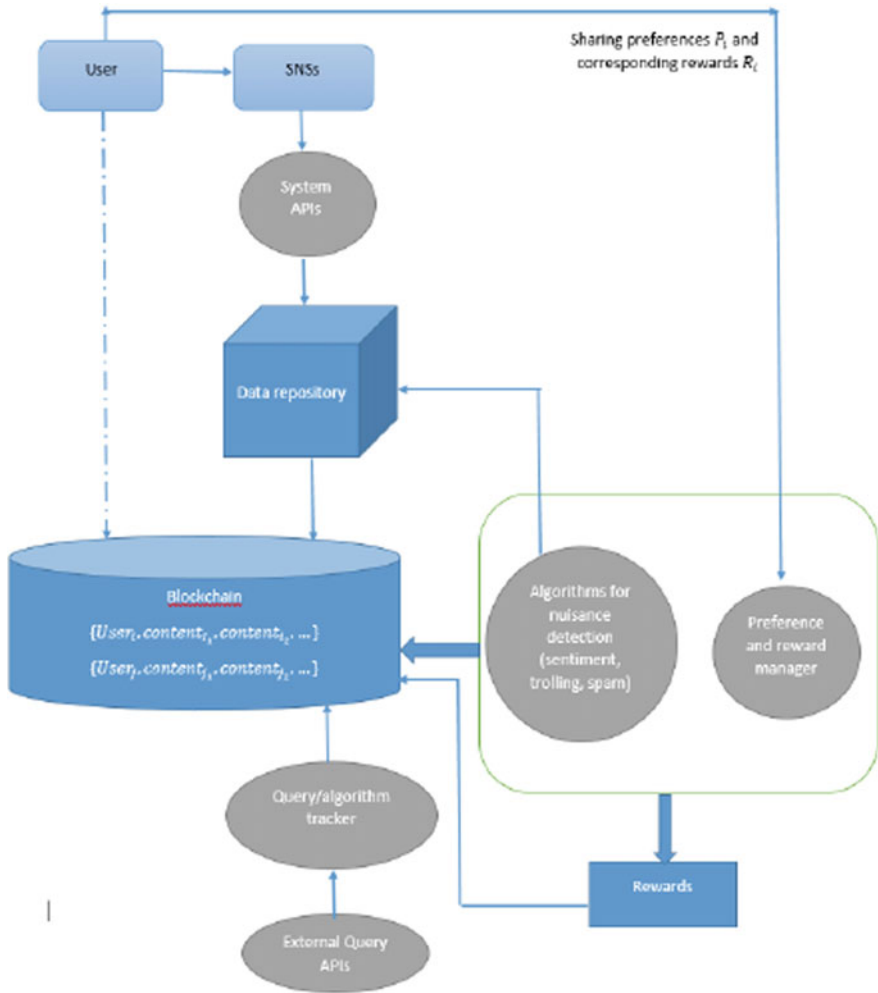


Fig. 1 System architecture and data model for BEV-SNS

shows the status of the DApps by grouping them into the categories in which the DApps is built in 2019. As we expected, there are several DApps concerning the financial and game DApps; as concerns the social scenario, social DApps are within the top 5 DApps categories, which describes the importance of DApps in the social environment. The trend of the DApps is increasing year by year, as we can see in diagram social DApps have increased from about 250 proposals in 2019 to about 300 proposals in both 2020 and 2021. This means that the scenario has big interest and potential by considering the new block chain proposed. Indeed, the scenario concerning the block chain technology applied to social DApps have completely changed from 2019 to 2020 and 2021, as we can see in diagram Figs. 4, 5 and 6.

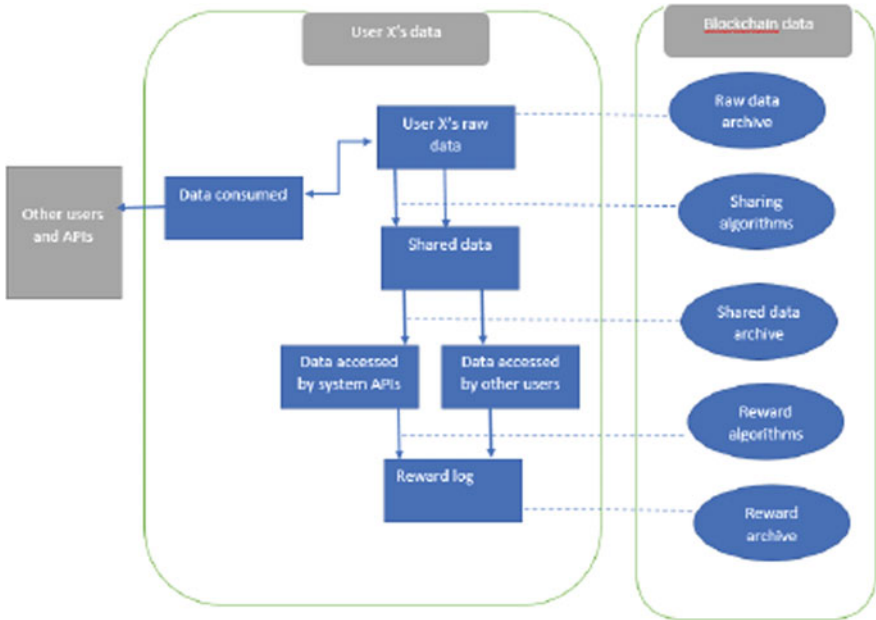


Fig. 2 System model and data flow

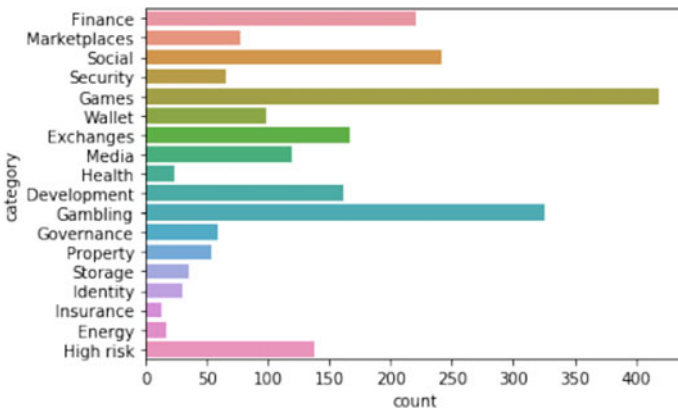


Fig. 3 Status of DApps (information collected from the site www.stateofthedapp.com the 14th of May, 2021). a List of DApps (2019)

Also analyze the top 5 blockchains, considering Fig. 6 in the social category, in order to understand the characteristics of the blockchain technology tailored in the social scenario.

There are a number of social DApps built on top of the most used blockchain, such as Ethereum, EOS, Steem, and so on. Some of them are always active, which

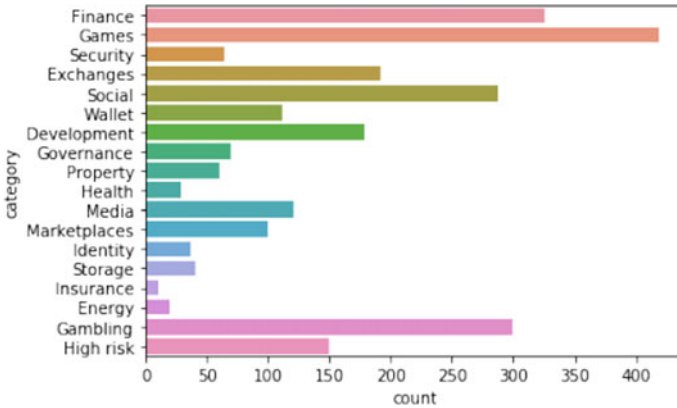


Fig. 4 List of DApps (2020)

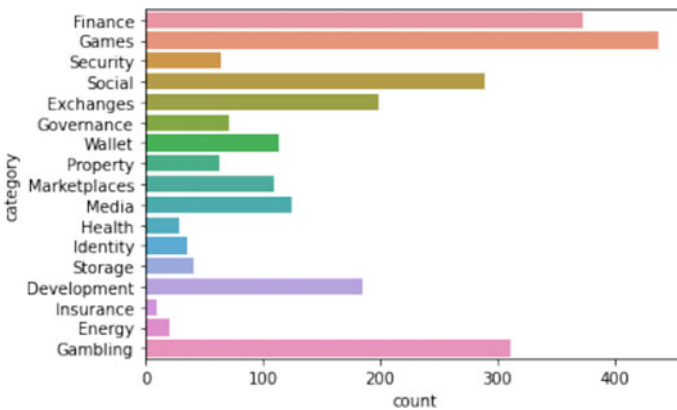


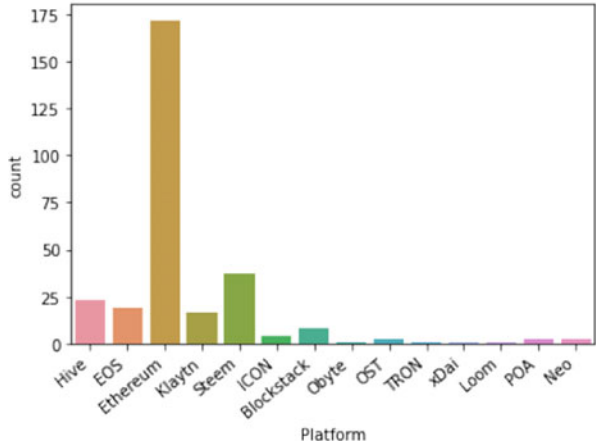
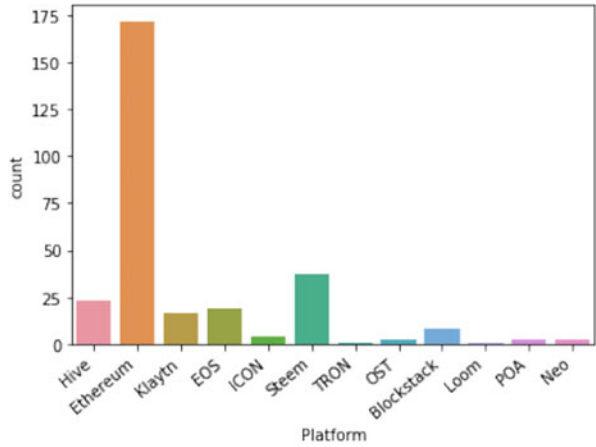
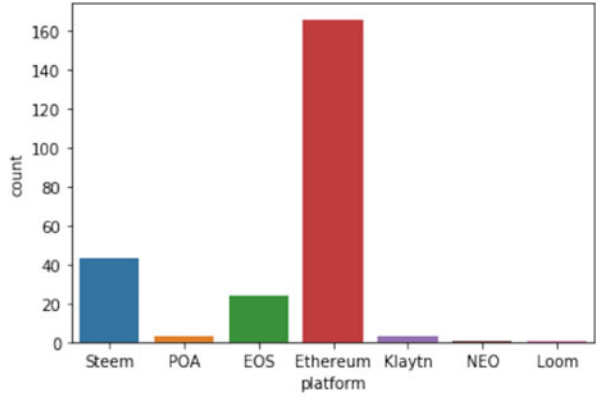
Fig. 5 List of DApps (2021)

means that there are, at the time of writing, new transactions corresponding to those DApps, while some of them have seemingly been dead for a long time.

In this methods, described the most used blockchain by analyzing the list of the DApps from the last two years, collected from the site www.stateofthedapps.com (accessed on 1 September 2021). We found that almost all the social DApps are based on Ethereum. Other important block chains are Steem and Hive, which are born to be social; for this reason, they provide a set of social features that are not provided by other block chains. The other block chains used in this scenario are EOSIO and Klaytn, even if they are not so well investigated in the social environment. As suggested in this work, scalability and transaction fees are the most important points in choosing a block chain.

Here the designed system is a block chain-based framework for decentralized OSN termed BCOSN as shown in above Fig. 7. Combining with smart contracts,

Fig. 6 Block chain used by DApps considering the social category. (information collected the 14th of May, 2021) **a** Overview of the Social DApps (2019); **b** Overview of the Social DApps (2020); **c** Overview of the Social DApps (2021)



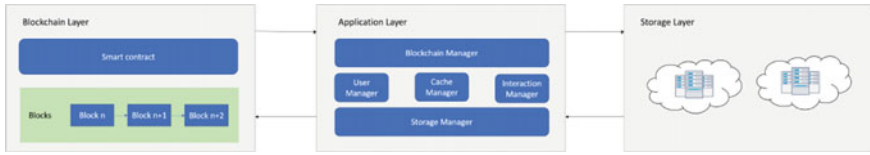


Fig. 7 Overview of the BCOSN’s architecture

also taken the block chain as a trusted server to implement the functionalities that are used to be provided by central servers in traditional OSNs. Compared to the existing DOSNs, the BCOSN can provide efficient, safe, and privacy-aware functionalities of authentication, newsfeed notification, and friend recommendation. Meanwhile, it also provided users with fine-grained encryption to protect data privacy. A series of algorithms has been designed based on smart contracts to construct a concrete scheme under the framework of the BCOSN. In addition, the experimental results have shown the effectiveness of the BCOSN.

5 Comparison of Different Security Features in Blockchain for Social Network

Feature	Peer level algorithm	BEV-SNS algorithm	BCOSN algorithm	Decentralized based application
Third party Authentication/information	We no longer require a third party for information Verification	Control over data access	Fine-grained encryption to protect data privacy	Promotes user privacy
Accuracy	83%	Less than peer level algorithm 70%	Accuracy is more as compare to other two	More than 85%
Efficient & flexible	This gives the user a very high level of protection against being compelled to disclose its contents	Efficient task execution. & ML algorithms to better understand sharing preferences and reward mechanisms on various kinds of SNSs	The experimental results shows the effectiveness of the proposed BCOSN and verify that the friend recommendation can be implemented	An app requires significant computations and overloads a network, causing network congestion

Also We can compare block chain social media versus Popular social media network.

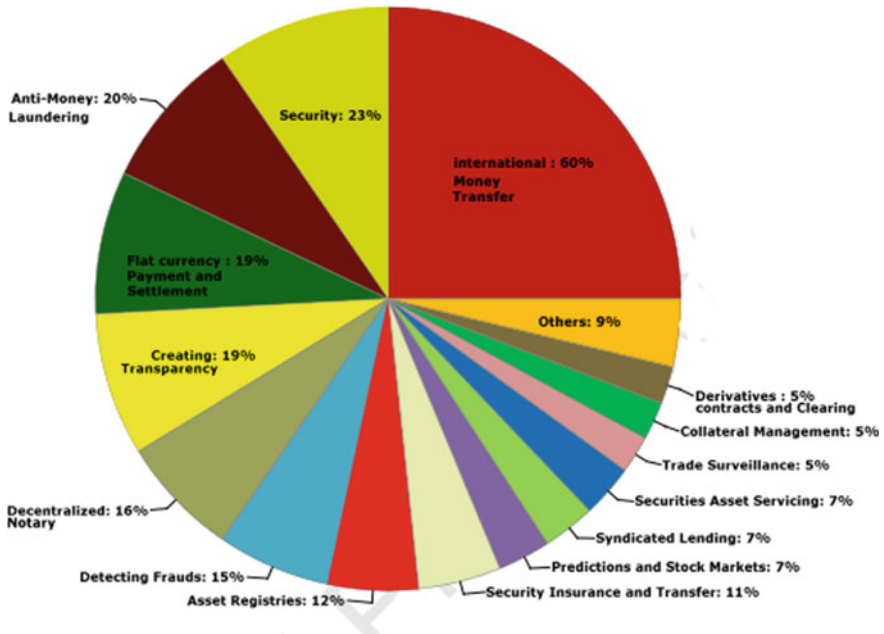


Fig. 8 Various financial applications of block chain across world in 2016

- The app publishes real videos and events, and other users upvote the content to receive ‘Karma’. Earn enough Karma points, and it gets converted into real money which you can withdraw at the end of the day.
- On the other hand, we have Instagram, where you can share a photo or a video with other users. It’s like the simplified version of Facebook, with a strong emphasis on visual sharing. If you are a business owner, you can post your product images on Instagram or maybe use an Instagram influencer to give a shout-out to your business. These efforts may direct traffic to your site but the post itself won’t generate any money. The Karma app, however, can help you make money with the post itself.

Therefore, the Karma app comes with its own secure wallet, direct messaging, and charity integration. You can buy stuff from other Karma users directly, unlike Instagram, which is yet to embrace the in-app shopping feature also given in terms of pie chart in below diagrams Figs. 8 and 9.

6 Conclusion

The main purpose of our paper is to secure social network using block chain technology, Application, algorithm and improve security based on authorization, token id,

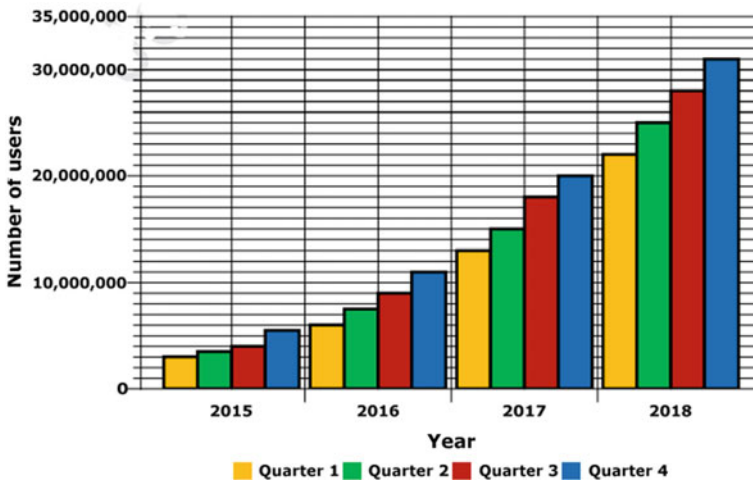


Fig. 9 Statistics in terms of increase in number of users of blockchain wallet

session id. The traditional method of fake information detection is unable to find the source of the message generator in the social network. We have simulated the information propagation using the block chain protocol, also The design of ML algorithms to better understand sharing preferences and reward mechanisms on various kinds of SNSs are areas of potential research. In this paper, we introduced BOSMs from the technical point of view by taking into account the issues concerning current block chains and determining which properties are the most important ones for choosing a suitable block chain. In this regard, one possible solution in the future is to store user data in the block chain anonymously rather than pseudonymously.

Blockchain-based social media represent a good alternative to current OSNs. Users gain full control of their content and are rewarded in order to encourage engagement, participation, and, in particular, the production of valuable content [5].

References

1. Guidi B, Conti M, Passarella A, Ricci L (2018) Managing social contents in decentralized online social networks: a survey. *Online Soc Netw Media* 7:12–29
2. Guidi B (2020) When blockchain meets online social networks. *Pervasive Mob Comput* 62:101131
3. KONECT (2017) Facebook wall posts network dataset [Online]. <http://konect.uni-koblenz.de/networks/facebook-wosn-wall>
4. Doerr B, Fouz M, Friedrich T (2012) Why rumors spread so quickly in social networks. *Commun ACM* 55(6):70–75
5. Datta A, Buchegger S, Vu LH, Strufe T, Rzacca K (2010) Decentralized online social networks. In: *Handbook of social network technologies and applications*; Springer, Boston, MA, USA, pp 349–378

6. Gujarala S, White J, Hudson B, Matthews J (2015) Fake twitter accounts: profile characteristics obtained using an activity-based pattern detection approach. In: Proceedings of the 2015 international conference on social media and society. <https://doi.org/10.1145/2789187.2789206>
7. Walter FE, Battiston S, Schweitzer F (2008) A model of a trust-based recommendation system on a social network. *Auton Agent Multi-Agent Syst* 16(1):57–74
8. Yaga D, Mell P, Roby N, Scarfone K (2019) Blockchain technology overview. [arXiv:1906.11078](https://arxiv.org/abs/1906.11078)
9. Sharma R, Datta A (2012) SuperNova: super-peers based architecture for decentralized online social networks. *Proc IEEE Int Conf Commun Syst Netw* 1–10
10. Jin F, Dougherty E, Saraf P, Cao Y, Ramakrishnan N+ (2013) Epidemiological modeling of news and rumors on twitter. In: Proceedings of the 7th workshop on social network mining and analysis. ACM, p 8
11. Chen S, Shi R, Ren Z, Yan J, Shi Y, Zhang J (2017) A block chain based supply chain quality management framework. In: e-business engineering (ICEBE), 2017 IEEE 14th international conference on. IEEE, pp 172–176
12. Adikari S, Dutta K (2014) Identifying fake profiles in linkedIn. In: PACIS 2014 Proceedings, vol 278. <https://aisel.laisnet.org/pacis2014/278>
13. Yan Z, Feng W, Wang P (2015) Anonymous authentication for trustworthy pervasive social networking. *IEEE Trans Comput Social Syst* 2(3):88–98
14. De Salve A, Mori P, Ricci L (2018) A survey on privacy in decentralized online social networks. *Comput Sci Rev* 27:154–176
15. De Salve A, Mori P, Ricci L (2015) A privacy-aware framework for decentralized online social networks. In: Proceedings of the international conference on database and expert systems applications. Springer, Cham, Switzerland, pp 479–490
16. Li C, Palanisamy B (2019) Incentivized blockchain-based social media platforms: a case study of steemit. In: Proceedings of the 10th ACM conference on web science, Boston, MA, USA, 30 June–3 July 2019, pp 145–154
17. Steemit Inc (2018) Steem. An incentivized, blockchain-based, public content platform
18. Yu R et al (2017) Authentication with block-chain algorithm and text encryption protocol in calculation of social network. *IEEE Access* 5:24944–24951
19. Klukovich E, Erdin E, Gunes MH (2016) POSN: a privacy preserving decentralized social network app for mobile devices. In: Proceedings of the international conference on advances in social networks analysis and mining, pp 1426–1429
20. Buchegger S, Schiöberg D, Vu L-H, Datta A, “PeerSoN: P2P
21. Social networking: Early experiences and insights (2009). *Proc ACM Eur Conf Comput Syst Workshop Social Netw Syst* 46–52
22. Ba, C.T.; Zignani, M.; Gaito, S. Social and Rewarding Microscopical Dynamics in Blockchain-Based Online Social Networks. In Proceedings of the Conference on Information Technology for Social Good, GoodIT’21, Rome, Italy, 9–11 September 2021; pp. 127–132.
23. Guidi B, Michienzi A, Ricci L (2020) A graph-based socioeconomic analysis of steemit. *IEEE Trans Comput Soc Syst* 8:365–376
24. Guidi B, Michienzi A, Ricci L (2020) Steem blockchain: mining the inner structure of the graph. *IEEE Access* 8:210251–210266
25. Jiang L, Zhang X (2019) BCOSN: a blockchain-based decentralized online social network. *IEEE Trans Comput Soc Syst* 6:1454–1466
26. How Trump consultants exploited the facebook data of millions. Accessed 5 May 2018. <https://www.nytimes.com/2018/03/17/us/politics/cambridge-analytica-trump-campaign.html>
27. Rathore S, Loia V, Park JH (2018) SpamSpotter: an efficient spammer detection framework based on intelligent decision support system on Facebook. *Appl Soft Comput* 67:920–932
28. Gervais A, Karame GO, Wüst K, Glykantzis V, Ritzdorf H, Capkun S (2016) On the security and performance of proof of work blockchains. In: The ACM SIGSAC conference on computer and communications security, pp 3–16

29. Kapanova K, Guidi B, Michienzi A, Koidl K (2020) Evaluating posts on the Steemit Blockchain: analysis on topics based on textual cues. In: Proceedings of the 6th EAI international conference on smart objects and technologies for social good, Antwerp, Belgium, 14–16 September 2020, pp 163–168

Euphonia: Music Recommendation System Based on Facial Recognition and Emotion Detection



Eliganti Ramalakshmi, Huma Hussain, and Kritika Agarwal

Abstract Emotions can be challenging to describe and interpret, which is why music has been proposed as an art. In recent times, music can be used as a mood regulation mode, to assist someone balance, understand and deal with their emotions better. 'Euphonia' is intended at easing that process. The purpose of 'Euphonia' is to use real-time facial recognition to acquaint the machine with abilities to recognize and examine human emotions. With this, the machine will be trained to provide the user with suitable songs for that particular mood. Besides this, the machine will also recommend the user with a general playlist pertaining to the user's likes and dislikes which they can access whenever they wish to. Machine learning concepts and the available datasets have been utilized to classify a vast set of music that is stored using automatic music content analyses. It was implemented using Python, Pandas, OpenCV, and NumPy.

Keywords Music · Emotions · Classification · Computer Vision · Facial expression · Recognition

1 Introduction

Music is the art of arranging different types of sounds using various resources to create soulful and peace-giving melodies. It is one of the universal cultural aspects of all human societies and is known for altering the listener's emotion. An emotion is a complex psychological state that describes what a person feels at all points in their lifetime. In common words, it is a feeling originated from one's circumstances. Positive emotions dominate musical experiences. Pleasurable music may result in the discharge of neurotransmitters that release happiness-inducing hormones like dopamine. When one listens to music, their mood is bound to be altered and stress is relieved. Mental health, though has always been important, has received a vast amount of focus in the last few years. Due to this, more and more individuals make

E. Ramalakshmi (✉) · H. Hussain · K. Agarwal
Department of Information Technology, Chaitanya Bharathi Institute of Technology, Hyderabad, India
e-mail: eramya2@gmail.com

extra efforts to take care of their mental health. We were pushed to create a project like this since music can act as a catalyst to make one's mood better. One needs to know how they feel and know when they need to calm down and relax. This project aims at that. One can see an exponential change in their mental health when good music is involved. On a wider scale, this reduces the additional time and labor normally required to manually do this. People often specify their moods, especially with facial expressions. Music has continually been recognized to modify the temper of a human. Encapsulating and spotting the emotion portrayed by the person and suggesting suitable songs suiting one's temper may calm the thoughts of a person and result in a pleasant and enhanced experience. This project pursues the emotion expressed via their facial expressions. Facial and emotion recognition technology, which has widely gained traction due to its colossal application value and potential, has been implemented in this project (Fig. 1).

By using libraries in Python like TensorFlow, OpenCV, NumPy, and Panda, the music system is trained to retrieve and identify one's mood through the system's web camera. The program captures a snapshot of the user. Following this alongside the assistance of image classification, it extracts several features from the user's face and attempts to discover the emotion that the person is undergoing. In various cases, temper alteration may additionally assist in battling conditions like melancholy and sorrow. Using the resource of expression evaluation, many psychiatric dangers may be prevented, and additionally, there can be attempts that can assist the user to bring their temper to a healthy and better stage. This project aims at lightening the mood of the user. When it comes to your mental health, music can help you rest better, lift your mood, reduce stress, and help you comprehend your emotions better. It helps one tackle anxiety and it can act as a coping mechanism.

Fig. 1 Music's impact on human psychology



2 Literature Survey

Artificial intelligence is a prominent, and imperative domain that has taken over the world in recent times. As an apparent result, several music recommendation systems are being built and a few of them have become extremely popular. These systems take various factors into account for providing the best recommendations to the users. Some of them use the most frequently listened to songs as the basis of recommendation while others make use of factors like the user's favorite artist or their preferred genre. Many existing systems can recognize emotions through facial analysis. On the other hand, numerous existing systems recommend music.

One of the first proposals of the venue-recommender system is by Cheng and Shen [1] who proposed a venue-recommender system, for identifying suitable songs for different venues.

S Matilda Florence [2] provided recommendations to the users by considering users' choices by proposing a music recommendation system, accordingly.

Hardik Sharma, Shelly Gupta, Yukti Sharma, Archana Purwar [3] used Russell's scale which predicts arousal and valence, rather than emotion via Multi-linear Regression, and studies the model using various data mining techniques.

Vincenzo Moscato, Antonio Picariello, and Giancarlo Sperlli [4] proposed the design and implementation of a music recommender system based on the identification of personality traits, moods, and emotions of a single user.

Wenjuan Gong and Qingshuang Yu [5] developed a music recommendation algorithm that utilized a deep learning algorithm, dance motion analysis applied quantitative measures to evaluate it.

Madhuri Athavle, Depali Mudale, Upasana Shrivastav, Megha Gupta [6] used convolution neural network for the purpose of emotion detection and PyGame and Tkinter for music recommendation system.

Yading Song, Simon Dixon, and Marcus Pearce [7] used user modelling, item profiling, and match algorithms that uncovered the two basic techniques in recommendation, that is, collaborative filtering and content-based modeling.

Vuong Khuat [8] implemented a recommendation model using collaborative filtering with an enhanced runtime by the use of a more effectual data-representation scheme and considering a partial part of the dataset.

Aldiyar Niyazov, Elena Mikhailova, Olga Egorova [9] used two approaches of building a content-based music recommender system are considered in this paper. One used acoustic features analysis and another incorporated deep learning and computer vision methods application aimed at improving the efficiency and accuracy of the recommender system.

Steve Lawrence, C. Lee Giles, Ah Chung Tsoi [10] generated a hybrid neural-network which compared favorably with other techniques when it came to facial recognition. The system combined local image sampling, a self-organizing map (SOM) neural network, and a convolutional neural network.

3 Problem Statement

To create and train an effective and accurate model that keenly classifies images, predicts the current mood of the user, and plays songs according to the emotion identified by the model. Thus, it is considered to be an image classification model.

4 Architecture

This project is aimed to design, implement, and analyze a system that would recommend music to the user on the basis of their current feelings. The suggested technique, in comparison to other algorithms utilized in earlier systems, is capable of handling large pose variations.

Large pose variations tend to wreak havoc on pre-existing algorithms. To decrease the size of the image, a standard image input format is used. Few systems identify faces first, then detect them. Other algorithms, on the other hand, rarely recognize and locate faces at the same time. Every face detection method has a set of processes in common.

The purpose is to use real-time facial recognition to acquaint the machine with abilities to recognize and examine human emotions. It is a coalesce of both music suggesting applications and emotion detecting applications that extracts the best from both worlds.

To begin, we established a response time, and then executed data dimensioning. Using data dimensions as a starting point, a few algorithms extract facial measures and then react to a specific facial region. The proposed algorithm benefits from using a static image and has a significant advantage when it comes to the problem of pose variations. The existence of identifying components such as spectacles or a beard, the quality of static photos, and unidentifiable facial gestures were the three most common issues when it came to facial emotion detection.

Through the emotion detected, the system identifies a song according to the emotion of the user. The intricacies of the face recovered from the image are used by the classifier to discover the emotion felt. It will provide easy access to any song the user wants to listen to according to his/her spirits (Fig. 2).

5 Algorithm

This project is based on the concept of Computer Vision and Emotion detection. Computer vision has been specifically used to detect the emotion a person is exhibiting. Through the emotion detected, the system identifies a song playlist according to the mood of the user. It will provide easy access to any song the user wants to listen to according to his/her mood thus decreasing the work of switching

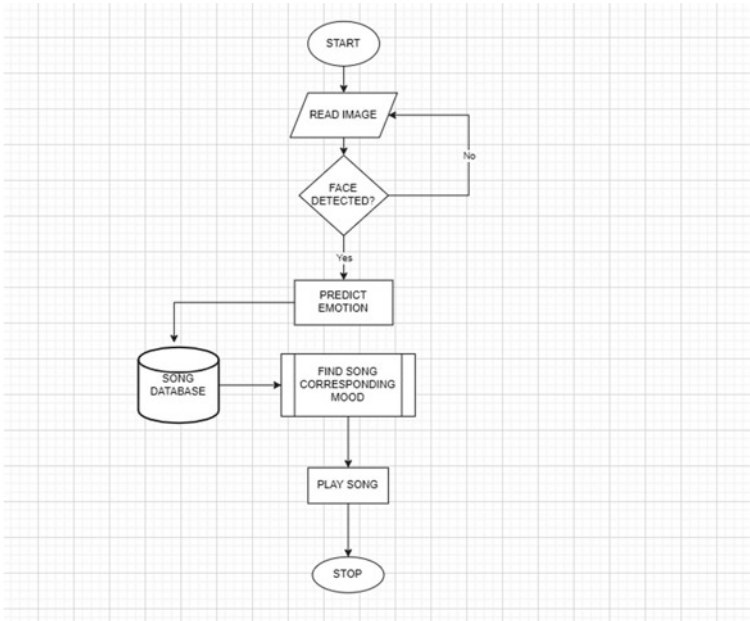


Fig. 2 Flow of the proposed system

to apps for the particular songs. This technology is used simultaneously with the causation of playing desired music based on the mood predicted.

It majorly implements a classification algorithm to classify images into their respective moods.

The proposed mechanism first distinguishes a face from a static picture by invalidating all non-required objects in the image. When the information picture is perceived and valid, the picture is further used for analysis. The picture is exposed to an image classifier to perceive the feeling displayed by the face. The feeling predicted by the emotion recognizer will in turn result in an appropriate song being played. The songs are already stored with respect to the mood they correspond to. Once the mood is reported by the model, the songs concerning the mood are played by the music player.

The project begins with a dataset comprising images for various emotions (angry, happy, neutral, or sad). Each picture has been keenly chosen which makes it complicated for the model to identify the face in the image, thus making it more efficient. The model created will be trained over these images to be able to make better predictions of one’s emotions. The dataset also comprises of a set of songs classified into each mood. These songs will be played and be most suitable when the user feels the song’s corresponding emotion. The model uses image classification to train itself with the images present in the dataset. The model can easily manage to detect the face of the person within the frame of the picture taken and resize the picture down to just the facial features of the user. It uses the graph-plotting technique and classification

to categorize these facial expressions into emotions. The project also makes use of the operating system present on the device to access the system's webcam to capture the face of the user.

6 Results and Discussion

We train the model using a special program in TensorFlow called `Retrain.py` written for accurately and efficiently training the model on the given dataset. While being trained, the command prompt returns the training accuracy, cross-entropy, and validation entropy of the model.

Next, run the `label.py` file. It detects the face using `cv2` (OpenCV) using Haar Cascade which is an ML methodology where several positive and negative images of different contexts are used to train the classifier. Then, the image of the face is provided to `label_image.py` which returns the detected emotion. It predicts and stores emotions 10 times for a single image, thus increasing accuracy.

The model checks the emotion with the highest count, it then maps the integer to the correct emotion as per the dictionary defined in the beginning.

Based on the emotion the model concludes, it checks for the resultant emotion's corresponding `.csv` file. This `.csv` file contains all the songs that are appropriate to listen to in that particular mood.

From the songs listed in the `.csv` file, the model reads a random song and provides its path to the operating system which then plays the corresponding song present in the songs folder. Thus, the song based on the person's mood is played.

The command prompt will also display necessary logs like `resume`, `pause`, `exit`, or `stop` to have better access to the song that is being played.

The figure below is the model's process to classify the image and display the key of the emotion that the model has predicted. The output below is of neutral or sad emotion (Figs. 3, 4).

When the face is analyzed and happy emotion is detected then the output below for happy emotion is shown (Figs. 5, 6).

7 Conclusion

After extensive testing and the final validation, it was concluded that the project is successful in carrying out all the functionalities mentioned throughout this report.

We were successful in proposing a music recommendation system based on user emotions. We take the human face as input and extract facial expression from it which then detects the emotion based on which music is played automatically. It suggests music by extracting different facial emotions of a person: Happy, angry, sad, or neutral. For feature extraction, we used the machine learning image classification

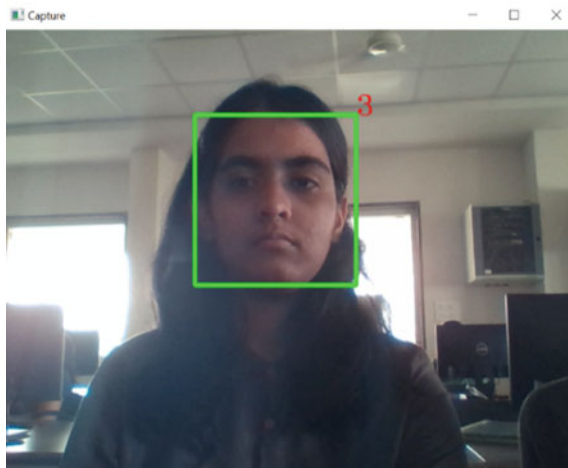


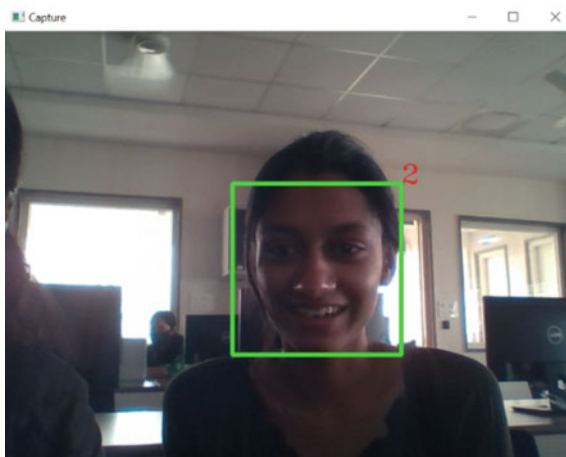
Fig. 3 Face recognition and emotion detection 1

```
Done
{'angry': '1', 'happy': '2', 'neutral or sad': '3'}
predictions = ['3', '3', '3', '3', '3', '3', '3', '3', '3', '3']
predicted freq = 3
I think you are feeling neutral or sad

Playing: Kygo_Selena_Gomez_-_It_Ain_t_Me_with_Selena_Gomez_Audio_(mp3.pm)
Actions:
  P: pause
  R: resume
  S: stop
  E: exit
You Pressed 'Stop' Key!
```

Fig. 4 The song played after detecting emotion sad

Fig. 5 Face recognition and emotion detection 2



```

Command Prompt
Done
{'angry': '1', 'happy': '2', 'neutral or sad': '3'}
predictions = ['2', '2', '2', '3', '2', '2', '2', '2', '2', '2']
predicted freq = 2
I think you are feeling happy

Playing: Ilahi
Actions:
P: pause
R: resume
S: stop
E: exit

C:\Users\Hussain\Euphonia>

```

Fig. 6 The song played after detecting emotion happy

algorithm and for model training we used OpenCV, both provided promising results with the desired accuracy. There is a scope for further upgrades and enhancements.

In the future, other emotions can also be added and our application can be extended and trained accordingly.

A graphical user interface can be made for a visually better experience for the user.

The application could allow the user to request recommendations manually and interact with the server to receive recommendations.

The application could gather information about the user and use the songs they listened to for providing better recommendations.

There could be a server-side for adding new songs to the list as and when required.

References

1. Cheng Z, Shen J (Apr. 2016) On effective location-aware music recommendation. *ACM Trans. Inf. Syst.*, 34(2), Art. no. 13. <https://doi.org/10.1145/2846092c>
2. Tambe P, Bagadia Y, Khalil T, Shaikh NU (2015) Advanced music player with integrated face recognition mechanism, undefined
3. Ayala D, Yaslan Y, Kamasak M (2018) Emotion based music recommendation system using wearable physiological sensors. *IEEE Trans Consum Electron*, vol. 64, pp. 196–203
4. S Metilda Florence and M Uma, Emotional Detection and Music Recommendation System based on User Facial Expression. <https://doi.org/10.1088/1757-899X/912/6/062007/pdf>
5. Vijay Prakash Sharma, Azeem Saleem Gaded, Deevesh Chaudhary, Sunil Kumar, Shikha Sharma, Emotion-based music recommendation system
6. Luntian Mou, Jueying Li, Ramesh Jain, Juehui Li, MemoMusic: a personalized music recommendation framework based on emotion and memory

7. Amrita Nair, Smriti Pillai, Ganga S Nair, Anjali T, Emotion based music playlist recommendation system using interactive chatbot
8. Hardik Sharma¹, Shelly Gupta, Yukti Sharma, Archana Purwar, A new model for emotion prediction in music
9. Jagendra Singh, Vijay Kumar Bohat, Neural network model for recommending music based on music genres
10. Vincenzo Moscato, Antonio Picariello, Giancarlo Sperli, An emotional recommender system for music
11. Pranesh Ulleri, Shilpa Hari Prakash, Kiran B Zenith, Gouri SNair', Jinesh, Kannimoola, Music recommendation system based on emotion
12. Wenjuan Gong, Qingshuang Yu, A deep music recommendation method based on human motion analysis
13. Liuchang Xu, Ye Zheng, Dayu Xu³, Liang Xu, Predicting the preference for sad music: The role of gender, personality, and audio features
14. Metilda Florence S, Uma M, Emotional detection and music recommendation system based on user facial expression
15. Madhuri Athavle, Depali Mudale, Upasana Shrivastav, Megha Gupta, Music recommendation based on facial emotion recognition
16. Yading Song, Simon Dixon, Marcus Pearce, A survey of music recommendation systems and future perspectives
17. Vuong Khuat, Music recommendation using collaborative filtering
18. Adiyansjaha, Alexander A S Gunawana, Derwin Suhartono, Music recommender system based on genre using convolutional recurrent neural networks
19. Aldiyar Niyazov, Elena Mikhailova, Olga Egorova, Content-based music recommendation system
20. Ferdos Fessahaye, Luis Perez, Tiffany Zhan, Raymond Zhang, Calais Fossier, Robyn Markarian, Carter Chiu, Justin Zhan, Laxmi Gewali, Paul Oh, T-RECSYS: A novel music recommendation system using deep learning
21. Steve Lawrence, Lee Giles C, Ah Chung Tsoi, Face recognition: a convolutional neural-network approach
22. Lixiang Li, Xiaohui Mu, Siying Li, Haipeng Peng, A review of face recognition technology
23. Zhigang Yu, Yunyun Dong, Jihong Cheng, Miaomiao Sun, Feng Su, Research on face recognition classification based on improved GoogleNet
24. Madan Lal, Kamlesh Kumar, Rafaqat Hussain Arain, Abdullah Maitlo, Sadaquat Ali Ruk, Hidayatullah Shaikh, Study of face recognition techniques: a survey
25. Daa Salama AbdELminaam, Abdulrhman M. Almansori, Mohamed Taha, Elsayed Badr, A deep facial recognition system using computational intelligent algorithms
26. Mingyuan Xin, Yong Wang, Research on image classification model based on deep convolution neural network
27. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, ImageNet classification with deep convolutional neural networks
28. Mohd Azlan Abu, Nurul Hazirah Indra, Abdul Halim Abd Rahman, Nor Amalia Sapiee, Izanoordina Ahmad, A study on image classification based on deep learning and Tensorflow
29. Neha Sharma, Vibhor Jain, Anju Mishra, An Analysis Of Convolutional Neural Networks For Image Classification
30. Abhinav Patil, Image Recognition using Machine Learning

Improvement of Makespan and TCTime in Dynamic Job Ordering and Slot Utilization for MapReduce Workloads



Tanmayi Nagale

Abstract The amount of data generated in today's environment is increasing at an exponential rate. These data are structured, semi-structured, or unstructured in some cases. Processing vast amounts of data is a difficult task. MapReduce is a technique for processing large amounts of data. For storing big data sets, Hadoop data file system (HDFS) is employed. A MapReduce workload is made up of a number of jobs, each of which has multiple map tasks and numerous reduce tasks. In this paper, we propose the Shortest Task Ordering (SJA) algorithm for optimising Mkspan (Makespan) and TCTime (Total completion time) for MapReduce Workloads using dynamic job ordering and slot design. We conducted a comparison analysis on data sets of various sizes. By comparing objective measurements such as Mkspan and TCTime, the experimental results show that performance in terms of speed has improved. Each set of jobs, such as WordCount, CharCount, LineCount, and Anagram, displays comparative improvement in our work. When compared to previous algorithms such as MkJR and MkTctJR, the results demonstrate an improvement in time efficiency, slot usage, and execution speed. In terms of Mkspan and TCTime, the Shortest Task Assigned (SJA) method for job ordering yielded results that were up to 95% better than MkJR. In comparison to the previous algorithms MkSfJR and MkTctSfJR, there is also an increase in slot usage. In terms of Mkspan and TCTime, the Shortest Job Assigned (SJA) algorithm achieved results that were up to 150 percent better than MkSfJR.

Keywords MapReduce workloads · Hadoop · Scheduling algorithms · Job order · Slot configuration

Abbreviations

HDFS	Hadoop Data File System
SJA	Hadoop, Shortest Job Ordering

T. Nagale (✉)

Thakur College of Engineering and Technology, Mumbai, Kandivali (E)Maharashtra, India
e-mail: tanmayinagale13@gmail.com

MkSpan	MakeSpan
TCTTime	Total Completion Time
MkJR	Makespan using Johnson's Rule
MkTctJR	Makespan and total completion time using Johnson's Rule
MkSfJR	Makespan for slot configuration using Johnson's Rule
MkTctSfJR	Makespan and total completion time for slot configuration using Johnson's Rule

1 Introduction

MapReduce [1] is used for refining a huge collection of data or information. A map function is specified by user that processes a key/value pair which is used for generating a set of inter-mediate key/value pairs. A reduce function combines all intermediate values associated with the same intermediate key [2]. Workload is considered as the size of the data for a job. The workload may vary within a job between different tasks. MapReduce and Hadoop support batch processing for jobs submitted from multiple users (i.e., MapReduce workloads). There are several benchmark applications such as WordCount, LineCount, CharCount, Anagram Jobs, etc. These applications are considered as workloads [3]. There are two key performance factors [1] such as Makespan (Mkspan) and Total Completion Time (TCTime). Mkspan is defined as the time period since the start of the first job until the completion of the last job for a set of jobs. It considers the computation time of jobs and is often used to measure the performance and utilization efficiency of a system. In contrast, TCTime is defined as the sum of completed time periods for all jobs since the start of the first job. The system improves the time efficiency and slot efficiency.

The job sequence optimization uses Greedy algorithm based on Johnson's Rule i.e. MkJR for sequencing of jobs. System uses Bi-criteria optimization algorithm such as MkTctJR to optimize the job ordering and improved the Mkspan and TCTime over MkJR. The proposed job ordering algorithm such as SJA that can optimize Mkspan and TCTime. The SJA for job sequencing algorithms will speed up the execution of our system as compare to MkJR and MkTctJR. It improves the system performance in terms of Mkspan and TCTime as well as resolving the time efficiency problem.

In case of slot optimization module, the MkSfJR algorithm will minimize slots configuration for map slot and reduce slots. MkTctSfJR algorithm provides better improvement over MkSfJR. The proposed slot utilization algorithm such as SJA (Shortest Job Assigned) for slots can optimized Mkspan and TCTime under specified slots. It improves the slot allocation of the system as compare to MkSfJR and MkTctSfJR as well as resolving slot efficiency problem.

2 Related Work

Shanjiang Tang, Bu-Sung Lee, and Bingsheng [1] have proposed job-scheduling techniques that reduce the makespan and TCTime. They show that even if some MapReduce servers fail during execution, the optimal orders generated by job ordering algorithms do not alter. The suggested enumeration techniques for map/reduce slot configuration optimization indicate that the optimal map/reduce slot configurations have a proportionate connection for any two alternative total slot sizes. It is critical to handle the suggested enumeration algorithms' time efficiency problem for a high number of total slots.

A new abstraction [2] such as MapReduce has designed by Jeffrey Dean and Sanjay Ghemawat allows expressing the simple computations. It hides the messy details of parallelization, fault-tolerance, data distribution and load balancing in a library. The user specified map and reduce operations which allow us to parallelization large computations easily and to use re-execution as the primary mechanism for fault tolerance.

DynamicMR: A Dynamic Slot Allocation Optimization Framework [3] has proposed by Shanjiang Tang, Bu-Sung Lee, and Bingsheng improves the performance of a single operation at the price of cluster efficiency. For balancing the performance tradeoff between a single work and a batch of jobs, they devised the Speculative Execution Performance Balancing approach. They presented Slot PreScheduling, a novel approach that can increase data locality while having no effect on fairness. Finally, DynamicMR was established as a result of combining these two approaches, which enhances the performance of MapReduce workloads while retaining fairness. The goal of Dynamic Hadoop Slot Allocation is to maximise slot usage by dynamically assigning map (or reduce) slots to map and reduce activities.

The new technique MROrder: Flexible Job Ordering technique [4] proposed by Shanjiang Tang, Bu-Sung Lee, and Bingsheng which is used to dynamically optimize the job order for online MapReduce workloads. MROrder is designed to be flexible for different optimization metrics, e.g., Makespan and TCTime. This work improved the system performance by up to 31% for makespan and 176% for TCTime.

Abhishek Verma, Ludmila Cherkasova, and Roy H. Campbell [5] have introduced a simple abstraction where each MapReduce task is represented as a pair of map and reduce stage durations in this basic abstraction. The Johnson algorithm was created to help you create the best employment schedule possible. This methodology tests the created schedule's performance advantages using a large number of simulations with a range of realistic workloads.

By using simple abstraction technique, the result was achieved up to 10%–25% of makespan improvements by simply processing the jobs in the right order. They have designed a novel heuristic, called BalancedPools [6], which improves Johnsons scheduling algorithm results (up to 15%–38%), exactly in the situations when it produces suboptimal makespan. Overall, they observed up to 50% in the makespan improvements with the new BalancedPools algorithm.

G. J. Kyparisis and C. Koulamas [7] considered a scheduling problem in a two-stage hybrid flow shop, where the first stage consists of two machines forming an open shop and the other stage has only one machine. The main objective is to minimize the makespan, i.e., the maximum completion time of all jobs. They first show that the problem is NP-hard in the strong sense, then they present two heuristics to solve the problem. Computational experiments show that the combined algorithm of the two heuristics performs well on randomly generated problem instances.

S. Tang, B. -S. Lee, and B. He [8], have proposed Dynamic slot allocation technique for mapreduce clusters. They proposed Dynamic Hadoop Fair Schedulers(DHFS) to improve the utilization and performance of MapReduce clusters while guaranteeing the fairness. The core technique is dynamically allocating map (or reduce) slots to map and reduce tasks. Two types of DHFS were presented, namely, PI-DHFS and PD-DHFS, based on fairness for cluster and pools, respectively. The experimental results show that our proposed DHFS can improve the performance and utilization of the Hadoop cluster significantly.

There is another one slot allocation technique such as Flex [9]. Flex is a flexible and intelligent allocation scheme for MapReduce workloads. It is flexible in the sense that it can optimize towards any of a variety of standard scheduling metrics, such as average response time, makespan, stretch, deadline based penalty functions, and even Service Level Agreements (SLAs). This technique has proposed by J. Wolf, D. Rajan, K. Hildrum, R. Khandekar, V. Kumar, S. Parekh, K. -L. Wu, and A. balmin.

Another optimization approach that forms a realistic model for non-preemptive scheduling in MapReduce systems is the Flexible Flow Shop (FFS) [10] issue. The longest-processing-time-first heuristic is utilised for machine allocation in this approach. The difficulty of selecting an appropriate loading sequence is solved using a genetic algorithm. B. Moseley, A. Dasgupta, R. Kumar, and T. Sarlos devised this approach..

T. Nykiel, M. Potamias, C. Mishra, G. Kollios, and N. Koudas [11] introduced a framework called MRShare that merges tasks into groups and evaluates each group as a single query, transforming a batch of queries into a new batch that would be completed more efficiently. They design an optimization issue based on our MapReduce cost model and give a solution that determines the best grouping of queries.

S. R. Hejazi and S. Saghafian [12] considered a flowshop problem with a makespan criterion and it surveys some exact methods (for small size problems), constructive heuristics and developed improving metaheuristic and evolutionary approaches as well as some well-known properties and rules for this problem.

T. Condie, N. Conway, P. Alvaro, J. M. Hellerstein [13] present an overview of the Hadoop MapReduce architecture in Section. They design of HOP's pipelining scheme keeping the focus on traditional batch processing tasks. They also shows how HOP can support online aggregation for long-running jobs and illustrate the potential benefits of that interface for MapReduce tasks. They describe support for continuous MapReduce jobs over data streams and demonstrate an example of near-real-time cluster monitoring.

K. Howard, S. Siddharth and V. Sergej [14] have proposed PRAM model of computation. They describes evaluations methods of a wide class of functions using the MapReduce framework. They developed connectivity methods for computing some basic algorithmic problems.

D. W. Jiang, B. C. Ooi, L. Shi, and S. Wu [15] have introduced about MapReduce programming model and current power enhancement methodologies for clusters that run MapReduce jobs.. They presented MapReduce capabilities, limitations, and current research efforts that aim to enrich MapReduce to manipulate with its limitations. They illustrated how performance of MapReduce can be enhanced by managing skew of data.

3 Proposed System

The proposed system is divided into two parts such as job sequence optimization and slot configuration optimization. In the existing system the job sequence optimization uses Greedy algorithm based on Johnson's Rule i.e. MkJR for sequencing of jobs. As well as it uses Bi-criteria optimization algorithm such as MkTctJR to optimize the job ordering and improved the Mkspan and TCTime over MkJR. The proposed job ordering algorithm (SJA) optimizes Mkspan and TCTime. We are applying the SJA algorithm for job sequencing. It will speeds up the execution of proposed system as compare to algorithms MkJR and MkTctJR used in existing system. It will improve the system performance in terms of Mkspan and TCTime as well as time efficiency. In case of slot optimization, the MkSfJR algorithm of existing system minimize slots configuration for map slots and reduce slots. MkTctSfJR algorithm has better improvement over MkSfJR algorithm of existing system. For slot utilization proposed algorithm SJA (Shortest Job Assigned) will optimize Mkspan and TCTime under specified slots. It will improve the slot allocation of the system as compare to existing algorithms MkSfJR and MkTctSfJR as well as slot allocation efficiency.

Figures 1 and 2 shows flow of Job ordering algorithm and slot configuration algorithm respectively.

Figure 3 shows architecture of system. Architecture consist of two parts Job ordering optimization and slot configuration optimization.

In this paper, we look at four optimization issues, which are as follows:

- i. Create an ordering sequence to execute a specific number of tasks in such a way that Mkspan is reduced.
- ii. Create an ordering sequence for jobs that will simultaneously increase Mkspan and TCTime under the stated slot configuration.
- iii. Identifying the map phase and reduction phase allocation of slots, as well as the ordering sequence for executing the tasks so that Mkspan is reduced under a given total slot value.

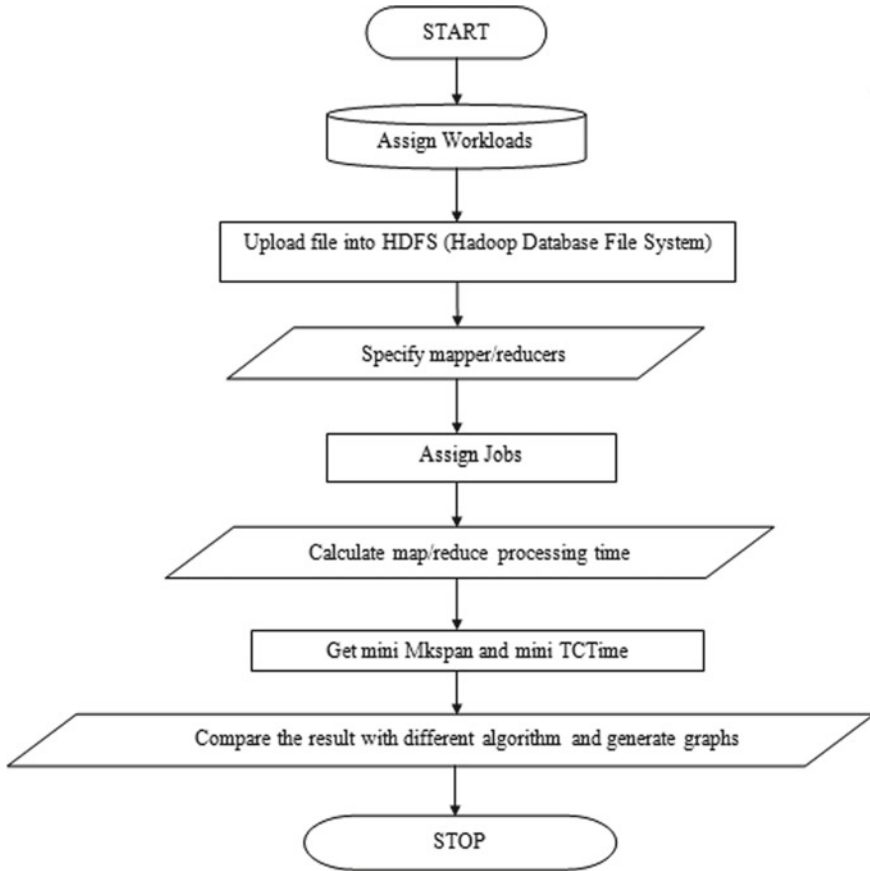


Fig. 1 Flow of Job ordering algorithm

- iv. Identification of map phase and reduction phase slot allocation and ordering sequence for executing tasks that maximise Mkspan and TCTime under a given total slot value. Experimental Setup

Software Requirements

- 1. Operating System: Ubuntu 14.04 or higher
- 2. Developing language: Java (JDK 1.8)
- 3. IDE: Eclipse Kepler
- 4. Hadoop: Hadoop 2.7.1
- 5. Hbase: Hbase 1.2.6

Hardware Requirements

- 1. Intel processor i3 or higher
- 2. 16 GB RAM or higher

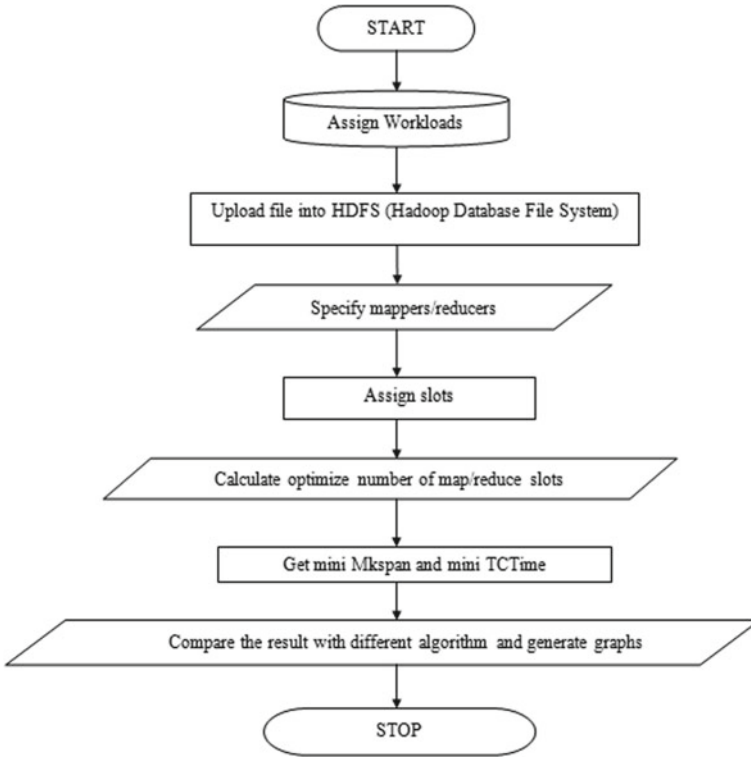


Fig. 2 Flow of slot configuration algorithm

- 3. 500 GB or more hard disk
- 4. Proposed Algorithm

(1) **Shortest Job Ordering (SJA) for sequencing of jobs**

Input

- J: The map reduce workload
- SM: The number of map slots
- SR: The number of reduce slots

Output

- J': The optimized job order
- MkSpan: Makespan
- TCTime: Total completion time

Steps

1. Job Ordering by MkJR
2. Arrange job in such a way that shortest Job will Assigned First using their execution time as weight

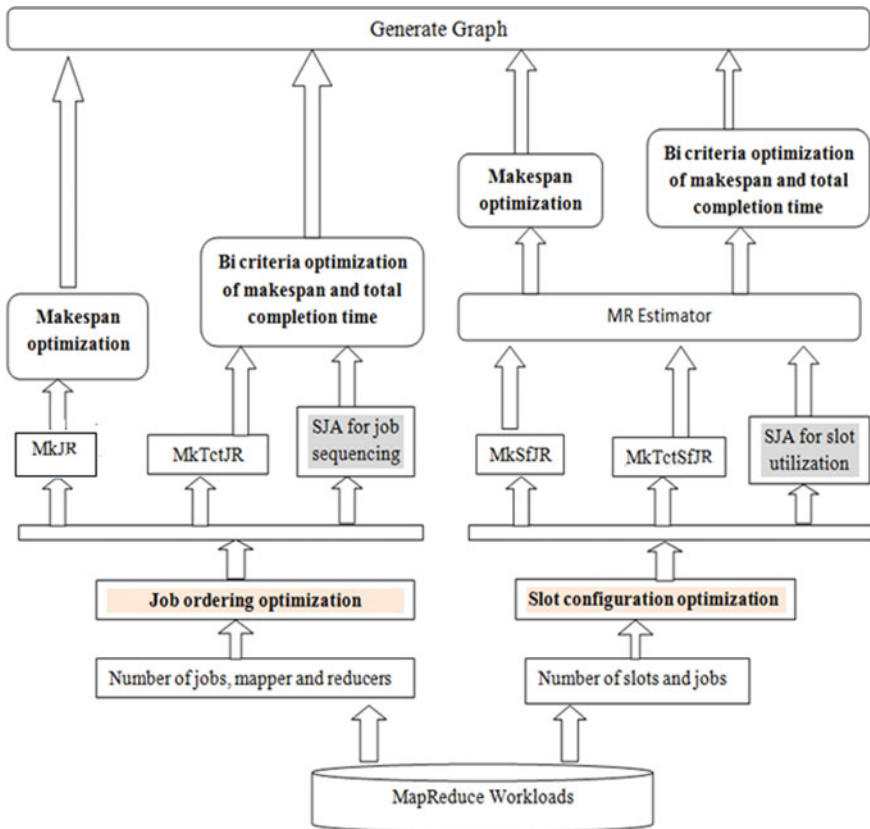


Fig. 3 System architecture

3. Repeat MkJR Algorithm, $MkJR(J, SM, SR)$
4. J' , MkSpan, TCTime.

(2) **Shortest Job Ordering (SJA) for slot utilization**

Input

- J: The map reduce workload
- SM: The number of map slots
- SR: The number of reduce slots

Output

- J' : The optimized job order
- MkSpan: Makespan
- TCTime: Total completion time

Steps

1. Slot allocation by MkSfJR.

2. Allocate the slots to each set of job.
3. Repeat MkSfJR Algorithm, MkSfJR(J,SM,SR).
4. J', MkSpan,TCTime.

4 Experimental Results

We considered four benchmark applications such as WordCount, LineCount, Char Count, AnagramJob.

1. **Word Count:** Computes the occurrence frequency of each word in a document.
2. **Line Count:** Computes the occurrence frequency of lines in a document.
3. **Char Count:** Computes the occurrence frequency of each characters in a document
4. **Anagram Job:** A word or phrase that is made by rearranging the letters of another word.

Figures 4, 5 shows Map and Reduce Phase time for jobs like Anagram, Charcount, Wordcount, Linecount, etc.

Fig. 4 Map phase time

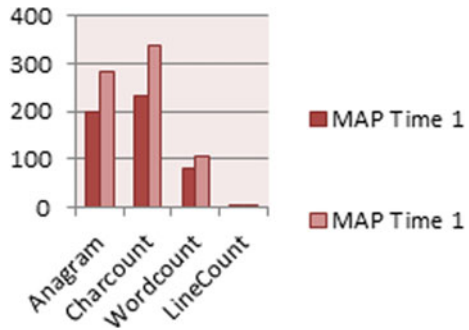


Fig. 5 Reduce phase time

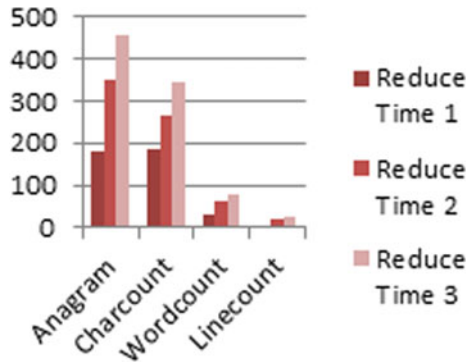


Fig. 6 Job ordering using MK_JR

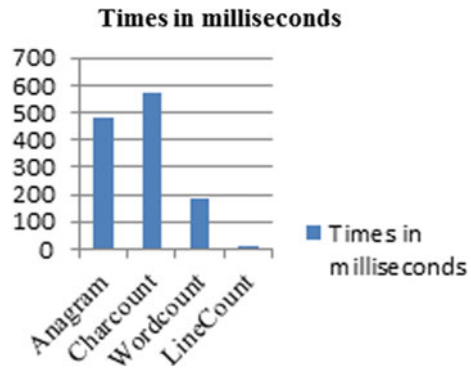
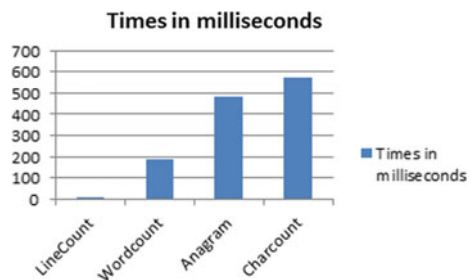


Fig. 7 Job ordering using MK_TCT_JR



Figures 6, 7 shows job ordering using Mk_JR and Mk_TCT_JR for jobs like Anagram, Charcount, Wordcount, Linecount, etc.

Figure 8 shows improvement in system performance in term of MkSpan for each set of jobs using MkJR and MkTCTJR algorithm. Using MkTCTJR there is improvement of 15 to 95% in terms of Makespan.

Figure 9 shows improvement in system performance in term of MkSpan for each set of jobs using MkJR and SJA algorithm. Using SJA there is improvement of 45 to 150% in terms of Makespan.

Figure 10 shows improvement in system performance in term of TCTime for each set of jobs using MkJR and MkTCTJR algorithm. Using SJA there is improvement of 45 to 150% in terms of TCTime.

Figure 11 shows improvement in system performance in term of TCTime for each set of jobs using MkJR and SJA algorithm. Using SJA there is improvement of 45 to 150% in terms of TCTime.

To evaluate job ordering with workload, we use SJA for jobs to compute the makespan as well as total completion time. The Shortest Job Assigned (SJA) algorithm is used for sequencing of jobs as well as to optimize the Makespan (Mkspan) and Total Completion Time (TCTime). It speeds up the execution of system and improves time efficiency. The proposed Shortest Job Assigned (SJA) algorithm produce the results that are up to, 15—95% better than existing Greedy algorithm based on Johnson’s Rule (i.e.MkJR) in terms of Mkspan as well as TCTime.

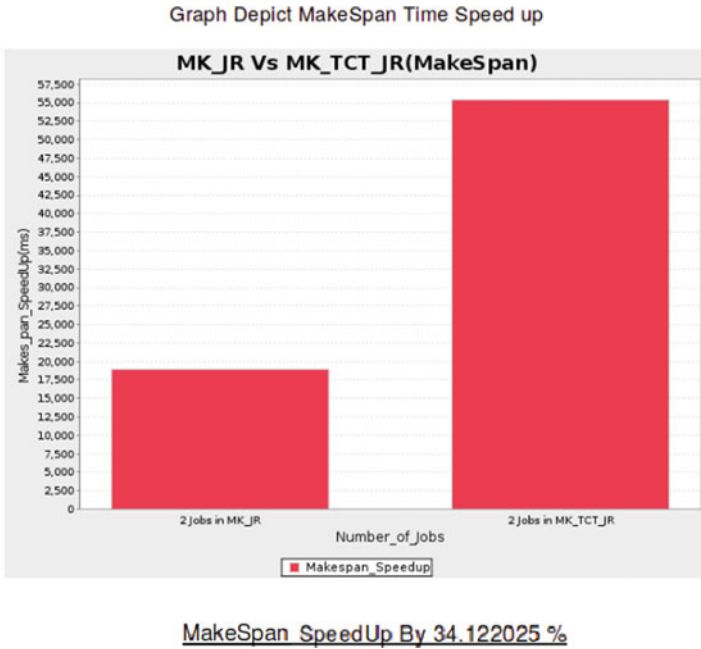
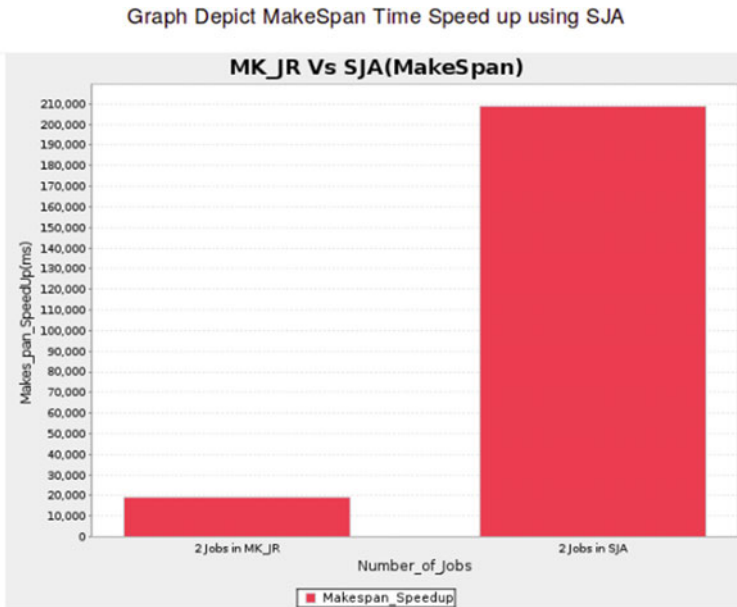


Fig. 8 Makespan speedup using MKJR versus MKTCTJR

The following Table 1 and Fig. 12 shows that comparative analysis of different file size inputs such as 160 MB, 322 MB, 467 MB, 768 MB, 1GB under Job Sequencing Optimization.

To evaluate slot utilization with workload, we use SJA for slots to compute the makespan as well as total completion time. The Shortest Job Algorithm (SJA) is used for utilizing slots as well as to optimize the Makespan (Mkspan) and Total Completion Time (TCTime). It speeds up the execution of system and improves slot efficiency. The proposed Shortest Job Assigned (SJA) algorithm produce the results that are up to, 45–150% better than existing Search algorithm for optimized slot configuration and job submission order (MkSfJR) in terms of Mkspan as well as TCTime.

The following Table 2 and Fig. 13 shows comparative analysis of different file size inputs such as 160 MB, 322 MB, 467 MB, 768 MB and 1 GB under Slot Utilization Module.



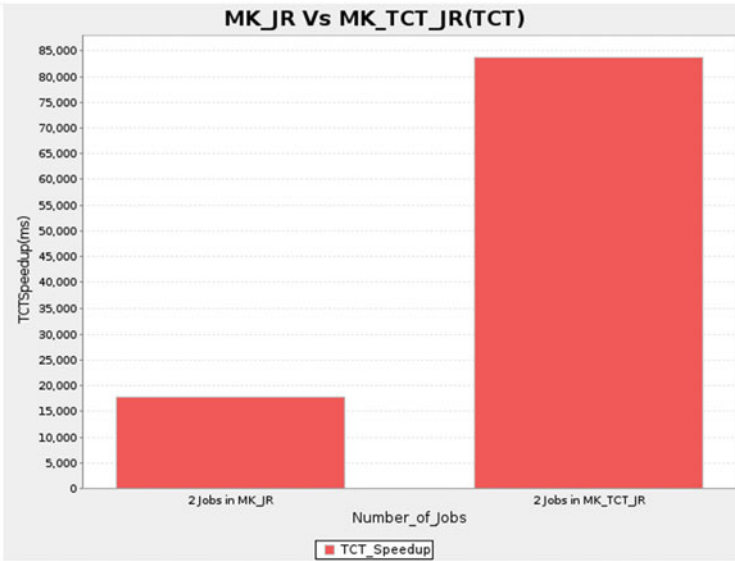
MakeSpan SpeedUp By 109.05011 %

Fig. 9 Improved Makespan speedup using MKJR versus SJA

5 Conclusion and Future Work

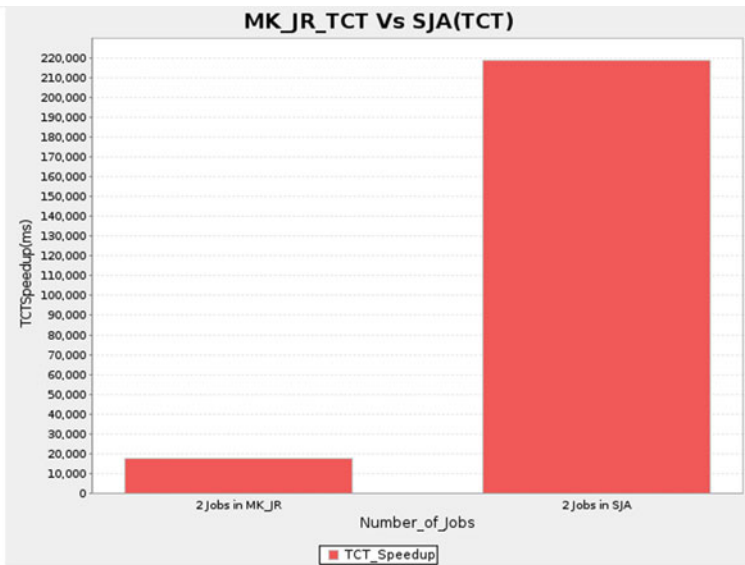
The comparative result analysis of proposed algorithm SJA with the algorithm used in existing system to optimize job sequencing for MapReduce workloads as well as to optimize slot configuration and its utilization improves significantly. The SJA algorithm for job ordering of MapReduce Workloads improved results up to 15 to 95% than MkJR and MktctJR in terms of MkSpan as well as TCTime. The SJA algorithm for slot configuration and its utilization of MapReduce Workloads improved results up to 45 to 150% than MkSfJR and MktctSfJR in terms of MkSpan as well as TCTime.

In future, we can work on different schedulers for improving the performance of system in terms of speed. As well as we can use different algorithms of job sequencing and slot optimization. We can also work on on-line workloads too.



TCT SpeedUp By 21.265808 %

Fig. 10 TCT Speedup using MKJR vs MKTCTJR



TCT SpeedUp By 108.13658 %

Fig. 11 Improved TCT Speedup using MKJR vs SJA

Table 1 Job sequencing optimization

Job sequencing optimization			MkJR and MkJTCTJR		MkJR Vs SJA	
File Size	Mapper	Reducer	Optimized makespan (%)	Optimized TCT (%)	Improved makespan (%)	Improved TCT (%)
160 MB	5	3	62	91	94	95
322 MB	3	1	84	96	98	98
467 MB	8	5	42	45	62	59
768 MB	6	1	87	96	98	98
1GB	8	5	50	84	90	91

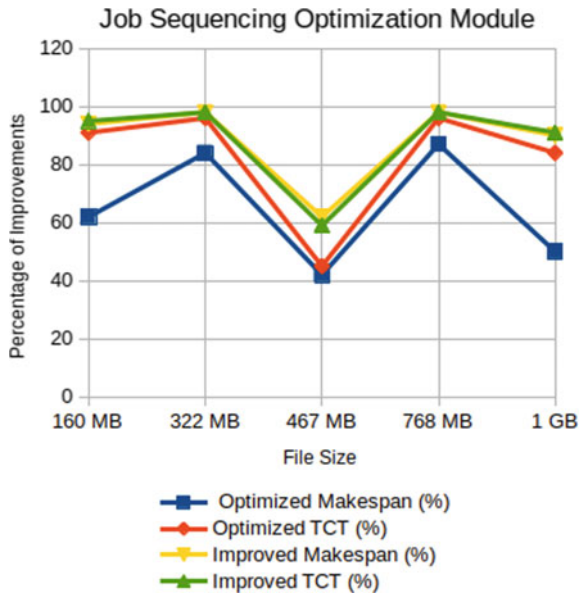
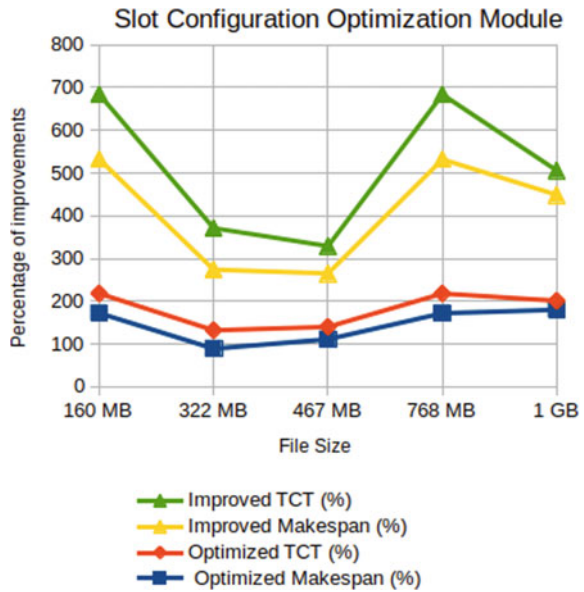


Fig. 12 Comparative result analysis of proposed algorithm SJA with existing algorithm to optimize job sequencing for MapReduce workloads

Table 2 Slot configuration optimization

Slot configuration optimization					
Inputs		MkJR and MkJRTJR		MkJR Vs SJA	
File Size	Slots	Optimized makespan (%)	Optimized TCT (%)	Improved makespan (%)	Improved TCT (%)
160 MB	5	172	46	314	152
322 MB	5	89	43	142	97
467 MB	5	111	29	125	64
768 MB	5	172	46	314	152
1 GB	5	180	21	248	57

Fig. 13 Comparative result analysis of proposed algorithm SJA with existing algorithm to optimize Slots for MapReduce workloads



References

1. Shanjiang Tang, Bu-Sung Lee, Bingsheng He (February 2016) Dynamic job ordering and slot configurations for mapreduce workloads, IEEE Trans Serv Comput Vol-9, pp 4–17
2. Dean J, Ghemawat S (2004) MapReduce: Simplified data processing on large clusters In: Proceedings of the 6th Conference on Symposium on Operating Systems Design Implementation, Vol-6, pp 10
3. Shanjiang Tang, Bu-Sung Lee and Bingsheng (July 2014) DynamicMR : A dynamic slot allocation optimization framework for MapReduce clusters, IEEE Trans Cloud Comput, 2(3), pp 333–347
4. Tang S, Lee BS, He B, (2013) MRorder: Flexible job ordering optimization for online MapReduce workloads. In Proceedings of the 19th International Conference On Parallel Processing, pp 291–304

5. Verma A, Cherkasova L, Campbell RH (December 2013) Orchestrating an ensemble of mapreduce jobs for minimizing their makespan, *IEEE Trans Depend Secur Comput*, 11(4), pp 390–391
6. Ludmila Cherkasova, Roy H. Campbell, Abhishek Verma (September 2012) Two sides of a coin: optimizing the schedule of MapReduce jobs to minimize their makespan and improve cluster performance, pp 11–18
7. Kyparisis GJ, Koulamas C (2006) A note on makespan minimization in two-stage flexible flow shops with uniform machines. *Eur J Oper Res* 175(2):1321–1327
8. Tang S, Lee B-S, He B, (Sep. 2013) Dynamic slot allocation technique for mapreduce clusters, In: *Proceedings. IEEE International Conference on Cluster Computing*, pp 1–8
9. Wolf J, Rajan D, Hildrum K, Khandekar R, Kumar V, Parekh S, Wu K-L, Balmin A (2010) Flex: A slot allocation scheduling optimizer for mapreduce workloads, In *Proc. ACM/IFIP/USENIX 11th Int. Conf. Middleware*, pp 1–20
10. Moseley B, Dasgupta A, Kumar R, Sarlos T, (2011) On scheduling in map-reduce and flowshops. In: *Proceedings of the 23rd Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pp 289–298
11. Nykiel T, Potamias M, Mishra C, Kollios G, Koudas N, (Sep. 2010) Mrshare: Sharing across multiple queries in mapreduce. In: *Proceedings of the VLDB Endowment*, 3(1/2), pp 494–505
12. Hejazi SR, Saghaifan S (2005) Flowshop-scheduling problems with makespan criterion: A review. *Int J Production Res* 43(14):2895–2929
13. Condie T, Conway N, Alvaro P, Hellerstein JM (2010) MapReduce online. In: *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, pp 21C21
14. Howard K, Siddharth S, Sergei V (2010) A model of computation for MapReduce. In: *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pp 938–948
15. Jiang DW, Ooi BC, Shi L, Wu S (2010) The Performance of MapReduce: An Indepth Study. *PVLDB* 3:472–483

Identification and Detection of Plant Disease Using Transfer Learning



Neelam Sunil Khasgiwala and R. R. Sedamkar

Abstract Many researchers have recently been inspired by the success of deep learning algorithms in the field of artificial intelligence to improve plant disease detection performance. Deep learning's main goal is to teach computers how to solve real-world problems using data or experience. Detecting diseases is a critical task for farmers. They take shortcuts such as using chemical pesticides, which have negative effects on consumable foods. So, in this paper, we used deep learning algorithms to detect plant diseases. Deep learning is a popular trend in which technological benefits can be imparted to the agricultural field. Detecting plant diseases with deep learning techniques is less expensive than using chemical pesticides. This paper reviews existing techniques and recommends the best technique that farmers can use to identify disease faster and at a lower cost.

Keywords Plant Disease · CNN · Disease Recognition · Image Processing · Agriculture · Deep Learning

1 Introduction

India, being a vast country with multiple cities and towns proves to employ large proportion of its population in rural areas. With an approximate figure of 60% the total land area is used for cultivation and feeding purpose of human and livestock [1]. The agriculture in today's time is heavily reliant on technology and tends to focus on maximizing profits from the selected crops, which in the long run degrade the soil's physical and biochemical properties. On the other hand, various measures can also be taken to maximize land yields while maintaining soil fertility. Hence this becomes a way for sustainable agriculture. In recent times, organic food is considered to be the urgent food, providing nutrition to billions of people across the globe. However, certain surveys have also been conducted that focuses on weed tracking and cultivation practices.

N. S. Khasgiwala (✉) · R. R. Sedamkar
Computer Engineering Department, Thakur College of Engineering & Technology, Mumbai, India
e-mail: neelamkhasgiwala@gmail.com

In the sector of agriculture practices of machine learning and non-destructive screening are considered to be very important. Also since machine learning is considered to be as one of the most enthralling branches of computer science it is majorly used in multiple domains for the same. The capability of machine learning is that it tends to think and act similar to that of a human when it is being provided with multiple historic data. This data is later fed into the system so that processing might take place in the initial stages. Machine learning tends to advance the overall accuracy of the system and traditionally tends to set rule algorithms. Hence, it is said that they are capable of analyzing large number of results in shortest span of time. The entire working of machine learning is based on developers input which it gives to the programming language. Also these algorithms are capable to analyze any situation by feeding combination of such conditions so that it can forecast the future and measure the extent of signification. CNN is majorly used as one of the deep learning techniques that make use of feature extraction and data reduction. Apart from this certain convolutional parameters are also being put to use by extracting salient features from the data. CNN differs from ANN in image classification due to feature extraction and dimensionality reduction. Traditionally, statistical analysis and models were used to forecast such parameters. In terms of computational potential, models based on deep learning are very much reliable and optimized in comparison to statistical models. On the other hand, the execution of plant leaf disease detection necessitates a large and diverse dataset for image classification of plants. Hence, the detection of plant diseases becomes important in agriculture as farmers frequently decide whether the crop they are harvesting is good enough or not. However this decision is critical to make as it might lead to serious problems in plants which in turn would lead to plant productivity. Diseases in plants often results into disease outbreaks on a regular basis that would resulting large scale deaths; thereby impacting the overall economy of the country. Hence, these issues must be addressed so that the lives of people can change drastically and be saved. On the other hand, the automated analysis of plant pathogens tends to become an important filed of research in this domain as it tends to monitor huge expanses of crops and further detect clinical signs that might appear on the leaf in the early stages. This result the exiting algorithms to perform image based automatic inspection with low labor incentive involved along with less price. This approach further allows all the plant related diseases to be identified in the early stages and further prevent them from occurring on a larger scale.

This Sect. 2 paper includes literature reviews of previously published papers. Section 3 contains a comparative study. Section 4 includes the conclusion. Section 5 discusses Deep Learning techniques in general, and Sect. 6 discusses future research and experiments in plant leaf disease detection, as well as the conclusion.

2 Literature Survey

In recent times, multiple papers and research work has been published that are completely based on the identification of plant diseases. In one of the presented research works by S. Ramesh et al. [2] the author proposed four common diseases found in plants and worked and contributed his work onto detecting those diseases. His work primarily focused on building an algorithm based on the concepts of DNN that could not only classify but also recognize the presence of the disease in similar plants. The images of the diseases leaves were directly captured to fulfil the purpose of creating a dataset. These images were further sent for the pre-processing stages; wherein the RGB images were converted to HSV images and later the binary images were used to generate diseases portions of plants. In the later stages, a clustering method was also adopted to detect and classify these portions. According to the results so obtained the overall accuracy so obtained was 92% for a healthy leaf and 96% for a diseased leaf. In another work by Reddy in [3] he proposed to detect the diseases in plants and further classified it as healthy and diseased. The author implemented the model in stages of pre-processing and feature extraction. He further segmented the plant images on the basis of various diseases the leaf underwent and performed image cleaning techniques. In the later stages he proposed a machine learning based model that could high light the spots on the diseased leaves. This was done by resizing the image. For the purpose of segmentation the author used three segmentation techniques namely: K means, HSI and Otsu's classification. The author then transported the images which were RGB based into the labs where the image could be pre-processed before segmentation could be performed on it. In the feature extraction stage, the author implemented the extraction strategy using contrast and pixel correlation of the images. The final classification was being made and the leaves were labelled as either healthy or diseased. This classification was performed by machine learning models namely: K means and SVM. In the execution phase, the implementation of the model using SVM outperformed the implementation of K means. The overall accuracy the model achieved using SVM was 96% and using K means was 85%. In another research study by Konstantious in [4] the author proposed a similar model for disease detection amongst plants. However he made use if the CNN model to classify and detect the same. The author implemented the image dataset on various pattern recognition techniques. This dataset contained a total of 87,848 images of plant leaves that was used to model the complex process of pattern recognition. This method was used to classify and detect 25 various other plants and further classify it as healthy or diseased. The model obtained an overall accuracy of 98%. This model however was used to detect similar other cases using the fundamentals of deep learning.

In another work by Juncheng Ma in [5] the author implemented the classification model using the concepts of DNN and also worked upon the identification of four kinds of disease commonly found in plant leaves. Once the disease was diagnosed, the author performed data augmentation techniques so that the dataset could be expanded. The process of data augmentation leads to the generation of 14,000 plant

images and the model proposed by the author was successfully able to achieve an overall accuracy of 93%.

In another work by an author in [6] he proposed and presented a system that could determine the efficiency of the crop production in the agricultural domain. However the author made use of three datasets from multiple domains that included the details of soil, rainfall and yield dataset. He combined these three datasets into one and applied multiple machine learning algorithms on it in order to determine the overall accuracy of the system. This accuracy was obtained after performing the training and testing phase. Once all the machine learning algorithms were used, a comparative analysis was carried out amongst all the algorithms. In the later stages, while the author compared all the algorithms he induced the error rate and the accuracy range for the proposed model. However, the best results were obtained with the accuracy of the training model being higher than the error rate; and the error rate being as minimal as possible. The author also calculated the actual cost of the crop and further labelled it as high or low. In a similar work by the authors in [7] they developed a CNN based neural network that detect the presence of such diseases in the leaves of plants. The authors collected a set of leaf images and executed their model on them. They made use of various image processing techniques that was further capable to identify and classify the leaves as healthy or diseased (Fig. 1).

In a similar work by authors in [2] they developed and created a method that could easily distinguish a plant leaf as healthy or diseased. This model was a bit different as it utilized a clustering algorithm based on fuzzy means that was able to detect the presence of disease in wheat leaves. However the authors achieved an overall accuracy of 96% for diseased images and 92% for healthy images. In a similar work the authors made use of DSIFT features that could automatically predict the disease



Fig. 1 Survey on application of machine learning in agriculture

in wheat leave. The authors combined its implementation with k means clustering and the DSIFT features were tested against the k means approach to implement the same. The authors managed to achieve an overall accuracy of 87% [3]. In a similar study of work, the authors developed a [4] robotic based system that could detect weed mapping that could be used for automating harvesting robots.

3 A Taxonomy on Deep Learning

After reviewing numerous methodologies, as discussed above in the survey section of this paper, we concluded that the machine learning technique is appropriate for solving this problem. This section focuses on a deep learning concept, which is a machine learning advancement inspired by human brain nervous system computation [5]. The term „deep’ refers to deep layers, as well as computation and interpretation using statistical methods and vector operations. The data is held in the internal workflow of deep learning models, which consists of nodes; the arrow indicates numerical operations on the data. Back propagation algorithm performs both forward and backward propagation in an artificial neural network, which is known as feed forward.

A. Transfer Learning

It is a concept where the information extracted and implemented on a given dataset can be applied to a new dataset with a much smaller population to train provided; both the datasets work on a similar task of CNN architecture [6]. In a conventional CNN, this process is executed by training initial parameters on large datasets. After successful training a CNN is capable to extract significant features. Based on the potential of a CNN to extract features, a specific model is selected for transfer learning [7]. This approach is called as feature extraction. The primary objective of this strategy is to retain both: Architecture of a CNN model and the neuron weights. This concept is generally used to compensate computational cost of developing a neural network from scratch. The second strategy involves, adopting one of the many CNN based variant models such as Alexnet, Densenet, Mobilenet, Inception and VGG-16, wherein certain parametric adjustments are made to the model, to achieve optimal results [2].

VGG is an acronym used for Visual Geometry Group and was released with two variants under the CNN model; VGG-16 and VGG-19 [3]. The fundamentals of this series were developed to be applied for image classification problems. All the 16 layers of a conventional VGG-16 model are depicted in figure below (Fig. 2).

The comprehensive framework of the model includes 5 sets of convolutional layers, and a MaxPool layer. The convolution kernel has a size of 3×3 , and the pooling kernel has a size of 2×2 . The most significant evolution to VGGNet is the minimised dimension of a convolutional kernel and an increase in the number of convolutional layers [4].

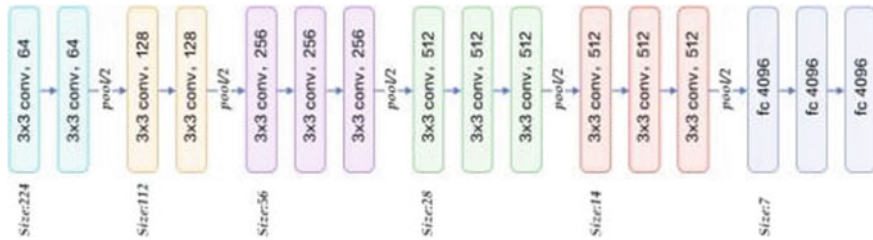


Fig. 2 Layers of VGG-16

Transfer learning is a form of deep learning that makes use of previously trained network so that the same functions can be able to perform on the new task [5]. This concept is widely used in deep learning as it allows the networking model to get trained on a much smaller amount of data. The fundamentals of this network make use of previous knowledge and utilize this knowledge to take further decisions. These decisions are majorly used to contribute in increasing the accuracy of the overall system [6].

B. Convolutional Neural Network (CNN)

CNN is more compatible than ANN because of Convolutional Neural Network feature extraction. ANN does not extract features while deep computing with deep hidden layers. Convolutional layers are filters that perform convolutional operations on given data to extract high level features [7]. Figure 3 depicts the CNN workflow.

The convolutional layer's job is to extract features. It learns to recognise spatial features in input images [2]. This layer is created by applying a variety of image filters to an input image. Convolutional kernels are the name given to these filters. A filter is a small grid of values that slides over an input image pixel by pixel to produce a filtered output image the same size as the input image. Different kernels will generate different filtered output images. If we have three different kernels, these three kernels

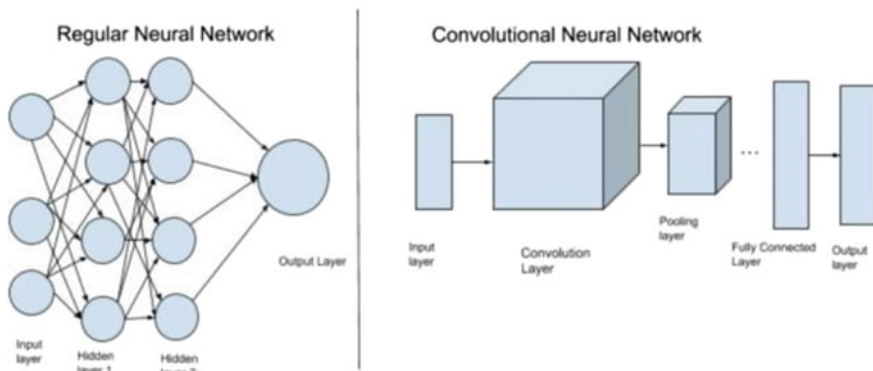


Fig. 3 Layers of CNN

will produce three different filtered output images. The basic idea is that each kernel will extract a different feature from an input image, and these features will eventually help in classifying the input image (ex: a cat or a dog). When these filters are stacked together, they form a convolutional layer. The convolutional kernels are in the form of matrices which are just grids of numeric values that modify an image.

Figure 4 depicts the operation of the convolution layer for a 5×5 image input, with the result being a 3×3 filter that has been reduced in size. In addition, the figure depicts the filter shifting starting from the upper left corner of the input image. The values for each step are then multiplied by the filter values, and the sum of the values is the result. The input image is used to create a new matrix with a smaller size.

Pooling layer minimizes overfitting and reduces neuron size for the down-sampling layer. Figure 5 depicts an example of a pooling operation. This layer reduces the size of the feature map, the number of parameters, the training time, the computation rate, and it controls overfitting. A model defines overfitting as achieving 100% on the training dataset and 50% on the test data. To reduce feature map dimensions, ReLU and max pooling were used.

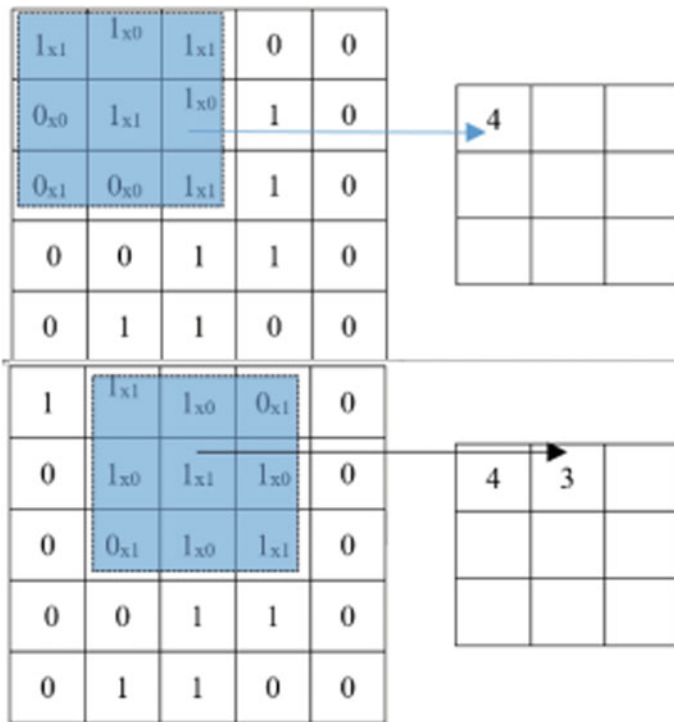


Fig. 4 5×5 input and 3×3 filter operation of convolution layer

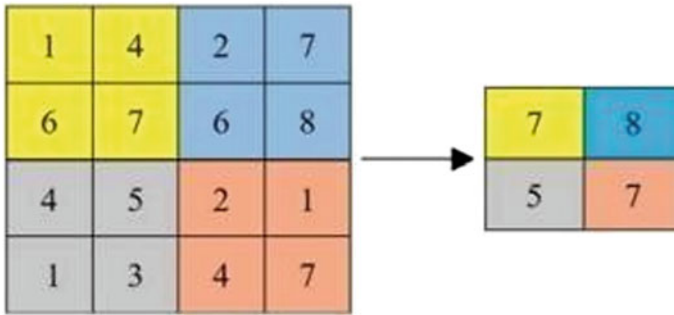


Fig. 5 Pooling layer

Activation Layer is a non-linear ReLU (Rectified Linear Unit) layer used in each convolution layer. This layer also employs dropout layers to prevent overfitting.

Fully Connected Layer this layer is majorly used to analyze the involved class probabilities with respect to the input of the classifier. Softmax is a well-known classifier generally used as an input.

The dataset is initially divided into training and validation purpose with a total of 80 and 20% of data respectively. The augmentation settings are applied in the beginning of the implementation with each operation carrying a weighted probability of trained epochs. These operational settings include flipping and padding mode followed by zoom with crop operations. All the plant images are resized using a compress function and the model is further given for training purpose. For the classification, we used the concept of transfer learning.

Figure 6 depicts the Input Dataset, Image Acquisition, Image Pre-processing, and Classification block diagrams.

4 Proposed Method

A. Dataset

The entire dataset used for the implementation of the study was taken from “The PlantVillage Dataset” that contained a total of 54,323 plant images. A specific number of classes from the dataset were chosen for every species. However, all the images were captured in a controlled environment. In the later stages, to access the dataset, a test data was generated containing a total of 50 images that were sourced from Google. These images were plant anatomy based with the background data containing various stages of the respective disease (Fig. 7).

Image Acquisition: The dataset of the image that was used to train the network was obtained from the Plant Village repository. The images used for the implementation of the study of plant diseases were downloaded from the repository using a Python



Fig. 6 Sample dataset

script. The acquired dataset contains approximately 35,000 images of plant varieties and diseases from 32 different classes.

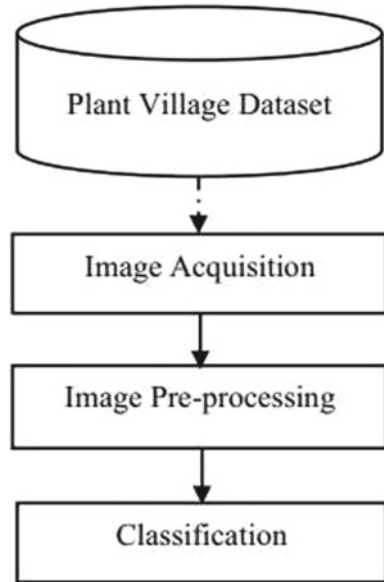
Image Pre-processing: Pre-processed images have their image size and cropped to fit a given input. It improves the obtained image to the required color scale. For processing, the study employs colored and resized images to 96×96 resolutions [3].

Classification: Fully connected layers are used for classification, while convolutional and pooling layers are used for feature extraction. The classification process determines whether or not a plant leaf is infected with a disease, identifies the type of plant disease, and recognizes the plant variety.

B. Model 1: CNN Implementation

The execution of this model takes place in two parts, namely—feature extraction and classification. Initially the image of the plant is preceded through four layers of a convolutional network that is responsible to extract features that are further given as

Fig. 7 Plant leaf detection and disease recognition methodology



inputs to the network for classification. The number of kernels in each convolutional layer is 64, 64, 128, 28, 256, 256 with the filter size of each kernel as 9×9 . ReLu is used as the activation function on all layers followed by pooling layer, primary caps layer, digitCaps layer and dense layer. The compilation of the model is achieved with adam optimizer with 0.001 as the initial learning rate. To calculate margin loss function, the authors operate on the entities of the vector by measuring its length whether the probability of plant features of a particular class exists. The higher level layer CNN for a digit class k will have a long entity vector depending on the presence of a digit in a particular image. However, the overall accuracy of the model network is achieved to be 91.71% when implemented using CNN.

C. Model 2: CNN + VGG16 Implementation

As described in the taxonomy, VGG16 is a heavily trained CNN based model that is used to extract features in order to assess the efficiency of convolutional characteristics. This model has the working principle of a VGG16 model that is built on the architectural concept of a CNN model. It is worthy to note here that after extensive research it was witnessed that the VGG16 model tends to be a better choice in comparison to other models such as MobileNet and AlexNet. All the existing images are scaled down to 224×224 pixels and fed to the input layer of the VGG16 network. The working implementation is carried out through the VGG16 layers preceded by average pooling and dense layer. The compilation of the model is accomplished through adam optimizer accompanied with a batch size of 8 and is further executed for 30 epochs. However, the overall accuracy of the model network is achieved to be 96.80% when implemented using VGG 16 (Fig. 8).

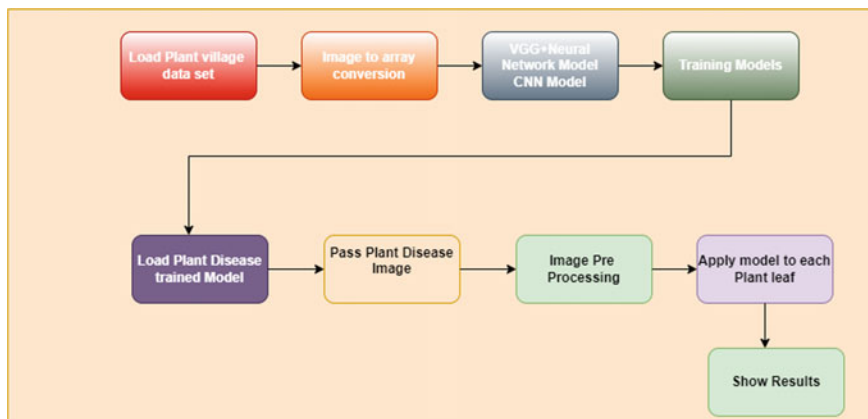


Fig. 8 Block diagram of the working process

5 Result & Conclusion

In this study, the authors conducted a transfer learning analysis based on CNN model to identify the existence of plant diseases. The study, proceeds by implementing the fundamentals of deep learning along with concepts of CNN. However, through an intense survey it was observed that CNN's do exhibit certain flaws and undergo specific limitations. The priority of the research is to detect the presence of infection in plants and further classify it as either infected or normal. This objective is achieved by executing the research study in two halves, using two models and concluding which model works better giving higher efficiency. Implementation of model 1 took place on capsule networks and model 2 on VGG16. Through parametric functions it was observed that model 1 attained an accuracy of 91%; and on the other hand, model 2 accomplished maximum accuracy of 96%. Therefore, on comparison it is observed that the VGG16 model works and performs better.

Acknowledgements I would thank my guide Dr. R.R Sedamkar, Professor of Computer Engineering Department and Dr. Sheetal Rathi , HOD of Computer Engineering Department, for their support and guidance for this paper on the topic "Identification and Detection of plant Disease using Transfer Learning". This current study was victoriously carried out in Thakur College of Engineering and Technology, Mumbai.

References

1. <https://en.wikipedia.org/wiki/Agriculture>
2. Ramesh S, Vydeki D (2019) Recognition and classification of paddy leaf diseases using Optimized Deep Neural network with Jaya algorithm, Information Processing in Agriculture
3. Reddy JN, Vinod K, Ajai ASR (2019) Analysis of classification algorithms for plant leaf disease detection InProceedings of 2019 3rd IEEE International Conference on Electrical, Computer and Communication Technologies ICECCT 2019, pp 1–6, <https://doi.org/10.1109/ICECCT.2019.8869090>
4. Konstantinos P (2018) Ferentinos, Deep learning models for plant disease detection and diagnosis. Comput Electron Agric 145:311–318
5. Juncheng Ma, Keming Du, Feixiang Zheng, Lingxian Zhang, Zhihong Gong, Zhongfu Sun, A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neuralnetwork, Comput Electron Agric 154, 2018, pp 18-24
6. Arun Kumar, Naveen Kumar, Vishal Vats, Efficient Crop Yield Prediction Using Machine Learning Algorithms. June, 2018
7. Ferentinos KP (2018) Deep learning models for plant disease detection and diagnosis. Comput Electron Agric 145:311–318

Blockchain Based E-Voting System



Mahima Churi, Anmol Bajaj, Gurleen Pannu, and Megharani Patil

Abstract The process of establishing democracy in a country is defined by election. Elections might just be a significant event in today's democracy, however many segments of population throughout the world lack faith in their electoral system, which is a major source of concern for democracy. Even the world's most powerful democracies, such as the Republic of India, the United States, and Japan, have a flawed legal system. To eliminate these drawbacks of the electoral system, Blockchain Technology is considered as an ultimate solution. With the present surge in sales and use of blockchain technology for a number of purposes, including banking, medical, and identity, a lot of attention has been focused on the legal concerns rather than its practical uses in administration. In this paper, we discuss the concept of Blockchain in a detailed manner making the reader understand the working of blockchain, its characteristics etc. everything from the scratch, as well as how this concept can be implemented as an efficient solution for public voting and how it is more beneficial from the traditional voting methods, with the goal of eliminating the drawbacks of India's current electoral system while also providing a better, more trustable, safe, and transparent means of public governance. Blockchain is really an emerging technology that promises to improve the resiliency of electronic voting systems. This method offers a way to profit from blockchain's advantages, such as cryptological foundations and transparency, to achieve an efficient theme for the e-voting system.

Keywords Distributed ledger · Hash value · Timestamp · Blockchain · Cryptocurrency

1 Introduction

Electoral integrity is important not just for democratic countries, but also for public voters' transparency and trust. Political voting methods are crucial in this sense. From

M. Churi (✉) · A. Bajaj · G. Pannu · M. Patil
Thakur College of Engineering and Technology, Mumbai, India
e-mail: mahimachuri.28@gmail.com

a government standpoint, electronic voting methods have the potential to enhance participation. Voter turnout and confidence are both increasing, and so are interest in the voting system, to meet all of these requirements, E-voting has shown to be the most effective way of satisfying voters' rights while also providing elected members with a sense of fulfilment. Electronic voting is a voting method in which electronic devices are used to record or tally the number of votes cast. Such systems may be capable of carrying out a variety of activities, ranging from election setup through vote storage [1]. Apart from the benefits listed above, E-voting has significant drawbacks that have led to it being deemed faulty by the security community, particularly owing to physical security issues. Such Security issues with respect to e-voting systems have been the topic of concern that is extensively studied in Literature [2]. This is when blockchain technology may be seen as a benefit to the E-voting system, allowing it to become more efficient and effective. Blockchain is a new technology that has revolutionized the industry in recent years. Blockchain has recently gained popularity as a technique for increasing the efficiency of systems in a variety of industries. The first and most well-known use of blockchain technology was to keep track of bitcoin transactions. In Cryptocurrency, the fundamental blockchain technology plays a vital role through which another application of it such as an e-voting system comes into existence. Blockchain-enabled voting systems were proposed as the next generation of modern electronic voting systems because the immutable feature of the blockchain has made it a decentralized distributed ballot box [3].

2 Concept of Blockchain

Election is a very major symbol of democracy activities but still a large portion of people in the world do not keep faith in their election system. Many countries are still using a centralized system for the election that can cause some discrepancies. Blockchain technology is one of the solutions, because it strains a decentralized system and the entire database system is owned by many users. The blockchain concept was initially proposed by Haber and Stornetta in 1991. The main purpose of designing this technology was to avoid tampering with documents. The first system based on blockchain is believed to be developed by Satoshi Nakamoto in 2008 [4]. Bitcoin is recognized as the first application of blockchain technology to create a currency that could be transacted among the related parties over the internet based on the cryptographic method to secure the transactions.

Blockchain is a database of records which is distributed on the network, or we can say, it is a public ledger of all transactions that have been executed and are shared between every user on that blockchain network. A blockchain is a list of records which keeps on growing as new data is added, called blocks, which are linked using cryptographic algorithms. Each block contains a cryptographic hash value of the previous block, a timestamp, transaction data, and a hash for its own complete block, making it like a linked chain. A hash is basically a unique code which is given to every block in the chain [5].

The blockchain is a distributed ledger which means it is completely open to anyone. It has an interesting property i.e. When some data is recorded inside a blockchain, it becomes very difficult to change it. If any changes are done inside the block it will lead to a change in the hash of the block. So blockchain secures itself by being distributed. Instead of using a central entity to manage the system, blockchain use Peer to Peer (P2P) networks where everyone is allowed to join. Here each node plays an important role and its main aim is to check whether data is tampered or not, by offering such a secure network Blockchain surely aims to improve the E-voting system.

2.1 Working of Blockchain

Whenever a block has some new data to store, it is added at the end of the blockchain. To add a new block to the blockchain, four things must take place [6]:

1. A Transaction or a particular event takes place which is being recorded over the blockchain network.
2. After a particular event occurs for example, a vote made by a person, this event is verified and in blockchain, the process of verification is done by the network of computers. These networks often consist of thousands or more computers spread across the globe. When an event occurs, that network of computers rushes to check whether that particular event happened in a way it is supposed to be done (Fig. 1).
3. After the event (vote making) has been verified as accurate, and complete, it goes to the next step. The Voter’s ID, name, name of the party which is being

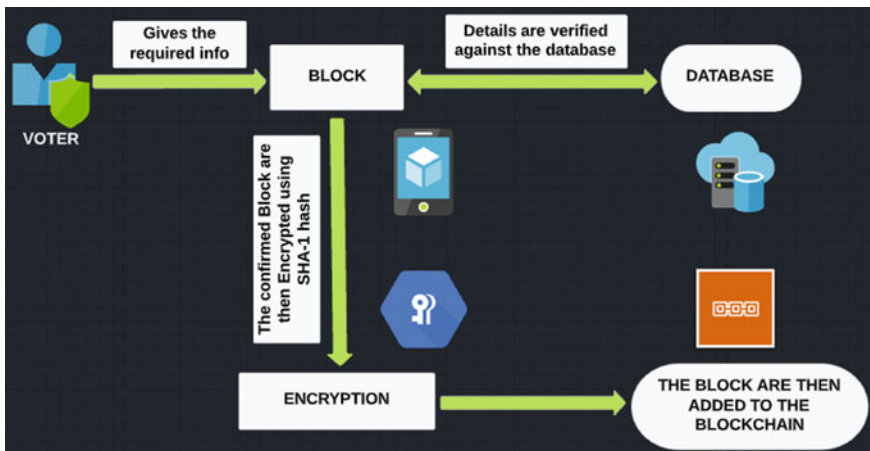


Fig. 1 Working of blockchain

voted etc., are stored in a block. Here, this particular data is likely to be joining hundreds, or thousands, of other data like itself.

4. The block must be given a hash. Once all the data of a block has been verified, it must be given a unique, identifying code called a hash. The block is also given the hash of the previous block that was the most recent block added to the blockchain. Once the hash value is generated, the block can be added to the blockchain. The block becomes publicly available for anyone to view, as soon as, it is added to the blockchain.

2.2 Characteristics of Blockchain

Blockchain has introduced a whole new technique to combat security threats, and it is the sole solution to today's security challenges. An electronic voting system must be user-friendly to each and every qualified voter, while also ensuring a high degree of security. However, ensuring the security of digital voting is always a challenge to the Voting systems [7]. However, this E-voting method is not without flaws, it is exposed to a variety of security risks and issues [8]. Blockchain has provided several new features to help with the problems and to make the technology stand out of the crowd (Fig. 2).

- (1) **Decentralization:** A database system that allows anybody connected to the network to access it. Votes are accurately, permanently, securely, and openly recorded. Furthermore, blockchain ensures that the participant's identity is

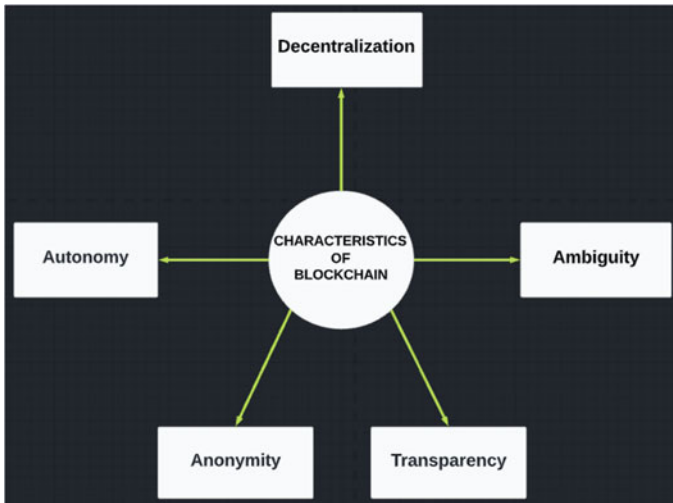


Fig. 2 Characteristic of blockchain

protected while yet allowing for public scrutiny. As a result, no one may alter or change votes [9, 10].

- (2) **Ambiguity:** Blockchain has the potential to minimise ambiguity. For example, In the 2017 Virginia House of Delegates election, the winner was picked from a pot of paper votes. One vote wasn't counted at first because the voter made a mistake by putting several ambiguous markings. Such ambiguity is less likely to lead to confusion with BEV [9, 10].
- (3) **Transparency:** Online voting had been used in 23 nations as of 2017. Some voters may be perplexed by current online voting procedures. It's difficult to tell whether the vote was cast in the intended manner or whether it was counted as cast [9]. Some electronic and online voting security mechanisms may have been created decades ago and are still in use today, unsecured against tampering. Blockchains' transparent nature might undoubtedly prevent data from being tampered with or stolen [10].
- (4) **Anonymity:** Since data is sent from node to node, the individual's identity remains anonymous, resulting in a more stable and trustworthy system [10]. This might motivate more people to vote.
- (5) **Autonomy:** The blockchain system is unique and autonomous, which means that any node on the network may safely access, store, and update data, making it reliable and free of external interference [10].

3 Existing Methodology

Voting systems currently in use in the current electoral system face significant technical challenges. Every voter must be registered to vote before the election, according to the current voting system. Their data must be stored in a secure, digitally protected format on a website, with the identity information kept private.

3.1 Traditional Ballot Paper System

Until the 1990s, India used paper ballots. As a result, in the ballot paper system, every voter must go to the polls and vote according to their motives. The results of the voting process are announced after the votes have been counted in person. There are several vulnerabilities in this system. As a result, it requires all voters to vote, and if someone does not vote, the situation may become enraged. Calculating the number of specific votes in an overcrowded country is also time consuming, costly, and difficult. The ballot paper system is very irresponsible because it replaces ballot boxes with duplicate paper, damages the ballot paper, and allows multiple people to vote.

3.2 *EVM*

The Electronic Voting Machine (EVM) is a mainstay in the electoral process, replacing the ballot box. The first time EVMs had been used was in a general election in Kerala in May 1982; however, due to the lack of a specific law mandating their use, the Supreme Court refuted that election. In 1999, the electronic voting machines (EVMs) were used for the first time in a general election (statewide) to the Goa assembly. Encouraged by the use of EVMs in all by-elections and state elections in 2003, the election commission decided to use EVMs in all 543 Parliamentary Constituencies in the country for the Lok Sabha elections in 2004 [11]. EVM, which comprises of 2 components, was introduced to overcome the problems associated with ballot paper systems:

1. **Control Unit:** It stores and assembles votes, used by poll operators.
2. **Ballot Unit:** It is placed in the polling place and is used by voters to cast their ballots.

Votes are properly recorded using EVMs, and none of the problems associated with them, such as calculation, measurement, accuracy, prompt declaration of effects, and system robustness, have become noticeable. However, the serious problem lies in the verification of authenticity; the person voting may not be present as a legitimate individual. Other issues such as political parties holding booths, voting for the elderly, as well as fraudulent voting all seem to be possible.

3.3 *E-Voting*

Electronic voting has become very popular in place of the ballot paper system since the late 1990s/early 2000s. Electronic voting is widely used, and the majority of applications are extensively tested and used on a limited basis. Despite concerns about audits and authenticity, electronic voting remains popular. Furthermore, in democratic societies, the most essential element is a strong electoral process that is transparent and confidential. It does, however, have some disadvantages. This results in a loss of privacy and makes calculating votes more difficult. Elections may be jeopardised by automatic vote buying and internal attacks on the voting system.

3.4 *Blockchain a Better Technology for E-Voting*

Blockchain is a digital ledger in its most primitive sense. To verify, process, and record all transactions across the system, the technology engages into the power of its peers or nodes on the network. This ledger is never stored; instead, it is kept on the “chain,” which is supported by millions of nodes at once. Blockchain’s transaction

database is incorruptible, and each record is easily verified, thanks to encryption and decentralisation. Because the network does not exist in a single location, it cannot be taken down or influenced by a single party.

Not only can blockchain be used for financial transactions, but it can also be used for any type of data transmission. This type of system infrastructure is extremely beneficial for voting because a vote is a small piece of high-value data. Due to necessity, modern voting systems are largely stuck in the twentieth century, and those who want to vote must leave their homes and submit paper ballots to a local authority. As a result, some of them attempted to implement this system on an internet platform, but the results were unsatisfactory due to major security flaws.

With blockchain, the various issues identified in these early attempts at online voting can be resolved. A blockchain-based voting application does not need to be concerned about the security of its Internet connection because any hacker with access to the terminal will be unable to affect other nodes. Voters can cast their ballots effectively without revealing their identities or political sentiments to the public. Because each ID can be linked to a single vote, no fakes can be made, and tampering is impossible, officials can count votes with absolute confidence [12].

A Genuine Democratic Republic: Blockchain is paving the way for direct democracy, in which citizens make their own policy decisions rather than relying on elected officials. While the rules of a political election may need to be changed to accommodate such a transparent system, blockchain can also be used to guide financial decisions, general meetings, polling, and surveys, among other things [13].

4 Literature Review

We shall look at numerous research articles and theses that looked into comparable subjects of study, such as Blockchain based electronic voting systems:

1. Kshetri, Nir and Voas, J. (2018). “*Blockchain-Enabled E-voting*” research paper, proposed the importance of BEV in context to reduce the security issues and it also highlighted various BEV implementations alongside giving a survey of Blockchain-based solutions deployed for voting at the community, city, and national levels [14].

Residents of Moscow began voting on a blockchain in December 2017, and the results were publicly auditable [15]. Neighbours should be able to influence their living circumstances in a convenient atmosphere, according to city officials. The officials also thought that a blockchain would improve citizen-government confidence [16]. Each community-discussed question was transferred to BEV. The results were released when the polling was completed [17].

The Ddabok Community Support Project was voted on using a BEV system in March 2017 in the South Korean province of Gyeonggi-do [18]. A blockchain platform established by the Korean financial technology firm Block that featured smart contracts was used to vote by 9,000 citizens. A blockchain was used to store

the votes, results, and other pertinent data. This approach involves no management or central authority [19]. This was the first time a technique like this was used in South Korea.

Estonian individuals and e-residents who own shares in the LVH Group, an Estonian technology business, may now utilise BEV to make corporate governance choices [20]. They can vote at LVH's annual general meeting by logging up with their verified national online ID. The e-residency platform in Estonia verifies e-resident shareholders [21]. Estonia intends to use blockchains in a variety of fields, including an e-residency programme (which permits foreign nationals to open businesses in Estonia) and healthcare (securing health data storage and allowing real-time monitoring of patient conditions).

Agora, a Swiss blockchain firm, gave a partial count of election results in Sierra Leone's general elections in March 2018 [22]. Agora was one of the approved observers who offered a comparable impartial count. Sierra Leone's elections were regarded by Agora as a "use case" rather than a "complete deployment" of BEV.

As a result, we carried out a detailed blockchain-based E-voting system survey, allowing us to better understand how far Blockchain's roots have scattered and how critical its implementation in various sectors has been.

2. The IRJET Journals published a research paper on "*Blockchain-Based Secured E-Voting System to Remove the Opacity and Ensure the Clarity of Election of Developing Countries*", in which they proposed an e-voting system based on blockchain technology that meets the inevitable e-voting properties while also providing a degree of decentralisation and putting as much control of the process in the hands of the voters as possible.

The Architecture was made in such a way that it meets the requirements like system integrity, data integrity, reliability, data confidentiality, and voter anonymity. The proposed architecture mainly consisted of four parts [23]:

- Voter Registration.
- Candidate Registration.
- Vote Casting Procedure.
- Result Publication (Fig. 3).

Because cell phones and smart cards are required in the given proposed system, the need for security to protect voter information is the primary concern. It utilizes a method called El Gamal Cryptosystem, which can produce a pair of keys as well as encrypt and decode data. It also employs the hash function (SHA-256) because it's one of the most important aspects of blockchain.

Despite the fact that the blockchain-based e-voting system is safer and more transparent, there were some drawbacks in the proposed system:

Since the entire voting process has been digitized, it is nearly difficult for people to cast votes if they do not have access to the internet.

The system may run slowly at times owing to workload because it is a highly secure and busy procedure.

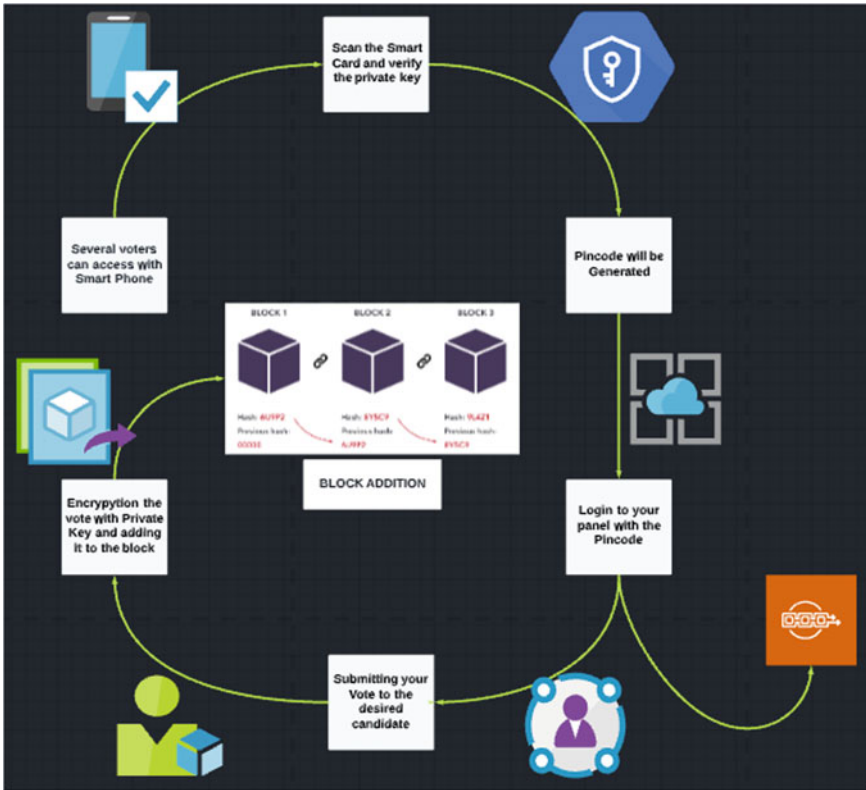


Fig. 3 Process of adding blocks in blockchain

Since the majority of people in impoverished nations have low literacy rates and little awareness of technology, it may be difficult for them to cooperate with this system.

3. Adida, B., Helios (2008). “Web-based open-audit voting.”, in Proceedings of the 17th Conference on Security Symposium, ser. SS’08. Berkeley, CA, USA: USENIX Association, 2008.

This paper proposes associated justifications for an adequate security model as well as comprehensibility criteria [24]. It also describes a web ballot theme, pretty understandable Democracy, and shows that it meets the adequate security model while being much more understandable than Pretty smart Democracy, which is currently the only theme that also meets the planned security model [25].

4. Chaum, D., Essex, A., Carback, R., Clark, J., Popoveniuc, S., Sherman, A. and Vora, P. (2008). “Scantegrity: End-to-end voter-variable optical- scan voting.”, *IEEE Security Privacy*, vol. 6, no. 3, pp. 40–46, May 2008.

Scantegrity is the first independent E2E verification mechanism that keeps optical scan as the underlying voting system and does not interfere with a human recount [26].

5. Dalia, K., Ben, R., Peter Y. A, and Feng, H. (2012). “*A fair and robust voting system by broadcast.*”, *5th International Conference on E-voting, 2012.*

This paper proposes a recovery round to allow for the announcement of the election result if voters abort, as well as a commitment round to ensure fairness. Additionally, it provided a computational proof of ballot secrecy security [27].

Secure digital identity management is one of the most recent major technical challenges relating to e-voting systems, but it is not the only one. Before the elections, any potential voter should register with the electoral system. Their information should be in a format that can be processed digitally. Furthermore, their personal information should be kept private at all times. The following issues could arise if the old E-voting system is used:

- Anonymous vote-casting.
- Individualized ballot processes.
- Ballot casting verifiability by (and only by) the voter.
- High initial setup costs.
- Increasing security problems.
- Lack of transparency and trust.
- Voting delays or inefficiencies related to remote/absentee voting.

Limitations of Existing system:

- (1) *Anonymous vote-casting:* Each vote, which may or may not include a choice for each candidate, should be anonymous to everyone, including system administrators, once submitted through the system.
- (2) *Individualized ballot processes:* The manner in which votes are represented in web applications or databases is still up for debate. A hashed token, on the other hand, is more likely to provide uncertainties and integrity than a transparent text message. Meanwhile, the vote should be untrustworthy, as the token resolution cannot attach it.
- (3) *Ballot casting verifiability by (and only by) the voter:* When an elector submits a vote, he or she should be prepared to see and verify his or her own vote. This is important to understand in order to prevent, or at the very least detect, any potential malicious activity. This justification, aside from providing pre signals, can certainly increase voters’ feelings of trust. In some recent applications, these issues are partially self-addressed. However, evidence suggests that e-voting is currently in use in a number of countries, including Brazil, the United Kingdom, Japan, and the Republic of Estonia. The Republic of Estonia should indeed be evaluated differently than the others because they provide a complete e-voting solution that is said to be equivalent to traditional paper-based elections.
- (4) *High initial setup costs:* While maintaining and operating online voting systems is much less expensive than traditional elections, initial deployments can be costly, especially for businesses.

(5) *Increasing security problems:* Cyber-attacks pose a significant threat to public opinion polls. If a degree hacking attempt succeeds during an election, no one wants to take responsibility. DDoS attacks have been documented, but this is not the case in the elections. The United States' citizen integrity commission recently filed an affidavit regarding the state of the elections in the North American country. As a result, Ronald Rivest expresses that "hackers have a wide range of ways to assault pick machines." For instance, in the hacking method, barcodes on ballots and smartphones in specific locations may be used. Apple has consistently stated that we must not overlook the fact that computers can be hacked and that evidence will be erased. Double voting and voters from opposing regions are also common problems.

To counteract these dangers, software mechanisms that provide the following benefits should be implemented:

- Avoiding the erasure of evidence.
- Transparency while maintaining privacy.

Lack of transparency and trust: When everything is done online, how can people trust the results? Perceptual issues need to be addressed.

Delays or inefficiencies in voting due to remote voting: In voting schemes, timing is critical; technical capabilities and infrastructures must be reliable and operate at peak performance in order for distant voting to be synchronous.

5 Proposed System

Working Process

We have presented our proposed methodology with a user-friendly interface, as well as implementation of the blockchain idea at the back-end for the security of the voting process, in order to make the voting process easier and more efficient. The working process of the proposed system is mentioned in the following steps (Fig. 4).

Step 1: As the candidate would open the website, he would be directed to the login page, where he would put his credentials and would go to the main page of the website.

Step 2: On the main page the user would be briefed about all the information regarding the elections held and also about the candidates and their progress through the dashboard.

Step 3: Through the website the voter would register himself for the current elections by putting all the necessary details and by creating a password, and once the process is done the voter would get registered and a unique voter id would be generated for that particular user.

Step 4: A specific transaction representing the Candidate will be the first transaction added to the block. This transaction serves as a starting point and will not be counted as a vote. The user's information will be cross-referenced against the

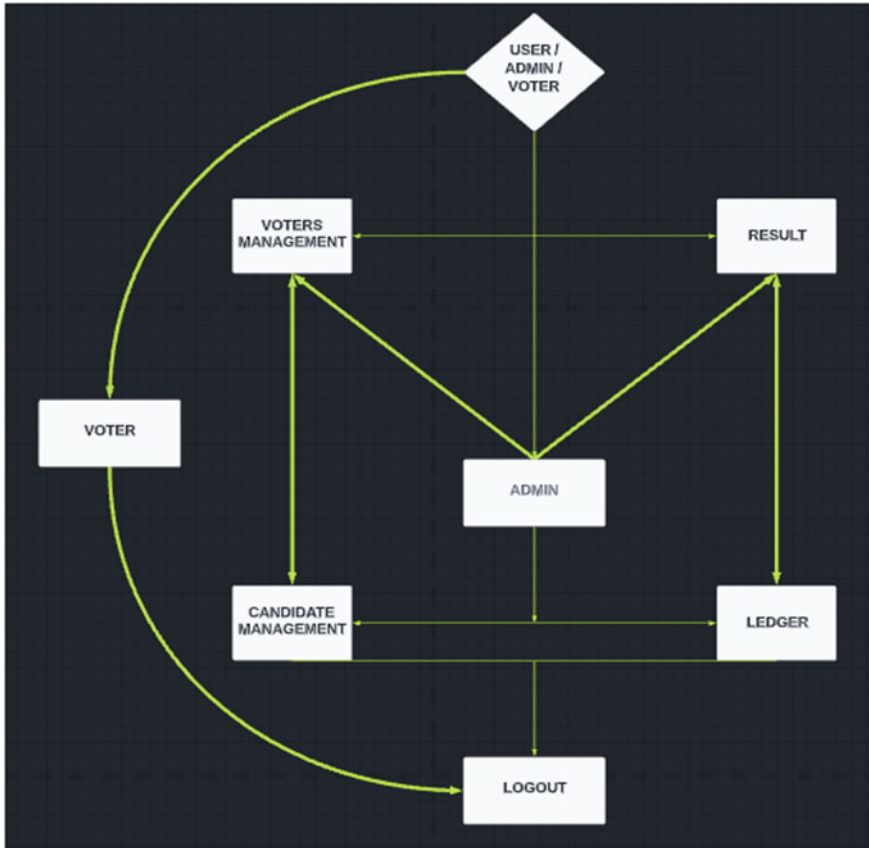


Fig. 4 System architecture

government database in the previous step. This is the final phase in the verification process. The procedure will not be able to proceed without verification.

Step 5: After the registration process is completed, the user would go to the vote casting website where he would cast a vote by putting his username that would be the unique voter id generated and the password that the voter has set, after putting all the details the voter would be directed to the vote casting section.

Step 6: In the vote-casting section, the voter will get the information about the candidates that are currently being selected for the election, the voter will vote for his desired candidate through the vote casting prompt and then the voting process will be completed. The candidate would not be able to change his vote and tamper the data once the vote is casted.

Step 7: After the voting procedure is done, the votes will be encrypted using the SHA-1 one-way hash algorithm to protect them from outside influence. This also eliminates any manipulation of the user’s votes. The block will be given a timestamp and transmitted to the next checkpoint, which will be considered as a pointer for the

last made update to the blockchain, as soon as the user's block is encrypted. At the next checkpoint in the procedure, this timestamp will be verified. This phase entails linking this block to the node of the next checkpoint and adding it to the blockchain. This way the data would be added in the blockchain.

Hardware and Software Requirements

1. Frontend Requirements

- Code Editor
- Browser
- PHP

2. Backend Requirements

- XAMPP
- My SQL
- PHP

3. Blockchain Requirements

- Truffle framework
- Ganache
- Meta mask
- Solidity language.

6 Result and Discussion

As previously stated, the primary goal of incorporating blockchain into the E-Voting system was to achieve decentralization. Where everyone has a right and can come up with a single decision if anything changes, and where the majority of them can appeal if any issues arise. In this era of technological advancement, we must accept and adapt to change, and this is how we can take a step forward in the direction of improving our voting system. As we all know, tampering with ballots has occurred in the past, and by engaging in such behavior, the majority of people are losing interest and faith in the current voting system. To avoid such tampering and to rekindle public interest in voting, a Blockchain-based E-voting system is the best solution for today's era, and now is the best time to implement one. Transparency and security are maintained with the help of blockchain technology. As we all know, the government spends a lot of money to hold elections, and with the help of technology, we can invest the money to greater use in different sectors for development. It has a wide range of applications, including corporate elections and opinion polls. So, the main goal would be and see if it has the potential to grow to the point where real-world problems can be addressed without resorting to the current voting system. As a result, the world requires a model in which optimal solutions are generated and improvements in their implementation can be applied by both the public and private sectors. We are also aware that we live in a country with a much higher percentage of youth than any other country on the

planet. As a result, focusing on such a central issue can put our country ahead in terms of development. Because we are a diverse country, people from all walks of life can come together and contribute to such a cause (Figs. 5, 6, 7, 8, 9, 10, 11, 12 and 13).

This is Admin Login page where it will lead to main page.

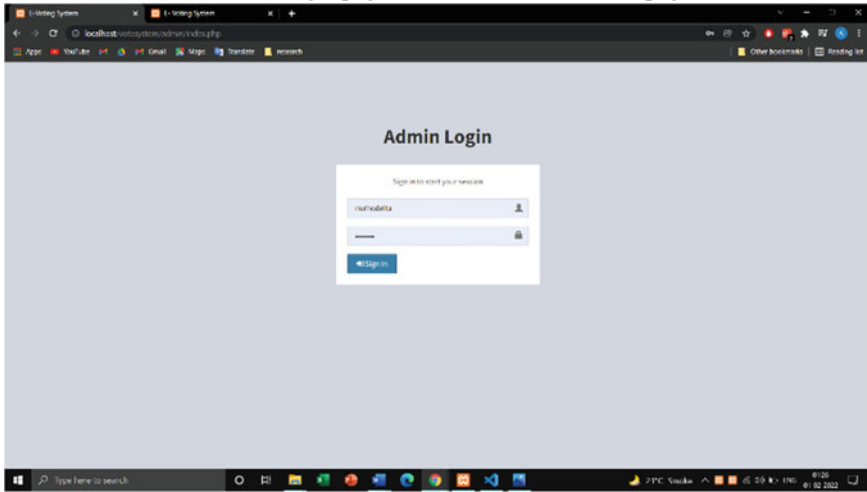


Fig. 5 Admin login page

This Page consists of Dashboard

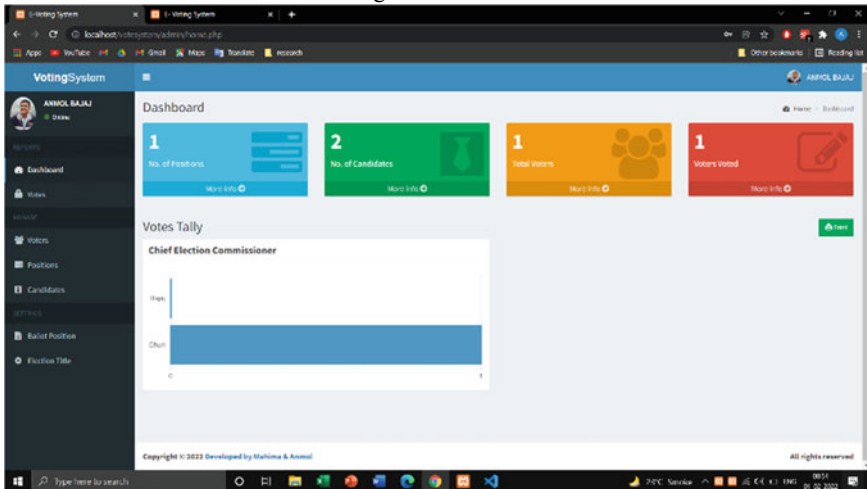


Fig. 6 Dashboard

This page shows the number of voters had registered.

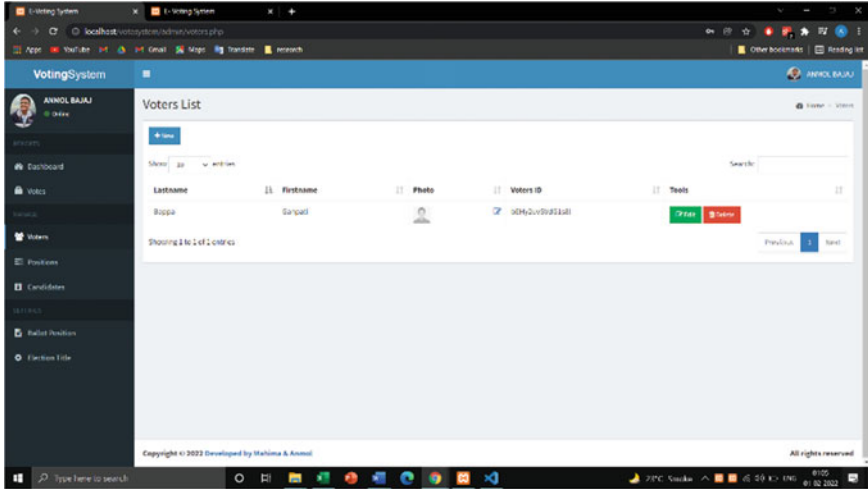


Fig. 7 Voters list

This Page is all about the Position for which election is going to occur.

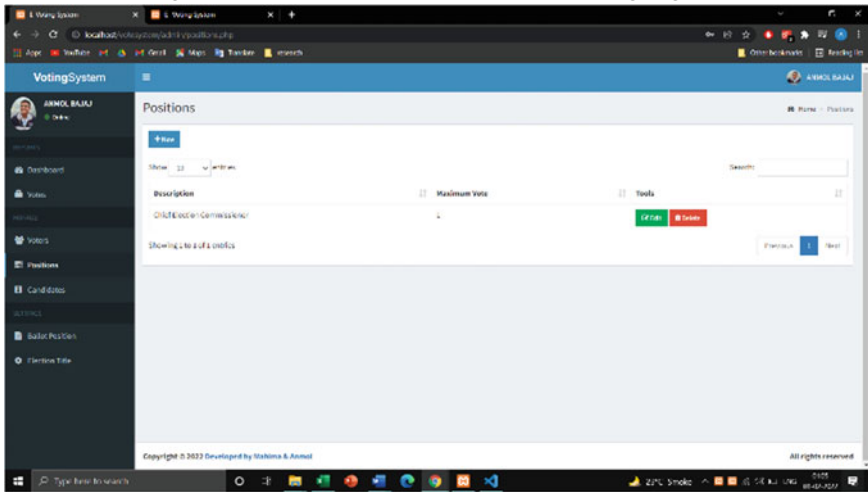


Fig. 8 Position

7 Conclusion

Businesses all across the world are commenting on the technology’s possibilities and where it will go in the upcoming years. Blockchain, which is now a trendy topic, promises to improve the accuracy, efficiency, and security of business, government

This page incorporates about Candidate who are appointed for the required position.

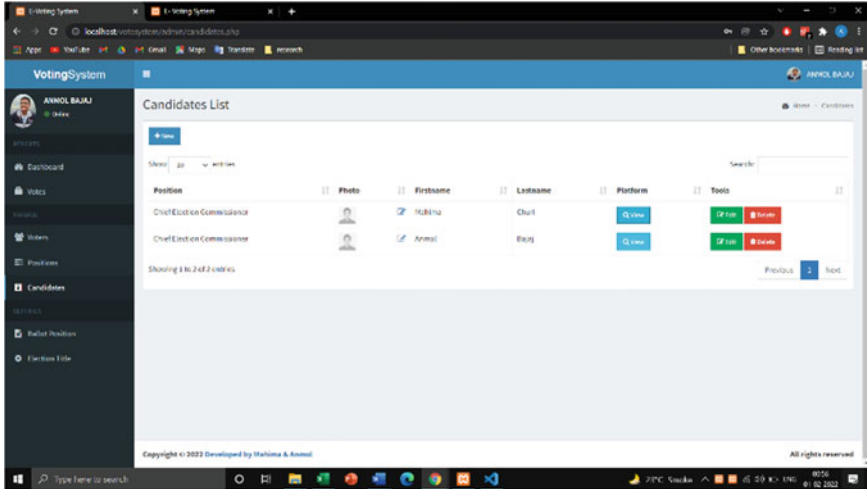


Fig. 9 Candidate list

This page is for voters login in which have to join with their Unique Voter ID and Password.

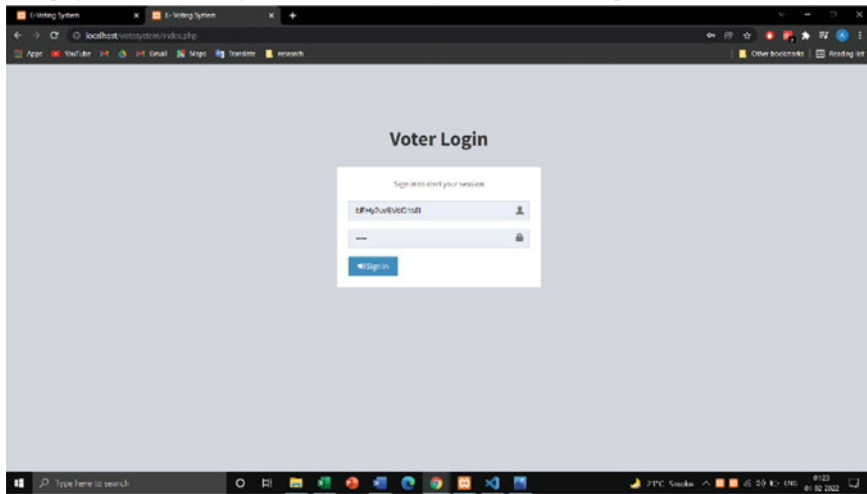


Fig. 10 Voters login

activities, and organizational activities, with its fundamental properties of persistency, security, decentralization, and integrity. Blockchains have the potential to impact established industries and institutions. Traditional administrative systems must be replaced with newer and more advanced technology, such as Blockchain,

This page is about to cast a ballot of their own choice

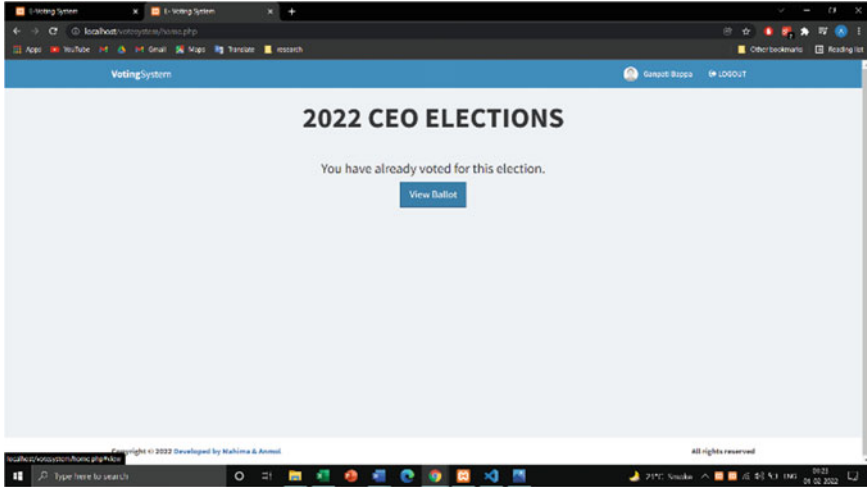


Fig. 11 Vote casting

The vote has been casted.

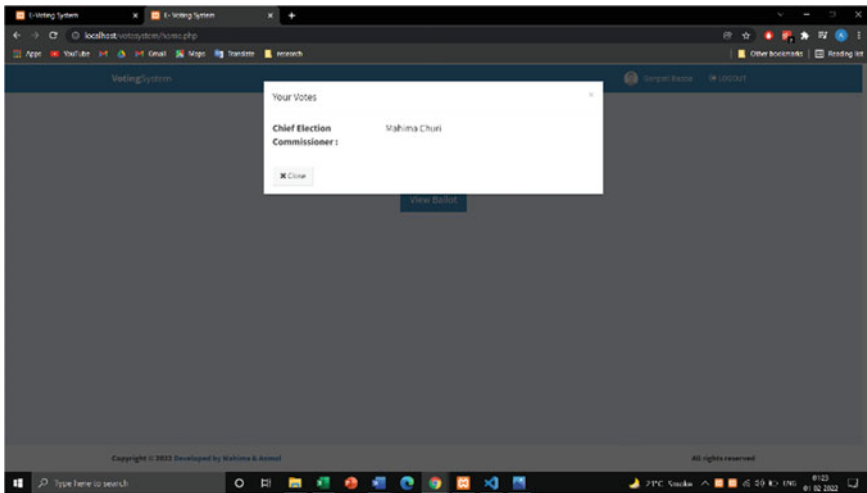


Fig. 12 Casting of vote prompt

given the present rate of growth that our country is experiencing. A comprehensive description of the blockchain idea was provided in this paper. An overview of blockchain technology, including how they function and their many characteristics, were also mentioned. We then examined the many conventional voting techniques used in our nation and how blockchain is the most efficient and successful of all

The casted ballot has been submitted in the admin page.

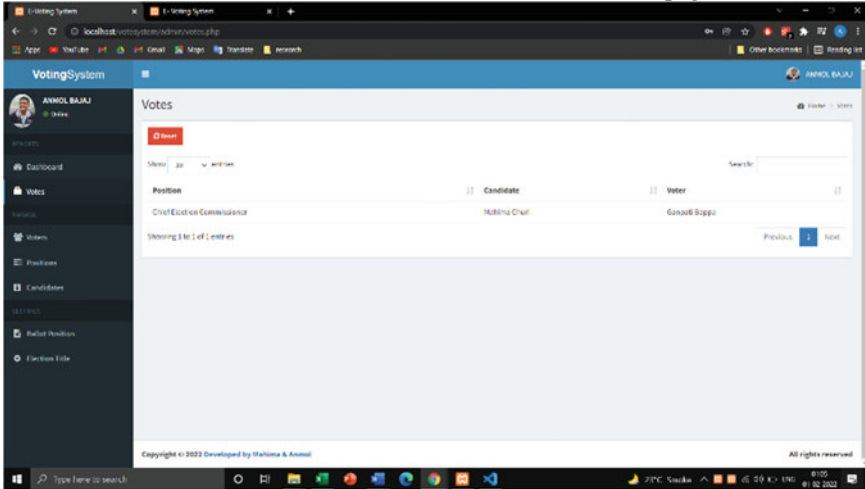


Fig. 13 Viewing casted ballot

of them. A few literature surveys showing the integration of blockchain in enterprises and organizations were also included. The uses, benefits, and drawbacks of Blockchain were also highlighted. As a conclusion, we propose a Blockchain-based E-Voting system that is more trustworthy, secure, and labor-saving. Such mechanisms will increase openness in Indian democracy, and we may anticipate our country to rank first in the World Democratic Index as a result of this system, as well as improved governance.

Acknowledgements Our special thanks to the research head and our guide Dr. Megharani Patil who have contributed toward development of the paper. Also, our sincere thanks to our principal Dr. B. K. Mishra for encouraging us to write this paper.

References

1. Article-systematic review of challenges and opportunities of blockchain for E-voting, pp 4–24
2. Ryan P, A, Schneider S, Teague V (2015) End-to-end verifiability in voting systems, from theory to practice. *IEEE Secur Priv* 13:59–62 [CrossRef]
3. Zhang S, Wang L, Xiong H (2020) Chaintegrity: blockchain-enabled large-scale e-voting system with robustness and universal verifiability. *Int J Inf Secur* 19:323–341 [CrossRef]
4. Nakamoto S (2019) Bitcoin: a peer-to-peer electronic cash system. <https://bitcoin.org/bitcoin.pdf>. Accessed 27 Aug 2019
5. <https://blockgeeks.com/guides/what-ishaing/>. Accessed 2 Dec 2021
6. <https://www.investopedia.com/terms/b/blockchain.asp/>. Accessed 1 Dec 2021
7. Srivastav G, Dwivedi A, Singh R (2018) Crypto democracy: a decentralized voting scheme using blockchain technology. In: Proceedings of the 15th international joint conference on

- e-business and telecommunications (ICETE 2018), SECRIPT, vol 2, pp 508–513. <https://doi.org/10.5220/0006881905080513>
8. Chaeikar SS, Jafari M, Taherdoost H, Chaeikar NS (2012) Definitions and criteria of CIA security triangle in electronic voting system. *Int J Adv Comput Sci Inf Technol (IJACSIT)* 1(1):14–24. ISSN 2296-1739
 9. Patil HV, Rathi KG, Tribhuwan MV (2018) A study on decentralized E-voting system using blockchain technology. *Int Res J Eng Technol (IRJET)* 05(11)
 10. Siyal AA, Junejo AZ, Zawish M, Ahmed K, Khalil A, Soursou G (2018) Applications of blockchain technology in medicine and healthcare: challenges and future perspectives. Received 6 Nov 2018. Accepted 26 Dec 2018. Published 2 Jan 2019
 11. <https://eci.gov.in/voter/history-of-evm/>
 12. <https://www.newindianexpress.com/opinions/2021/apr/26/from-evms-to-blockchain-based-e-voting-2294834.html>
 13. <https://www.investopedia.com/news/how-blockchain-technology-can-prevent-voter-fraud/>
 14. Kshetri N, Voas J (2018) Blockchain-enabled E-voting. *IEEE Softw* 35(4):95–99. Hochstein M (2018) Moscow’s blockchain voting platform adds service for high-rise neighbors. *CoinDesk*, 15 Mar 2018
 15. <https://www.coindesk.com/moscows-blockchain-votingplatform-adds-service-for-high-rise-neighbors>
 16. Holder S (2018) Can the Blockchain Tame Moscow’s wild politics? *CityLab*, 22 Dec 2017. <https://www.citylab.com/life/2017/12/can-the-blockchain-tame-moscows-wild-politics/547973>
 17. Digital home Blockchain voting system, active citizen in Moscow opens. *BitcoinExchangeGuide.com*. <https://bitcoinexchangeguide.com/digital-home-blockchain-voting-system-active-citizen-in-moscow-opens>
 18. A South Korean Province used Blockchain tech for resident voting. *CCN.com*, 8 Mar 2017. <https://www.ccn.com/south-korean-province-used-blockchain-tech-resident-voting>
 19. South Korea uses Blockchain technology for elections. *Krypto Money*, 2 May 2017. <https://kryptomoney.com/south-korea-uses-blockchain-technology-for-elections>
 20. Waterman S (2017) Nasdaq says Estonia E-voting pilot successful. *Cyber-Scoop*, 25 Jan 2017. <https://www.cyberscoop.com/nasdaq-estonia-evoting-pilot>
 21. Aasmae K (2016) Why Ripples from this Estonian blockchain experiment may be felt around the world. *ZDNet*, 14 Apr 2016. <https://www.zdnet.com/article/why-ripples-from-thisestonian-blockchain-experiment-may-be-felt-around-the-world>
 22. Kimathi B (2018) Why you shouldn’t get carried away by Sierra Leone’s blockchain elections. *Crypto-Lines*, 13 Mar 2018. <https://cryptolines.com/2018/03/13/blockchainelections>
 23. Finnan D (2018) Sierra Leone’s electoral commission distances itself from use of Blockchain during polls. *RFI*, 18 Mar 2018. <http://en.rfi.fr/africa/20180319-sierra-leones-electoralcommission-distances-itself-use-blockchain-during-polls>
 24. IRJET-blockchain-based secured E-voting system to remove the opacity and ensure the clarity of election of developing countries. *IRJET J. Academia.edu*, Jan 2020
 25. ITM web of conference. Volume 32. In: 2020 international conference on automation, computing and communication 2020 (ICACC-2020). Blockchain technology based e-voting system
 26. Adida B (2008) Helios: web-based open-audit voting. In: *Proceedings of the 17th conference on security symposium, ser. SS’08*. USENIX Association, Berkeley, CA, USA
 27. Chaum D, Essex A, Carback R, Clark J, Popoveniuc S, Sherman A, Vora P (2008) Scantegrity: end-to-end voter-verifiable optical- scan voting. *IEEE Secur Priv* 6(3):40–46
 28. Dalia K, Ben R, Peter YA, Feng H (2012) A fair and robust voting system by broadcast. In: *5th international conference on E-voting*

29. Wang B, Sun J, He Y, Pang D, Lu N (2018) Large-scale election based on blockchain. *Procedia Comput Sci* 129:234–237
30. Venkatapur RB, Prabhu B, Navya A, Roopini R, Niranjana SA (2018) Electronic voting machine based on blockchain technology and Aadhar verification. *Int J Innov Eng Sci* 3:12–15
31. Ayed AB (2017) A conceptual secure blockchain-based electronic voting system. *Int J Netw Secur Appl (IJNSA)* 9(3)

An Intelligent Voice Assistant Engineered to Assist the Visually Impaired



Rishabh Chopda , Aayan Khan , Anuj Goenka , Dakshal Dhere, and Shiwani Gupta

Abstract Visually handicapped people's lives are subject to a multitude of unrelenting challenges because they've been made bereft of the gift of sight. The proposed solution is a wearable Smart Voice Assistant that is developed to accommodate the needs of the visually impaired to aid them in every aspect of their everyday lives. It takes advantage of recent breakthroughs in the fields of language processing and computer vision to provide a broad spectrum of applications, including emergency response functionality, object recognition, and optical character recognition. It comprises hardware components that provide feedback in the form of sound, haptics, and speech to help with obstacle avoidance. The voice assistant also interacts with a smartphone application to enhance the user's experience by enabling them to read the messages from their phone, send an SOS message to their closest connections in an emergency, customize the device settings through the mobile application, and find the device with the press of a button if it is misplaced. The proposed solution will enable the user to live a life in relative safety and comfort, which is essential for people suffering from varying levels of visual impairment.

Keywords Voice assistant · SOS · Object avoidance · Object recognition · Optical character recognition

1 Introduction

As we approach a stage in human civilization where the average age of the population is increasing at an unprecedented rate, human physical functions are failing, and that visual faculties are declining at a behooving rate. Globally, World Health Organization (WHO) [1] that 43 million people are visually disabled, with another 295 million suffering mild to severe vision impairment. It was essential to create a gadget that assists them in traversing their environment and empowering them to do tasks that would otherwise be difficult or simply impossible.

R. Chopda (✉) · A. Khan · A. Goenka · D. Dhere · S. Gupta
Thakur College of Engineering & Technology, Mumbai, Maharashtra 400101, India

The condition of lacking vision is referred to as blindness, which is caused by a physiological or neurological imbalance. Despite tremendous advancements in technology, blindness remains a serious problem [2, 3]. Researchers have been concentrating on this topic to produce helpful tools or aides for those who are blind or visually impaired. For blind people, very few assistive tools and devices are already available, however, the efficacy of their applications is fairly limited by speed, scope, and above everything, the manner they have been implemented in.

The previous works in the domain of assisting the blind have yielded fruition in various domains of the field. Functionalities related to OCR, fruit ripeness estimation, object detection, and identification, and navigation assist for the blind have been developed, albeit not necessarily to the point of utility, but to generate a solution to the many problems faced by the blind. S. M. Felix and the team [1] have created a mobile application that works to provide functionalities similar to the proposed system like voice assistant, OCR, and even image recognition. Safe navigation for the visually impaired [4, 5] has been worked on extensively using stereo-cameras [6] and GPS. Similarly, P. Bose [2], has developed a mobile-based application to work in assisting the blind to perform functionalities like speech recognizing and speech synthesis as the means of interaction through voice input to recognize text on a real-life object and provide audio feedback [7]. The related works [3, 8] are a raspberry pi based smart device to assist the blind by providing object identification functionality and even obstacle detection and avoidance system. This is an idea involving the aforementioned hardware and machine learning based software. The premise for common object detection and identification is of a prime importance from a safety and utility standpoint and there is an abundance of related work in the object detection using Computer Vision [9–11]. Iyear [12] has used the current technology to increase the number of visually impaired users navigating the internet like reading articles or listening to music on Youtube, etc.

Additionally, there are a variety of technologies available to help the sight impaired navigate both indoors and outdoors. All of these devices rely solely on the Global Positioning System (GPS) to determine their location to navigate your way around. In reference [13], the paper offers a system that makes use of stereo vision a sonification approach and image processing methodology Support navigation for the blind. The system that has been created includes stereo cameras as vision sensors and stereo cameras as wearable computers. All of the earbuds are fashioned into a helmet.

To summarize, when considering the challenges of the visually handicapped, earlier attempts to solve this problem have focused on solving problems that have a limited range. Previous approaches have tended to solve only one key issue while leaving the others unaddressed. We hope to provide a one-stop solution to all the primary challenges that the visually impaired face with the proposed system. The proposed system not only assists the blind in walking by avoiding obstacles, but also allows them to read the newspaper, determine the maturity of fruits, and ask for assistance in an emergency. It is a significant improvement over prior art in that it addresses problems that were previously unresolved, provides a better user experience, and happens to be multifaceted.

2 The Proposed System

This paper describes a smart wearable voice assistant that leverages machine learning and deep learning to help blind individuals identify obstacles to help them in walking, also providing them with other recognition functionalities like facial and text recognition [14], which ultimately aims at decreasing the unfair challenges they encounter daily. When combined with Ultrasonic sensors, the device allows users to move freely without much caution as it alerts them of approaching impediments. Furthermore, the Voice Assistant will be used in combination with a companion app [15] that adds a slew of new functions to an already feature-rich device. The mobile app has a host of additional features that make the user's life easier. It uses the Voice Assistant to keep the user updated on recent activity on their phone, but it most importantly functions as an integral part of the SOS functionality, which involves sending a distress signal to the user's preferred emergency contacts.

Largely, the proposed idea can be divided into three major parts:

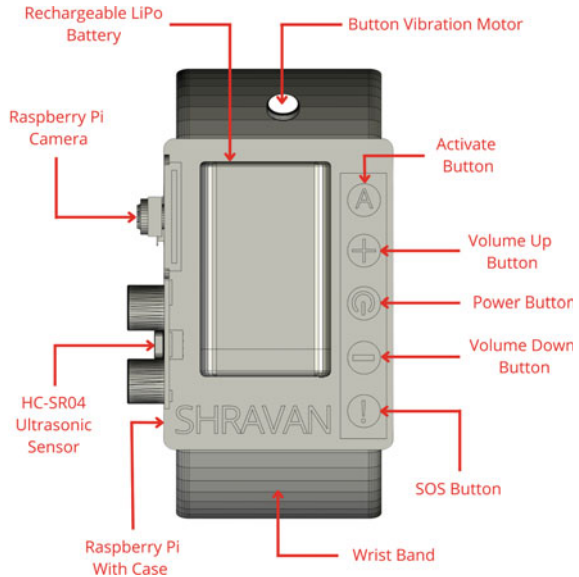
1. Wearable Wrist Device
2. Voice Assistant
3. Companion Mobile Phone Application.

2.1 *Wearable Wrist Device*

The proposed wearable wrist device is tailored to be as user-friendly as possible. The buttons are etched with symbols to make them distinguishable when felt by the user's fingertips. The device is engineered to handle all the rudimentary tasks of an individual's day-to-day life efficiently. Figure 1 shows the diagram of the proposed device that includes an embedded system module (Raspberry Pi 4 Model B) for handling all the computational tasks, an ultrasonic sensor that measures the distance to an object using ultrasonic sound waves wherein the user is alerted if an object lies within 40cms of the sensor's radius, a rechargeable LiPo battery to power the device, a Raspberry Pi camera for accommodating all the functionalities involving computer vision, five buttons with a distinct set of functionalities and a vibration motor to provide haptic feedback. The functions of the buttons are as followed:

- The Activate button is used to activate and deactivate the voice assistant.
- The Volume Up button is used to raise the volume of the voice assistant.
- The Power button is used to turn on/off the voice assistant
- The Volume Down button is used to lower the volume of the voice assistant.
- The SOS button is used to activate and deactivate the ultrasonic sensor when pressed once and activate the SOS functionality when pressed thrice.

Fig. 1 Structure for the proposed device



2.2 Voice Assistant

The wearable wrist device includes Computer Vision-based functionalities like object recognition, fruit ripeness detection, and optical character recognition. After the user asks for the function through the voice assistant, the camera captures a snapshot of the desired subject, converts it to a vector, sends it to the central processing unit where it is identified, and the result is sent back to the user in audio format.

At the user's request, the voice assistant works in conjunction with its companion mobile application to read incoming messages from the paired phone. The voice assistant can perform the following key tasks in addition to the features listed above:

- Getting live cricket match scores by scraping data from the web using BeautifulSoup which is a Python package used for parsing HTML and XML documents.
- Describing weather conditions of the user's location by fetching the user's geolocation from the phone and getting the weather data for the desired location using the OpenWeatherMap API.
- Carrying out a quick Wikipedia search based on the spoken keyword and reading out the article summary of the keyword using the Wikipedia python library which makes it easy to access and parse data from the Wikipedia website.
- Updating the user with the latest news headlines by fetching data from the News API which is a straightforward and easy-to-use REST API that returns JSON search results for current and historic news articles gathered from a multitude of sources.

The results of the modules will be read out by the voice assistant upon the user's request.

2.3 Companion Mobile Application

In the age where staying connected 24/7 has become more of an unspoken societal norm than an option, Cell phones have become a necessity for many individuals all around the world. They are becoming increasingly instrumental for a large variety of reasons.

To give the Smart Voice Assistant a new dimension, we propose integrating a Mobile Application [16, 17] whose functionalities would communicate directly with the Voice Assistant Device's features. The mobile app also facilitates composing text messages without having to interact with the phone along with a feature for the user to locate the phone. Additionally, an on-the-fly settings configurator allow users to alter the Voice Assistant's settings directly from the app. The SOS functionality [18] intends to help the user if they find themselves in any of the following situations: Component failure or traumatic emergency. When these conditions are detected, a message is sent to 4 pre-determined contacts of user making them prompt in a possible emergency.

Message enable connection with the consumer that's less intrusive in nature. To overcome this stumbling block of no vision, we've integrated the connected phone's messaging system with the Voice Assistant, allowing the user to summon the Assistant and request a message readout or compose a text message. Unfortunately, some of the information that your brain may consider unimportant may be required for you to remember where you put your keys, phone, or wallet, and if it has been erased, you will have to spend time attempting to locate some of your daily things. The feature of FindMyDevice helps in locating the device while it is not strapped to the user. Single button press, triggers the device to make sound for the user to locate it (Fig. 2).

3 Architecture and Algorithms

3.1 Optical Character Recognition

Optical character Recognition is a procedure that includes multiple sub-processes that must be completed as precisely as possible. The first subprocess in the process includes pre-processing the image for which we have used the OpenCV library. This library contains a lot of tools to help us pre-process the image with simplicity.

- Firstly, we read the image in a grayscale format which is the only format supported by the next two algorithms.

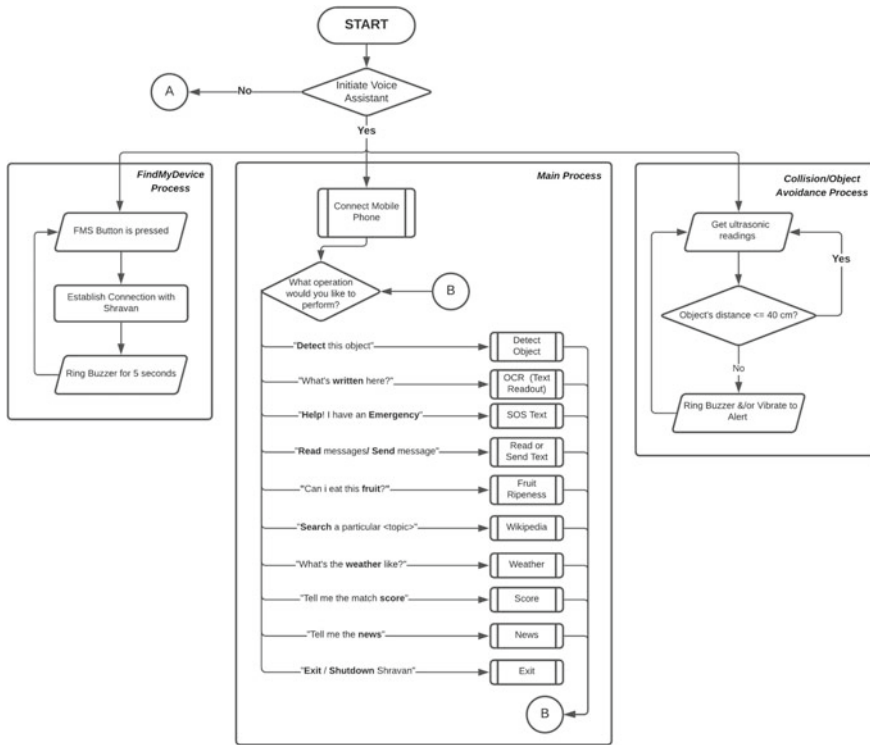


Fig. 2 Flowchart depicting the voice assistant functionality

- Secondly, we use the bilateral filter algorithm [19] which is a picture smoothing filter that preserves edges while lowering noise. It uses a weighted average of intensity data from surrounding pixels to replace the intensity of each pixel. A Gaussian distribution can be used to calculate this weight. The weights are determined not just by the Euclidean distance between pixels, but also by the radiometric differences. Sharp edges are preserved as a result.
- Then we use a thresholding algorithm that converts blacks in the images to pitch black and white snow becoming white. We mainly chose a threshold value by calculating a mean of all the pixels in the image (pixels in a grayscale image range from 0 to 255), and any pixel above that threshold gets a value of 255 which is deepest black and anything below the mean gets a value of 0 which is lightest white.

The next subprocesses include text localization, character segmentation, character recognition and post processing. Tesseract OCR [20] was chosen to do the above mentioned subprocesses. This engine uses the Long Short Term Memory (LSTM) [21] network, which is a form of Recurrent Neural Network (RNN) [22]. To use the tesseract engine for our code we use pytesseract [23] which is a wrapper class for the

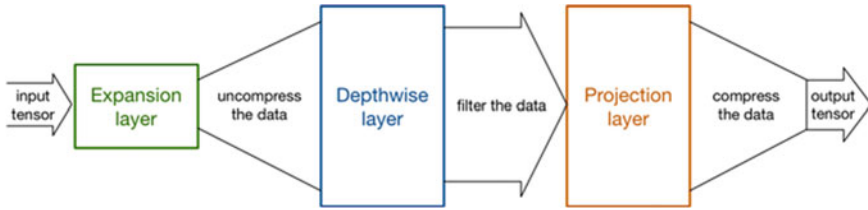


Fig. 3 MobileNetV2 architecture

Tesseract OCR Engine. Preprocessed image when passed through this library returns a text as an output using the text localization, character segmentation, character recognition and post processing subprocesses and the LSTM algorithm.

3.2 *Fruit Ripeness Detection*

The fruit ripeness detection model was trained on a dataset that was scraped from the internet using Selenium [24]. The scripts are executed by a browser-driver on a browser-instance on your device.

The first step is the data pre-processing. We will be going to use the ImageData-Generator class in Keras. Images are converted to an input shape of 224, 224 since it is the acceptable input shape for our algorithm. The images are rescaled and divided by 255 which is mainly for normalization.

The next step is building the model for which we use transfer learning [25]. This improves the learning in the new task greatly. For this purpose, we will use the tensorflow hub to load a pre-trained MobileNetv2 [26] model (Fig. 3).

To this base layer of MobileNetV2, we add our global spatial average pooling layer, a fully connected layer and a logistic layer at the end. We use ReLu activation function in all the layers except the last one which is the logistic layer or the output layer. In that layer we use the sigmoid function which returns a probability for each class between 0 and 1. With 1 being the most probable class and 0 being the least (Figs. 4 and 5).

3.3 *Object Detection*

Object detection is a computer technology that deals with finding instances of semantic items of a specific class (such as individuals, buildings, or cars) in digital photos and videos. To obtain more accuracy, computer vision models are becoming deeper and more sophisticated. However, these advancements increase the size and latency of the system, making it incompatible with systems that are computationally challenged. MobileNet comes in handy in these situations. This is a model created

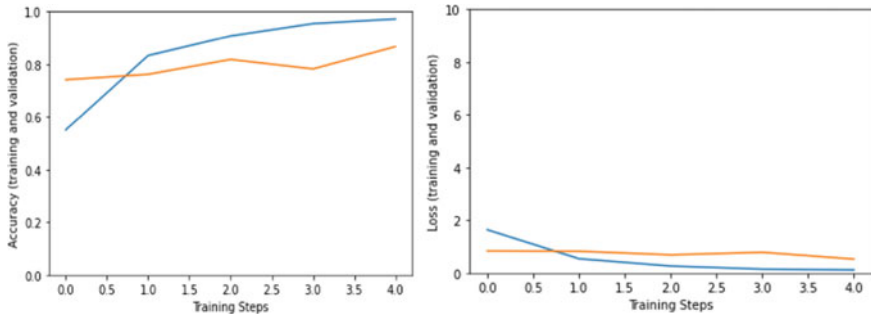


Fig. 4 Model training accuracy and loss chart

Model predictions (green: correct, red: incorrect)



Fig. 5 Result of fruit ripeness detection model

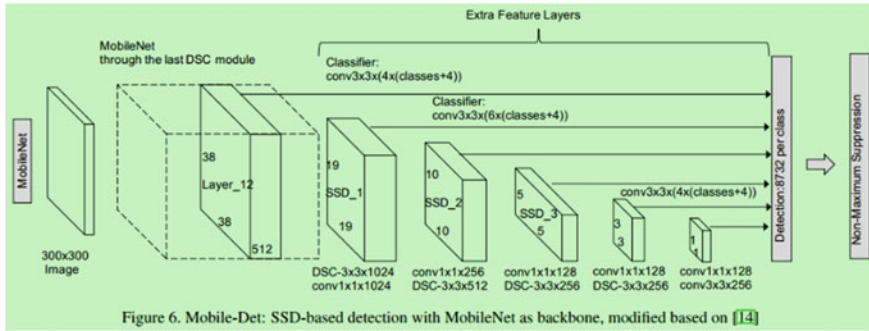


Fig. 6 MobileNetSSD architecture

primarily for high-speed mobile and embedded applications. Although it provided good frames per second on low computation, it lacked accuracy. To counteract this, we used the MobileNetSSD [27] model (Fig. 6).

MobilenetSSD is an object detection model that uses an input image to compute the bounding box and category of an object. Using Mobilenet as a backbone, this Single Shot Detector (SSD) object detection model may enable quick object detection optimised for mobile devices [28]. SSD requires only one shot to recognise many objects within an image, but RPN-based techniques such as the R-CNN [29] series require two shots, one for generating region suggestions and the other for detecting the item of each proposal. As a result, SSD is substantially faster than two-shot RPN-based techniques.

4 Result and Discussion

The proposed device will have modules ranging from object detection to voice assistant, from SOS functionality to fruit ripeness estimation and OCR. All of these would need to be implemented so that they work seamlessly with each other and be tailored to suit the needs of the blind. The voice assistant that is integrated into the device is a smart bot that is designed to answer user queries like the latest news, current weather, score of the current match etc.

The Fruit ripeness estimation works to determine whether a fruit is ripe enough or not and helps determine whether it's buyable. The machine learning module was made based on a supervised learning model. The OCR module has to be able to read printed text on hard copies in order to convert it into voice output for the blind user to listen to. Additionally, the sos functionality is an emergency alert feature that can be activated on the device. The voice assistant even enables the user to ask for weather related information from the voice assistant.

5 Conclusion and Future Scope

In this paper, we propose a smart wearable device wherein the system comprising: An Ultrasound sensor, used to detect the optical obstacles, objects and person during the walking of the person; A camera unit, used to read text using OCR, identify objects placed in front, identify faces of people; A feedback unit, used to alert the person on presence of the obstacles, articles and person in path of the person and also respond to the voice queries of the user including but not limited to; whether conditions, current time and location; A mobile application containing GPS to send information about location to the device, an SOS functionality that can be activated remotely via the Device to send current location of the user to saved contacts on the phone; and A processing unit used to process the information received from the ultrasound sensor and the mobile application, where the processing unit sends the processed information to the alert unit, and thereafter the alert unit sends the information of the obstacles, reads out written script taken from the camera, responds to user queries with computer generated voice, this forms the part of the OCR module of the proposed device. The proposed device on its own is sufficient and enough to enable the visually challenged to lead a comfortable and safe life and even provide an opportunity to go beyond hamstrung opportunities that blindness presents them with (Fig. 7).

One of the primary problems that the proposed device deals with concerns the safety [30] of the user and aims to direct help to the user in the event of an emergency. In view of that, events that involve the user stumbling or falling because an object or hurdle, for example, a raised platform, could not be detected by the proposed device, need to be considered. And should such events be fatal, the closest contacts should be immediately intimated of such occurrence right away so that their help can be directed. This paves way for a system that can detect if the user has stumbled or even fallen. This module can be called the Fall Detection module. Under the Fall Detection module, the device can identify an event and send an alert to predetermined contacts of the user that the user has fallen down and might be in need of help. To realize this module, hardware consisting of a fall detection circuit will have to be implemented. These circuits primarily consist of accelerometers that are a type of low-power radio wave technology sensor, to monitor the movements of the user. Some advanced systems may even consist of gyroscopes, infrared sensors, acoustic sensors, etc. to make the fall detection even more accurate. Once a fall is detected, the device will rely on the information to the companion mobile app of the proposed device. The mobile application can then alert the saved contacts of the user a location of the user along with a message informing them of the fall. Events like the user being involved in a car accident or any other road accident would also trigger the alert being sent. Such a functionality would add an extra level of security to the lives of the user in the case of post event damage control (Fig. 8).

Moreover, the other wider domains like navigation while avoiding obstacles, voice assistant, OCR can be further improved by working on their speed and accuracy. In

Fig. 7 Wearable top view

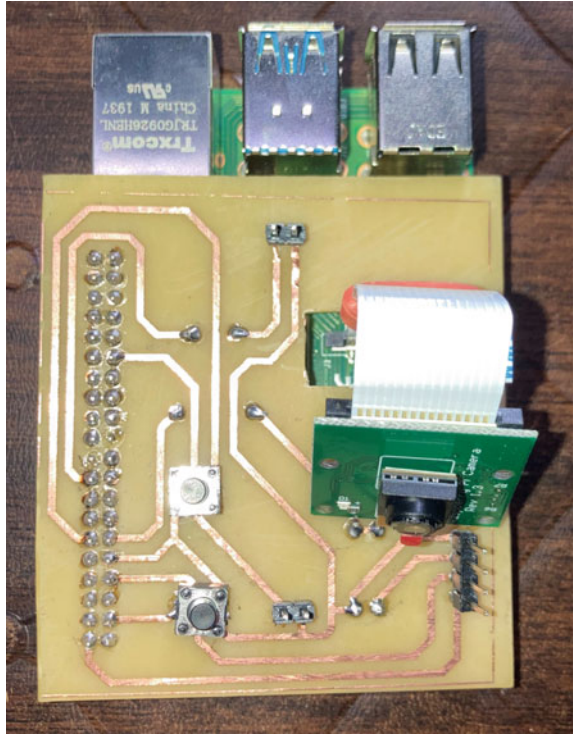
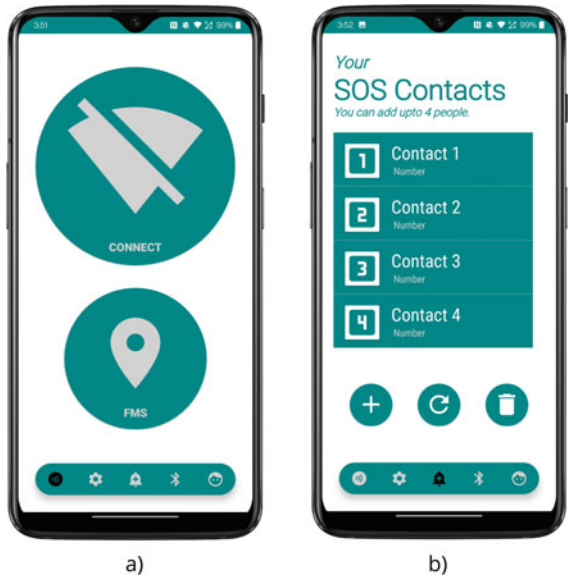


Fig. 8 a Home screen of the proposed mobile device application. b The SOS contacts screen in the application



the voice assistant module, additional functionalities like setting an alarm, etc. can be added.

References

1. Felix SM, Kumar S, Veeramuthu A (2018) A smart personal AI assistant for visually impaired people. In: 2018 2nd international conference on trends in electronics and informatics (ICOEI), pp 1245–1250. <https://doi.org/10.1109/ICOEI.2018.8553750>
2. Bose P, Malphthak A, Bansal U, Harsola A (2017) Digital assistant for the blind. In: 2017 2nd international conference for convergence in technology (I2CT), pp 1250–1253. <https://doi.org/10.1109/I2CT.2017.8226327>
3. Tahoun N, Awad A, Bonny T (2019) Smart assistant for blind and visually impaired people. In: Proceedings of the 2019 3rd international conference on advances in artificial intelligence (ICAAI 2019). Association for Computing Machinery, New York, NY, USA, pp 227–231. <https://doi.org/10.1145/3369114.3369139>
4. Bai J, Liu D, Su G, Fu Z (2017) A cloud and vision-based navigation system used for blind people. In: Proceedings of the 2017 international conference on artificial intelligence, automation and control technologies (AICT '17). Association for Computing Machinery, New York, NY, USA, Article 22, pp 1–6. <https://doi.org/10.1145/3080845.3080867>
5. Khan A, Khan A, Waleed M (2018) Wearable navigation assistance system for the blind and visually impaired. In: 2018 international conference on innovation and intelligence for informatics, computing, and technologies (3ICT), pp 1–6. <https://doi.org/10.1109/3ICT.2018.8855778>
6. Balakrishnan G, Sainarayanan G, Nagarajan R, Yaacob S (2008) A stereo image processing system for visually challenged impaired. World Academy of Science
7. Sharma V, Singh VM, Thanneeru S (2020) Virtual assistant for visually impaired. SSRN <https://ssrn.com/abstract=3580035>. <https://doi.org/10.2139/ssrn.3580035>
8. Saffoury R et al (2016) Blind path obstacle detector using smartphone camera and line laser emitter. In: 2016 1st international conference on technology and innovation in sports, health and wellbeing (TISHW), pp 1–7. <https://doi.org/10.1109/TISHW.2016.7847770>
9. Le V-H, Vu H, Nguyen TT (2018) A frame-work assisting the visually impaired people: common object detection and pose estimation in surrounding environment. In: 2018 5th NAFOSTED conference on information and computer science (NICS), pp 216–221. <https://doi.org/10.1109/NICS.2018.8606899>
10. Kim JU, Man Ro Y (2019) Attentive layer separation for object classification and object localization in object detection. In: 2019 IEEE international conference on image processing (ICIP), pp 3995–3999. <https://doi.org/10.1109/ICIP.2019.8803439>
11. Koskowich BJ, Rahmehoonfai M, Starek M (2018) Virtualot—a framework enabling real-time coordinate transformation & occlusion sensitive tracking using UAS products, deep learning object detection & traditional object tracking techniques. In: IGARSS 2018—2018 IEEE international geoscience and remote sensing symposium, pp 6416–6419. <https://doi.org/10.1109/IGARSS.2018.8518124>
12. Iyer V, Shah K, Sheth S, Devadkar K (2020) Virtual assistant for the visually impaired, pp 1057–1062. <https://doi.org/10.1109/ICCES48766.2020.9137874>
13. Young M (1989) The technical writer's handbook. University Science, Mill Valley, CA
14. Pise A, Ruikar SD (2014) Text detection and recognition in natural scene images. In: 2014 international conference on communication and signal processing, pp 1068–1072. <https://doi.org/10.1109/ICCSP.2014.6950011>
15. Bhowmick A, Prakash S, Bhagat R, Prasad V, Hazarika S (2014) IntelliNavi: navigation for blind based on kinect and machine learning. Multi-disciplinary trends in artificial intelligence (MIWAI '14), vol 8875, pp 172–183. https://doi.org/10.1007/978-3-319-13365-2_16

16. Awad M, Haddad JE, Khneisser E, Mahmoud T, Yaacoub E, Malli M (2018) Intelligent eye: a mobile application for assisting blind people. In: 2018 IEEE Middle East and North Africa communications conference (MENACOMM), pp 1–6. <https://doi.org/10.1109/MENACOMM.2018.8371005>
17. Mambu JY, Anderson E, Wahyudi A, Keyeh G, Dajoh B (2019) Blind reader: an object identification mobile-based application for the blind using augmented reality detection. In: 2019 1st international conference on cybernetics and intelligent system (ICORIS), pp 138–141. <https://doi.org/10.1109/ICORIS.2019.8874906>
18. Mohapatra S, Rout S, Tripathi V, Saxena T, Karuna Y (2018) Smart walking stick for blind integrated with SOS navigation system. In: 2018 2nd international conference on trends in electronics and informatics (ICOEI), pp 441–447. <https://doi.org/10.1109/ICOEI.2018.8553935>
19. Kornprobst P, Tumblin J, Durand F (2009) Bilateral filtering: theory and applications. *Found Trends Comput Graph Vis* 4:1–74. <https://doi.org/10.1561/06000000020>
20. Patel C, Patel A, Patel D (2012) Optical character recognition by open source OCR tool tesseract: a case study. *Int J Comput Appl* 55:50–56. <https://doi.org/10.5120/8794-2784>
21. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9:1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
22. Salehinejad H, Sankar S, Barfett J, Colak E, Valae S (2017) Recent advances in recurrent neural networks
23. Saoji S, Eqbal A, Vidyapeeth B (2021) Text recognition and detection from images using pytesseract. *J Interdiscip Cycle Res XIII*:1674–1679
24. Bressoud T, White D (2020) Web scraping. https://doi.org/10.1007/978-3-030-54371-6_22
25. Wang K, Gao X, Zhao Y, Li X, Dou D, Xu C (2020) Pay attention to features, transfer learn faster CNNs. *ICLR*
26. Howard A, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) MobileNets: efficient convolutional neural networks for mobile vision applications
27. Chiu Y-C et al (2020) Mobilenet-SSDv2: an improved object detection model for embedded systems. In: 2020 international conference on system science and engineering (ICSSE). *IEEE*
28. Shuai Q, Wu X (2020) Object detection system based on SSD algorithm. In: 2020 international conference on culture-oriented science & technology (ICCST), pp 141–144. <https://doi.org/10.1109/ICCST50977.2020.00033>
29. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards realtime object detection with region proposal networks. In: *Neural information processing systems (NIPS)*, pp 1–14
30. Gaikwad D, Baje C, Kapale V, Ladage T (2017) Blind assist system. *IJARCCCE* 6:442–444. <https://doi.org/10.17148/IJARCCCE.2017.63101>

Analysis of Python Libraries for Artificial Intelligence



Anand Khandare, Nipun Agarwal, Amruta Bodhankar, Ankur Kulkarni, and Ishaan Mane

Abstract Python libraries are a collection of essential functions that eliminate the need for users to develop code from scratch. Python is a plethora of libraries that serve a range of purposes and it has become a necessity to have sound knowledge of the best ones. Human and machine data production greatly outpaces humans' ability to absorb, assess, and make complicated decisions based on that data. AI (Artificial Intelligence) is the foundation of all computer learning and the future of all intricate decision making. These technologies are being looked upon as tools and techniques to make this world a better place. It's application ranges from various fields like healthcare, finance, transport, manufacturing, fraud detection and so on which evidently depicts its potential to transform the future. This paper intends to well verse the readers with the top libraries used to implement concepts of Artificial Intelligence like Machine Learning, Data Science, Deep Learning, Data Visualization and so on. It provides meticulous and unambiguous details about the essential building blocks necessary to execute and perform such ideas. It also includes a comparative analysis of various libraries to provide a detailed understanding and overview of them.

Keywords Artificial intelligence · Machine learning · Data science · Python · Python libraries · Deep learning · Healthcare

A. Khandare · N. Agarwal (✉) · A. Bodhankar · A. Kulkarni · I. Mane
Computer Department, Thakur College of Engineering and Technology, Mumbai, India
e-mail: nipunagarwal2001@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
V. E. Balas et al. (eds.), *Intelligent Computing and Networking*, Lecture Notes in Networks and Systems 632, https://doi.org/10.1007/978-981-99-0071-8_13

157

1 Introduction

Artificial Intelligence (AI) is a group of mighty interrelated technologies that can be used to perform complex and multiplex tasks openly with little or no human guidance. It is a vast topic of applied sciences and is used to demonstrate intelligence even beyond natural human intelligence. AI has progressed from data models for problem-solving to artificial neural networks, a computational model based on the structure and functions of human biological neural networks. Existing methodologies must be merged in order to convert vast amounts of data into value for individuals, businesses, and society. Furthermore, new barriers have evolved, not just in terms of magnitude (“Big Data”), but also in terms of the questions that should be answered. It encompasses broad topics like Experts Systems, Robotics, Machine Learning, Data Science, Neural networks, Fuzzy Logic, Natural Language Processing and so on which are booming rapidly in the times we live in. AI and DS are influenced by python. It helps to slow down the efforts of the human brain. From speech recognition to data interpretation, python and its libraries have taken AI to the sky with remarkable success. Neural networks and machine learning are the branches of AI that illuminate the future of python. Python language focuses on the readability of the code. Python has incorporated mathematical libraries and functions, making it easy to calculate mathematical problems and perform data analysis. Python libraries are in great demand even in the IoT sector. It is widely used in home and office automation to speed up the work process easily. There are numerous tools, programming languages and applications to build AI-based systems out of which Python is the most popular language for AI because of its flexibility, platform autonomy, better data visualisation and optimization options. The Python libraries are a powerful way to invent AI-based systems in a very efficient and pragmatic manner. This article aims to concentrate on and analyse the characteristics of the most popular Python libraries, as well as their potential for data mining and big data research. Throughout the paper, each of these techniques will be examined in-depth, with examples of their use in diverse domains. These libraries are the most frequently used and respected resources for solving real-world issues and developing high-tech systems.

2 Tools in Python Aiding to Artificial Intelligence

Python is a powerful interpreted language with a solid core foundation and a robust modular component that extends the language with external modules that provide new features. As a result, we now have an extensible language with tools for doing a particular operation as efficiently as feasible. Packages are frequently used to arrange modules. A package is a logical grouping of modules that all serve the same function.

2.1 *Numpy*

The Python programming language has a large number of high-level data structures available such as lists that enumerate collections of objects, dictionaries for building hash tables, and more. However, these structures are not ideally suited for high-performance numerical calculations. All the fundamental operations in programming include mathematical tools such as arrays, matrices, integration tools, linear equation solvers, differential equation solvers, etc. Python provides the necessary tools to easily perform these complicated calculations and difficult mathematical operations in the form of its Numpy package.

In the mid-nineties, an international team of volunteers began developing data structures for efficient array calculations [1]. This structure has evolved into what is now known as the N-dimension NumPy array. NumPy is a general-purpose array-processing package. Its main object is the homogeneous multidimensional array. It is a table containing elements (commonly numbers) of the same type that are all indexed by a tuple of non-negative integers. In NumPy dimensions are called axes. The package provides a fast interface for storing and manipulating dense data buffers. NumPy arrays are similar to Python's built-in list type in certain aspects, but NumPy arrays allow far more efficient storage and data operations as the arrays grow in size. NumPy arrays are at the root of basically the entire Python ecosystem of data science tools. Python NumPy arrays provide tools for integrating C, C++, etc. It is also useful in linear algebra, random number capability etc. The NumPy array can also be used as a multi-dimensional container for general data [1]. We can initialize NumPy arrays from nested Python lists and access its elements. The NumPy package, consisting of NumPy arrays and their corresponding set of mathematical functions, is widely used in academia, national labs, and industry in applications from games to space exploration.

Here are the top four benefits that NumPy can bring to your code:

1. More speed: NumPy uses algorithms written in C that complete in nanoseconds rather than seconds.
2. Fewer loops: NumPy assists you in reducing loops and avoiding becoming entangled in iteration indices.
3. Clearer code: Without loops, the code will look more like the equations which are being tried to calculate.
4. Better quality: There are thousands of contributors working to keep NumPy fast, friendly, and bug free.

Because of these advantages, NumPy has become the de facto standard in Python data science for multidimensional arrays, and many of the most popular libraries are built on top of it. It is a very important library on which almost every data science or machine learning Python packages such as SciPy (Scientific Python), Matplotlib (plotting library), Scikit-learn, etc. depend on to a reasonable extent. Learning NumPy is an excellent approach to lay a solid basis for furthering one's expertise in more specialised fields of data science.

```

In [70]: C = np.array([9,4,3,6,4,5,2,1])
print(np.sort(C))
[1 2 3 4 4 5 6 9]

In [71]: D = np.array([11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22])
Reshape = D.reshape(2, 3, 2)
print(Reshape)
[[[11 12]
  [13 14]
  [15 16]]
 [[17 18]
  [19 20]
  [21 22]]]

In [72]: Trig = np.array([np.pi/2, np.pi/4, np.pi/6, np.pi/8])
x = np.sin(Trig)
print(x)
[1.         0.70710678 0.5         0.38268343]

In [73]: LOG = np.arange(1, 20)
print(np.log(LOG))
[0.         0.69314718 1.09861229 1.38629436 1.60943791 1.79175947
 1.94591015 2.07944154 2.19722458 2.30258509 2.39789527 2.48490665
 2.56494936 2.63905733 2.7080502  2.77258872 2.83321334 2.89037176
 2.94443898]

```

Fig. 1 Basic matrix operations using NumPy

The Numpy library uses multidimensional ndarrays to perform numerical processing by allowing broadcasting i.e. each and every element is processed one after another. It also allows wrapping codes of C, C++ and FORTRAN to compute core mathematical operations including linear algebra. NumPy operates on in-memory arrays using the central processing unit (CPU). To utilize modern, specialized storage and hardware, there has been a recent proliferation of Python array packages [2]. Numpy offers N-Dimensional arrays (ndarray), a storage object and a Universal Function Object (ufunc) for computing data efficiently. The ndarray consists of two essential pieces of information first being the size of the array (also referred as the shape of the array, it is a tuple of N integers containing the number of rows and columns) and second being the datatype of the items stored inside the Numpy array (Fig. 1).

2.2 Pandas

The pandas package is the most important data manipulation and analysis tool at the disposal of Data Scientists and Analysts working in Python today. Although advanced machine learning and fancy visualisation tools get all the attention, pandas is the foundation of most projects and initiatives. Through pandas, one can get acquainted with their data by cleaning, transforming, and analyzing it. The library's name springs from panel data, a general term for multidimensional data sets encountered in statistics and econometrics. It is a Python library that includes a variety of knowledge structures and tools for working with structured datasets that are used in statistics, finance, social sciences, and a variety of other fields. The library is incorporated to

conduct typical data processing and analysis of certain datasets, with user-friendly procedures.

The `panda` library, which has been developed since 2008, aims to bridge the gap in the abundance of data analysis tools available between Python, general-purpose systems, and scientific computing languages, and many domain-specific statistical computing platforms and database languages. The aim is to provide equivalent functionality and implement many features, such as automatic data alignment and hierarchical indexing, which are not available in a tightly integrated manner like so in other libraries.

`Pandas` is constructed on top of the `NumPy` package, which means that much of `NumPy`'s structure is used or recreated in `Pandas`. `Pandas` data is frequently used to feed statistical analysis in `SciPy`, graphing functions in `Matplotlib`, and machine learning algorithms in `Scikit-learn`.

The `Series` and `DataFrame` are the two main components of `pandas`. A `Series` is just a column, while a `DataFrame` is a multi-dimensional table composed of `Series`. Exploring, cleaning, transforming, and visualizing data with `pandas` in Python is an essential skill in data science. Just cleaning wrangling data is 80% of the job of a Data Scientist.

Operations after loading dataset in the Python Environment:

- You may compute the fundamental statistics of your dataset and answer typical inquiries such as what the mean, median, minimum, and maximum values are.
- A correlation between two or more columns in the dataset can also be discovered.
- Clean up the data by deleting missing or blank values and filtering entries based on a criterion.
- Use other modules to visualise the data, such as `seaborn`, `matplotlib`, and so on.
- Save the cleaned data frame to a CSV or database of your choosing.

We believe that in the coming years there will be great opportunities to attract users in need of statistical data analysis tools to Python who might have previously chosen R, MATLAB, or another research environment [3]. By designing robust, easy-to-use data structures that cohere with the rest of the scientific Python stack, we can make Python a compelling choice for data analysis applications. In our opinion, `pandas` provides a solid foundation upon which a very powerful data analysis ecosystem can be established (Fig. 2).

2.3 *Matplotlib*

Representation of data in visual form is a necessity nowadays. As the amount of data is increasing day by day, it isn't easy to manage and represent data in text. Human brains are more flexible and adjustable to visual representation, and this helps to comprehend, analyze, and make decisions for AI and ML. `Matplotlib` is a Python plotting tool that produces high-quality graphics. `Matplotlib` was created with the goal of allowing users to produce basic as well as complicated plots with only a

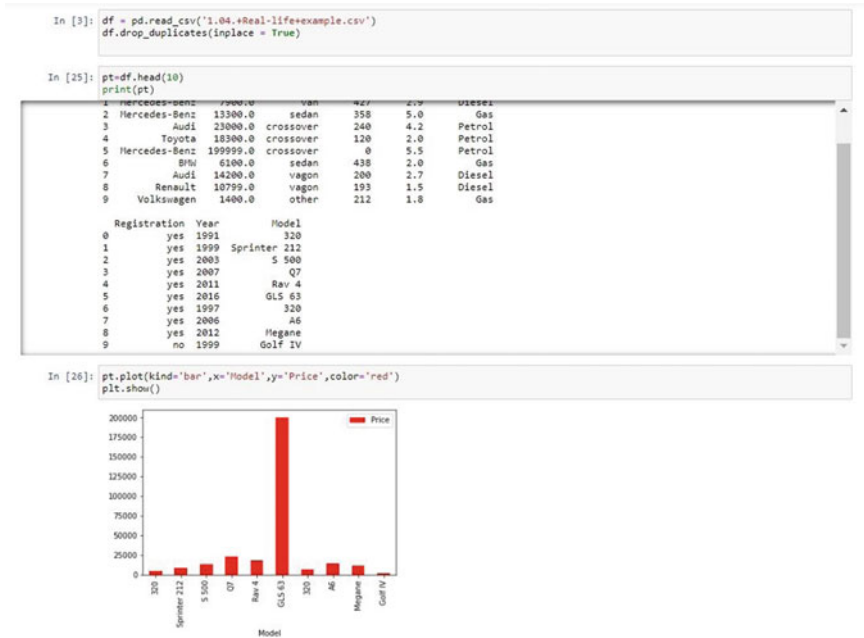


Fig. 2 Data analysis using Pandas and plotting its graph

few instructions [4]. Jupyter notebooks and web application servers can also utilise Matplotlib.

Graphs of production quality may be generated with Matplotlib’s Python module for 2D plotting. In addition, it can export pictures in a variety of output formats and offers both interactive and non-interactive plotting capabilities. In addition, it supports a broad range of plot kinds and can work with different window toolkits. Aside from that, it’s very customisable, versatile, and simple to use. Due to Matplotlib’s dual nature, it may be used both in interactive and non-interactive scripts. Use it in scripts without a visual display, in graphic apps, or in web pages. Python or IPython can also be used interactively with matplotlib.

John D. Hunter initially wrote Matplotlib in 2003. It is open-source software and can be downloaded, used, and distributed freely.

Key features

Many different situations may be addressed using Matplotlib. Plots and pictures may be created interactively using the command-line, which is well-known to most users. A simple pop-up window is used to show and manipulate the data. The main plotting module of matplotlib, which is operating system and GUI agnostic, is the true power of the library [5]. Can be used as part of a webserver to produce plots and pictures in different hardcopy output formats, or it can be integrated in a more complete programme using one of many GUIs, running on one of several OSs.

Pyplot—*matplotlib.pyplot* is a group of functions and commands that make matplotlib work like MATLAB. These commands are helpful to create figures, make changes in them, create plotting areas, and embellish the plots.

Pylab: Pylab is an interface to matplotlib for object-oriented plotting. The Pylab module is installed alongside matplotlib.

The types of plots include-Bar, Barht, scatter, stack, Box Plot, step, quiver, violin, Hist, hist2d, pie, plot, polar plot, stemplot.

Axis functions like—Title (Add text to the axes), xlabel (Set the x-axis label of the current axis), xlim, xscale (Set the scaling of the x-axis), Axes (Add axes to the figure), Text (Add text to the axis), ylabel (Set the y axis label current axis), ylim (Get or set the y-limits of the current axes), scale (Set the scaling of the y-axis), are useful to embellish the plots.

Figure functions like- Figtext, Figure, Show, Savefig, Close are important functions for the creation of plots.

Matplotlib's ease of usage is largely due to the following features:

- Open-source, thus there's no need to pay for a licence: Students and teachers on a tight budget will find it intriguing.
- It's an actual programming language: The MATLAB language lacks many of the features of a general-purpose language like Python.
- It's much more complete: Python has many external modules that will help us execute all the functions we need to perform.
- With a simple interactive GUI, the plot window allows you to zoom in and out of the plot as well as remember the plot's history and save it to hardcopy.
- The command-line interface is based on the MatLab interface, which is straightforward to use.
- Plot and picture support on many pages
- For the GD, Agg and Paint backends as well as PostScript, there are TrueType fonts that may be used.
- Mathematical text LATEX math mode is available whenever TrueType fonts are available.
- Resampled images are automatically resized to the figure's dimensions.
- Assists in programming and development using a fully object-oriented design.

Matplotlib is mostly used to create graphical applications. Python has a convenient graphing package. The Matplotlib library is ideally suited to developing an interactive two-dimensional application, while the three-dimensional plots library is used to build a three-dimensional application. It may also be used to create high-quality photographs [6]. The features and the facilities of matplotlib are advancing day by day. Some of the new features include creating 3D plots using the mplot3d toolkit. Contour plots, wireframe plots, surface plots can also be made using matplotlib. Transformations in the coordinate axes can be quickly done and manipulated. Matplotlib can plot anything, however plotting non-basic plots or adjusting graphs to appear beautiful can be difficult (Fig. 3).

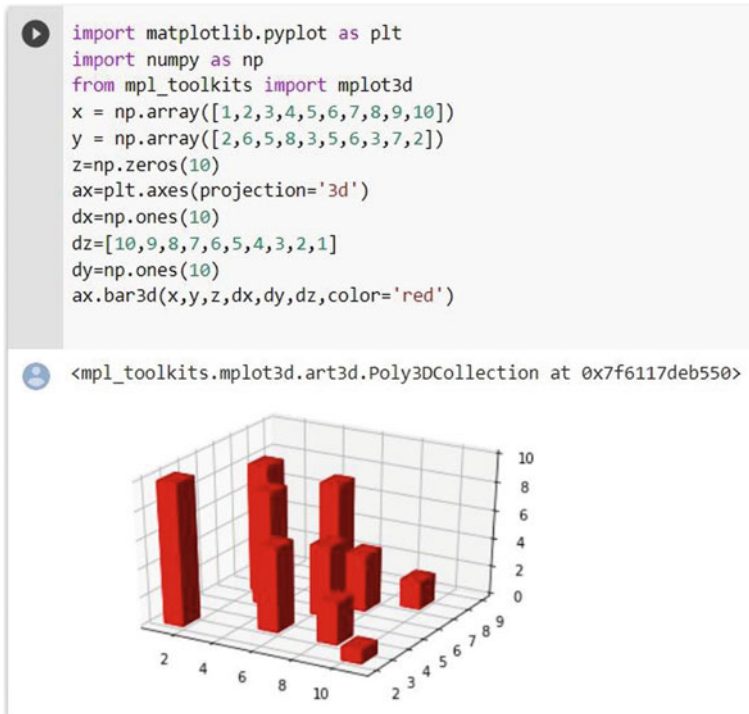


Fig. 3 Plotting a 3-dimensional graph for the given data using Matplotlib

2.4 Seaborn

In this arena of data analysis and interpretation, visualisation of data is the best way to get insights and gain information from the data. For this purpose Seaborn is an excellent library for making statistical graphs in Python. It provides an excessive-degree interface to matplotlib and is tightly integrated with the Pandas data structure [7].

The seaborn library functions expose a dataset-oriented declarative API that can easily convert questions about the data into charts. When a dataset and drawing specification is provided to produce, seaborn will automatically map the data values to visual attributes such as colour, size, or style, and calculate statistical transformations internally. Seaborn is designed to play a role throughout the life cycle of scientific projects. By generating complete graphs from a single function call with minimal parameters, seaborn can easily and quickly build prototypes and analyze exploratory data. By providing a wide range of customization options, in addition to exposing the underlying matplotlib objects, it can also be used to create polished shapes and visuals.

Users interact with seaborn through a number of drawing functions. These drawing functions share common APIs for drawing specifications and provide many more specific customization options. These functions range from basic drawing types, such as scatter plots and line graphs, to functions that apply various transformations and abstractions, such as histogram fusion, kernel density estimation, and regression model fitting. Functions in seaborn are categorized as “*axis level*” or “*figure level*”. The behaviour of the Axes level function is similar to most plot functions in matplotlib-pyplot namespace. By default, they are linked to a state machine that tracks the “current” figure and appends a layer to it, but they can also accept objects from the matplotlib axis to control the position of the graph, similar to using matplotlib’s “*object-oriented*” interface. The Figurelevel function creates its own graphs when called, allowing them to “facet” the dataset by creating multiple conditional subplots and adding conveniences, such as placing the legend outside of the graph space by default. Each figure level function corresponds to several axis level functions for similar purposes, using a parameter to select the type of drawing to be performed. For example, the displot function can generate several different distribution representations, including histograms, kernel density estimates, or empirical CDFs. The figure level function utilizes a seaborn class to control the layout of the figure and mediate between the axis layer function and matplotlib. These classes are part of the public API and can be used directly in advanced applications.

Seabron is one of the most widely used python libraries. With Seaborn we can visualise both univariate and bivariate data. It provides rapid, detailed and accurate graphics. It has built-in functions and themes for embellishing the plots, plus has an edge over matplotlib. Seaborn is a complement to matplotlib. This library is easy to comprehend and implement (Fig. 4).

2.5 Scikit

Scikit-learn is one of the most beneficial and a key python library which is a structured tool for machine learning and statistical modelling. It is an open-source and commercially available software. It is capable of performing numerous statistical, data mining and data analysis operations like- Classification, Clustering, Regression. Scikit-learn is straightforward in design, efficient and is easily approachable through non-experts. It first emerged through David Cournapeau as a Google summer time season code project in 2007. Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort and Vincent Michel, from FIRCA, took this project to the next possible degree and made the primary launch in 2010.

The features of the package Scikit-learn are:

Supervised Learning Algorithms—Nearly all supervised learning algorithms like, linear regression, Decision Tree, SVM (Support Vector Machine) belong to Scikit-learn. These algorithms help to estimate the outcomes for unforeseen data.

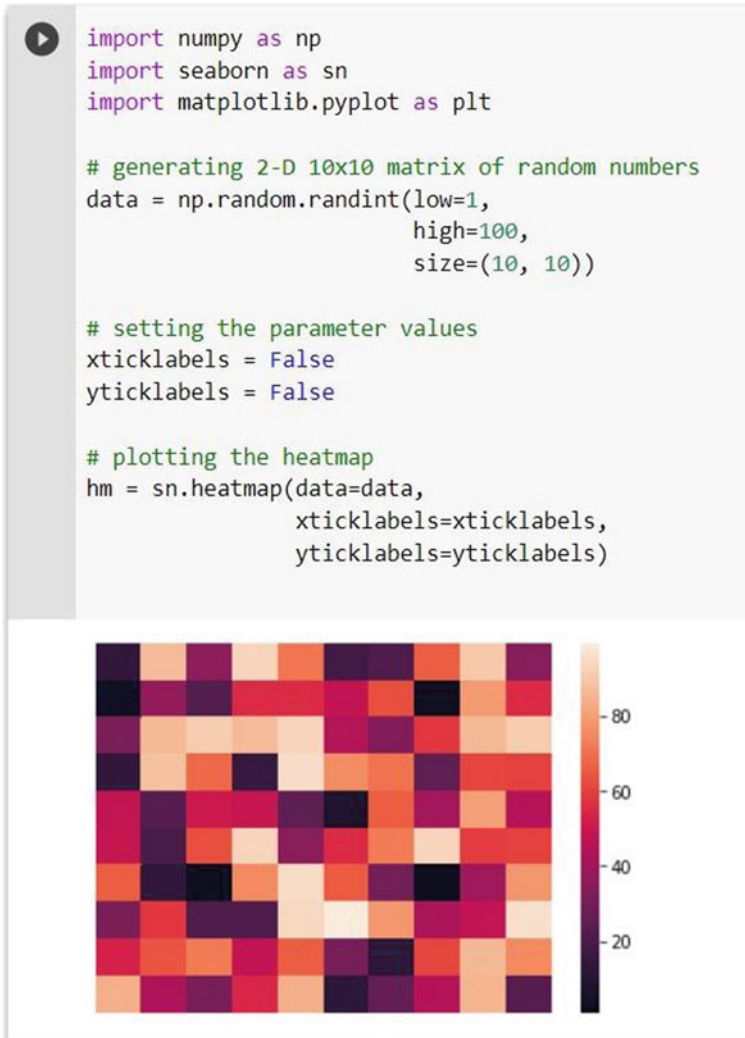


Fig. 4 Plotting a heatmap using Seaborn

Unsupervised Learning Algorithms—This library also includes very popular unsupervised learning algorithms of clustering, PCA (principal component analysis), factor analysis which helps in performing more complex processing tasks [8]. It allows the model to work on its own, without any supervision and discover information and data.

Cross Validation—It is used to verify the accuracy of supervised models on unseen data and helps in estimating the performance of models.

Feature Extraction—It is used to extract features from the data consisting text and images in formats supported by machine-learning. It includes functions like-DictVectorizer(), feature_name, CountVectorizer() and many more.

Feature Selection—This module is used to recognize useful attributes to create supervised models. Feature selection methods are used for simplification of models, improve data compatibility, and make the data easier for the users to interpret.

Dimensionality Reduction—This is used to reduce the attributes in the data for feature selection, summarisation and visualization. The transformed low-dimensional space retains important features of the original data and is convenient to analyze and process using machine learning techniques. This can be done using the PCA functions like-PCA (n-components, svd_solver), pca.fit().

Scikit-learn uses an enormous and substantial variety of machine learning algorithms. It supports machine learning in python and allows to build various machine learning models for predicting and deciphering abstract, unorganised and unexpected data. Scikit-learn is a set of successfully implemented machine learning algorithms that is well-documented and maintained by the community [9]. It is a useful tool to process large to small scale data. Both supervised and unsupervised learning methods can be adopted by using well suited and task-based interfaces. This enables assessment of methods and strategies for a given application (Fig. 5).

2.6 TensorFlow

Tensors are groups of data with an arbitrary number (zero to infinity) of dimensions. They can be arranged in scalars (dimensionless), vectors (unidimensional), matrices (2 dimensional), cubes (3 dimensional), sets of cubes (4 dimensional) and so on. TensorFlow is a platform for solving machine learning algorithms, and an implementation for such algorithms [10].

The flow of information between various tensors is controlled by thousands of parameters. In a neural network, elements in one tensor are bound to elements in the next. Tensor Algebra deals with this information flow between the various tensors.

The TensorFlow open-source software library is a collection of tools developed by Google for numerical computation. It is commonly used by IT firms and giants to perform various computational tasks. Its library can be executed on various platforms, such as mobile platforms and distributed systems with very little or no modification [11].

A few salient features of the TensorFlow framework are as follows:

1. Python is the language of choice for Theano and TensorFlow. Mx Net also consists of some useful Python APIs. TensorFlow and Theano are very similar when it comes to Deep Learning systems, however TensorFlow is preferable for distributed systems as it has better support for them [12].
2. TensorFlow uses Automatic Differentiators which are different from numeric and symbolic differentiation. Automatic differentiators are very useful and efficient

```

import numpy as np
from sklearn.linear_model import LinearRegression
X = np.array([[1,1],[1,2],[2,2],[2,3]])
y = np.dot(X, np.array([1,2])) + 3
#Creating linear regression object
regr = LinearRegression(
fit_intercept = True, normalize = True, copy_X = True, n_jobs = 2).fit(X,y)
#Using predict() method to predict using this linear model
regr.predict(np.array([[3,5]]))
#To get the coefficient of determination of the prediction we can use Score()
#regr.score(X,y)
#To estimate the coefficients by using attribute named 'coef'
#regr.coef_

array([16.])

[2] import numpy as np
from sklearn.linear_model import LinearRegression
X = np.array([[1,1],[1,2],[2,2],[2,3]])
y = np.dot(X, np.array([1,2])) + 3
#Creating linear regression object
regr = LinearRegression(
fit_intercept = True, normalize = True, copy_X = True, n_jobs = 2).fit(X,y)
#Using predict() method to predict using this linear model
#regr.predict(np.array([[3,5]]))
#To get the coefficient of determination of the prediction we can use Score()
#regr.score(X,y)
#To estimate the coefficients by using attribute named 'coef'
regr.coef_

1.0

```

Fig. 5 Basic program for linear regression using SciKit learn

in neural networks and can be easily understood using the simple chain rule of differentiation.

3. TensorFlow is compatible with various platforms like Android, IOS, Cloud as well as architectures such as CPU and GPU. TensorFlow applications can easily be executed on these platforms and architectures. This is primarily due to the ability of TensorFlow to train neural models using its own designed hardware known as TPU's (TensorFlow Processing Units)
4. TensorFlow is very proficient in data abstraction. A defined view level is available for the user so that the programmer does not have to focus on the procedure to provide or receive inputs, rather more emphasis is given on understanding and implementing the logic behind the problem statement.
5. TensorFlow is more effective to implement deep-learning models as the data structure tensor allows this framework to work with multidimensional arrays. A tensor can be categorized using the attributes rank, type and shape. [13] All tensors are immutable i.e., once some data is stored in a tensor it cannot be changed. To store new data, we need to define a new tensor all together (Fig. 6).


```
import tensorflow.compat.v1 as tf
tf.disable_v2_behavior()

node = tf.Variable(tf.zeros([3,3]))

with tf.Session() as sess:

    sess.run(tf.global_variables_initializer())

    print("Tensor value before addition:\n",sess.run(node))

    node = node.assign(node + tf.ones([3,3]))

    print("Tensor value after addition:\n", sess.run(node))
```

Tensor value before addition:
[[0. 0. 0.]
[0. 0. 0.]
[0. 0. 0.]]
Tensor value after addition:
[[1. 1. 1.]
[1. 1. 1.]
[1. 1. 1.]]

Fig. 6 Demonstrating matrix addition using TensorFlow

2.7 Keras

Keras is a Python-based deep learning framework that makes it simple to design and train nearly any deep learning model. It is a high-level neural network API that can be used with Tensorflow, Theano, and CNTK. It was created to allow for quick experimenting.

Keras relieves developer cognitive strain, allowing you to focus on the most important aspects of the problem.

The Keras principle of progressive disclosure of complexity states that simple processes should be quick and straightforward, whereas arbitrarily sophisticated workflows should be feasible via a clear route that builds on what you’ve already learned.

Keras is utilised by organisations and enterprises like NASA and YouTube to deliver industry-leading performance and scalability.

Keras has the following features

1. Convolutional and recurrent networks, as well as a mixture of the two, are supported.

2. It can handle a wide range of network architectures, including multi-input and multi-output models, layer and model sharing, and so on. As a result, Keras may be used to create deep learning models ranging from generative adversarial networks to a neural Turing machine.
3. Keras is a modular design. A model in the form of a graph or a sequence is considered. Keras gives you the option of saving the model you're working on.
4. Keras comes with a huge dataset that has been pre-defined. It gives you access to a range of datasets. You may use this dataset to import and load it directly.
5. Keras includes a number of models that have already been trained. Keras may be used to import these models. Applications. These models are useful for extracting features and fine-tuning them.
6. Keras is a Python library in its entirety. It employs all of Python's well-known ideas. It is a library built in the Python programming language. Keras delivers a user-friendly environment because it is Python-based.
7. Keras includes a number of functions for data pre-processing.

Keras is a human-centric API, not a machine-centric API. Keras adheres to best practises for decreasing cognitive load by providing consistent and straightforward APIs, minimising the amount of user activities required for typical use cases, and providing clear and responsive feedback in the event of user error.

Keras is thus simple to understand and use. As a Keras user, you are more productive, allowing you to attempt more ideas than your competition, faster, which helps you win machine learning competitions.

This simplicity does not come at the expense of flexibility: because Keras integrates strongly with low-level TensorFlow capabilities, it allows you to create highly hackable workflows in which any piece of functionality is customizable. Keras makes it simple to convert models into final products.

Keras uses a strong and clear deep learning library built on top of TensorFlow/Theano to give high-level neural networks. Keras is an excellent addition to TensorFlow because its layers and models work with pureTensorFlow tensors [14].

Keras is simple to understand and use as a result of this. Keras users are more productive, allowing them to test more ideas faster than their competitors, which helps them win machine learning contests.

Keras is a human-centric API. Keras adheres to best practices for minimizing cognitive load, such as providing consistent and straightforward APIs, decreasing the number of user steps necessary for typical use cases, and providing clear and responsive feedback in the event of a user error.

2.8 Theano

Python is a potent and pliant programming language for machine learning and in addition involves plenty of complex mathematical calculations, algorithms, arithmetic computations and mainly large matrices of multiple dimensions. To build such

complex machine learning algorithms and to advance the mathematical expressions Theano is a structured and ideal python library. This library can be run on CPU or GPU [15]. It is an open source software and released under a BSD license. For building different algorithms and codes Theano requires mainly the support of Numpy, SiPy, BLAS.

Features of Theano

Theano permits to define, optimize and evaluate mathematical expressions. It efficiently facilitates the development of Machine Learning models. Generating computational graphs is a key feature of Theano which helps in expressing and calculating a mathematical expression. There are also various data types in Theano like- Scalars, Tensor, Matrix, Vectors, Arrays, Plural Constructors, Complex, Double, Float, Byte, 16-bit integers, 32-bit integers and 64-bit integers. These data types are used in Theano with proper and structured syntax. It also involves variables and shared variables. Theano functions are bridges for interacting with symbolic graphs. Some parts of the code are compiled in C. The compiled code is then provided input to Theano function [16]. In this way an optimum code is executed. But creating complex set of codes and algorithms using Theano is possible due to-

- **Stability Optimization-** Theano is not just used for integrating mathematical expressions but can be used also to stabilize the unstable expressions in order to get optimum results.
- **Faster Execution Speed-** Theano utilizes current GPU's and can execute the expressions much faster. Plus it produces dynamic C code which helps in evaluation of expressions faster.
- **Symbolic Differentiation-** Theano is capable of automatically generating symbolic graphs for computing gradients. It performs derivatives of functions with one or many inputs.

Building Machine Learning models incorporates rigorous and complex computations. For this Theano is an excellent platform. It is a beneficial tool to enhance the execution time and perform repetitive computations of mathematical expressions. It can be used for deep learning and solving real-world problems. Theano is well developed and accepted world-wide by industries and academics (Fig. 7).

2.9 PyTorch

Traditionally, deep learning frameworks have prioritised either usability or speed, but rarely both. The machine learning toolkit called PyTorch demonstrates that these two aims may coexist. PyTorch provides an imperative and Pythonic programming language that allows code as a model, makes debugging easy while being efficient and enabling hardware accelerators such as GPUs. PyTorch, like Python, is a fantastic introduction to deep learning as well as a tool that can be used in sophisticated real-world applications [17].



```

from theano import *
a = tensor.dmatrix()
b = tensor.dmatrix()
c = tensor.dot(a,b)
f = theano.function([a,b], c)
d = f([[5, -1, 5],[-2, 10, 8]], [[3, -4],[3,6],[6,10]])
print (d)

```

[[42. 24.]
 [72. 148.]]

Fig. 7 Demonstrating matrix multiplication using Theano

Facebook, Inc. created PyTorch, an open source machine learning and deep learning library. It's Python-based, as the name implies, and attempts to be a faster NumPy alternative [17].

Uber's Pyro probabilistic programming engine is built on it. Using the same core C libraries for the backend code as Torch, PyTorch re-designs and implements Torch in Python using the same C libraries. To make Python run as efficiently as possible, PyTorch engineers optimised the backend code. Lua-based Torch retained the GPU-based hardware acceleration as well as the extensible capabilities that made it famous.

PyTorch key features

- *Front-end*: Using PyTorch, a user-friendly and flexible front-end is created, which seamlessly transitions from diagram format to C++ execution contexts for speed, optimization, and operability.
- *Dispersed training*: C++ and Python provide native asynchronous implementation of cooperative processes and peer-to-peer communications, which improves speed in both exploration and production.
- *Tools and libraries*: Scientists and innovators have collaborated to create an extensive network of tools and libraries to help disseminate PyTorch and advance research in fields such as Reinforcement Learning.
- *Cloud partners*: PyTorch is extremely well supported on the most popular cloud platforms, enabling not only frictionless development but also stress-free scaling with large scale preparation on GPUs, the ability to track models in a construction scale setting etc.
- *Programming*: Every time PyTorch reads a line of code, it does computations. In many ways, this is the same as running a Python application. Imperative programming is a term used to describe this type of programming. It also has the advantage of making it easier to debug and programme the logic.

- *Dynamic Graphing*: It is said that PyTorch is defined by run, which implies that execution time is when the real computation graph for neural network design is generated. However the major value of this characteristic is that it provides an elastic and programmable execution interface, which makes it possible to create or modify whole system structures through the use of process-linking techniques. PyTorch is a great framework for creating dynamic computing graphs, which may be modified during operation. If you don't have the memory to create a neural network model, this can be incredibly helpful. A new computational graph is created at each PyTorch advancing pass. TensorFlow's approach is significantly different. Inventors are often faced with reworking, training, adaptability, and scalability problems. All of these duties take a lot of time, and they demand a lot of work. This is why PyTorch was created to assist innovators and researchers in these fields with its sophisticated capabilities.

By combining an emphasis on usability with thorough performance considerations, PyTorch has become a popular tool in the deep learning research community. Pytorch offers a lot of customizability with minimum code, in addition to continuing to support the newest trends and developments in deep learning. While it may be difficult to grasp how the entire ecosystem is organised using classes at first, it is ultimately just Python (Fig. 8).

2.10 NLTK

Processing and understanding human language data are crucial for any interactive AI to function properly, provide more value and solve problems. NLP (Natural Language Processing) is a domain that focuses on understanding, processing and implementing human language data effectively to solve real world problems by ensuring that the computer-human interaction takes place smoothly. NLTK is implemented as a large collection of minimally interdependent modules, organized into a shallow hierarchy [18]. The Natural Language Toolkit (NLTK) was created in 2001 at the University of Pennsylvania in connection with a computational linguistics course. Assignments, demonstrations, and projects were the three pedagogical uses in mind when it was created.

NLTK (Natural Language Toolkit) is a python package which is predominantly used for NLP. NLTK preprocesses unstructured data containing human language references using computational linguistics, NLP data types and animated algorithms. NLTK also provides problem sets and tutorials to make the user familiar with this python library. Natural language processing functions are drawn up as transformations on Tokens [19]. NLTK is very beneficial for the students or programmers who are learning NLP or conducting research on the same topic.

Run the following instructions in your terminal to install NLTK

```
sudo pip install nltk.
```

```

✓ [1] import torch
29s

✓ [2] a = torch.Tensor([[1,2],[3,4]])
0s
print(a)

tensor([[1., 2.],
        [3., 4.]])

✓ [3] print(a**2)
0s

tensor([[ 1.,  4.],
        [ 9., 16.]])

✓ [4] y = torch.sum(a**2)
0s
print(y)

tensor(30.)

✓ [5] shape = (2,3,)
0s
rand = torch.rand(shape)
ones = torch.ones(shape)
zeros = torch.zeros(shape)

print(f"Random Tensor: \n {rand} \n")
print(f"Ones Tensor: \n {ones} \n")
print(f"Zeros Tensor: \n {zeros}")

Random Tensor:
tensor([[0.7965, 0.1374, 0.3329],
        [0.6186, 0.0169, 0.8499]])

Ones Tensor:
tensor([[1., 1., 1.],
        [1., 1., 1.]])

Zeros Tensor:
tensor([[0., 0., 0.],
        [0., 0., 0.]])

```

Fig. 8 Basic operations on tensors using Pytorch

Then, on your terminal, type python to launch the Python shell and run the following instructions.

```
import nltk
nltk.download('all')
```

Since NLTK is completely written in python, it has the following features:

- Easier and convenient to learn.
- Exceptional at string-handling.
- Well defined syntax.
- Data encapsulation is possible and data can be reused multiple times.

NLTK Implementation:

- Chatbots
- Machine Translation
- Speech Recognition
- Text Summarization
- Recommendation Engine
- Sentiment Analysis for Customer Reviews

NLTK has been effectively utilised as a teaching tool, as a tool for individual study, and as a platform for prototyping and developing research systems. NLTK offers a simple, versatile, and consistent framework for assignments, projects, and class presentations. It’s well-documented, easy to understand and utilise. The most popular tool for teaching NLP is NLTK. It’s also commonly used as a prototype and research tool [20].

3 Comparative Analysis

See Tables 1, 2 and 3.

Table 1 Matplotlib versus seaborn

Features	Matplotlib	Seaborn
Utility	Developed for basic plotting and extends MATLAB	Extends Matplotlib and specializes in statistics visualization
Flexibility	Highly customizable	Includes default themes
Handling multiple figures	Multiple figures can be opened	Automates creation of multiple figures
Dependency	Uses Numpy majorly for plotting	Uses Pandas heavily for plotting

Table 2 Comparative analysis of libraries

Parameters	Keras	Tensorflow	PyTorch
Open source	Yes	Yes	Yes
Level of API	High-level API	Both high- and low-level APIs	Lower-level API
Speed	Slower	Fast	Equivalent to Tensorflow
Architecture	Simple architecture	Not easy to use	Complex
Debugging	Less frequent need to debug simple networks	Difficult to perform debugging	Better debugging capabilities
Datasets	Small datasets	Large datasets	Large datasets
Popularity	1st in popularity	2nd in popularity	3rd in popularity
Trained models	Yes	Yes	Yes
Programed in	Python	Python, C++, CUDA	Lua
Community	Smaller community support	Large community support	Stronger community support
Ease of deployment	Deployment can be done with TensorFlow or flask	Easy to deploy TensorFlow serving	Deployment is easy but not as much as Tensorflow

Table 3 Matplotlib versus Seaborn

Plot type	Matplotlib	Seaborn
Spectrogram	Yes	No
3D plot	Yes	No
Pair plot	No	Yes
Heat map	No	Yes
Polar plot	Yes	No
Regression plot	No	Yes

4 Conclusion

Python's libraries, modules, and frameworks have made it very simple to implement Artificial Intelligence ideas. Python machine learning libraries have evolved into the most widely used language for building machine learning algorithms. Understanding Python is essential for building conceptual knowledge of Artificial Intelligence and specialising in it. Python libraries are crucial in developing machine learning, data visualisation, data science, image and data processing, and other applications. This paper properly discussed, compared, and emphasised the critical and necessary Python Programming Libraries involved in researching the vast topic of Artificial Intelligence.

Acknowledgements This paper on ‘Analysis of Python Libraries for Artificial Intelligence’ has been possible only because of kind cooperation lent by our teacher and project guide Dr. Anand Khandare without which this would not have been possible. We would also like to thank our parents, who have provided us with all possible resources to gain the best possible knowledge.

References

1. Van Der Walt S, Chris Colbert S, Varoquaux G (2011) The NumPy array: a structure for efficient numerical computation. *Comput Sci Eng* 13(2):22–30
2. Harris CR, Millman KJ, Van Der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Oliphant TE et al (2020) Array programming with NumPy. *Nature* 585(7825):357–362
3. McKinney W (2011) Pandas: a foundational Python library for data analysis and statistics. *Python High Perform Sci Comput* 14(9):1–9
4. N. Ari and M. Ustazhanov, “Matplotlib in python,” 2014 11th International Conference on Electronics, Computer and Computation (ICECCO), 2014, pp. 1–6, doi: <https://doi.org/10.1109/ICECCO.2014.6997585>.
5. Barrett P, Hunter J, Todd Miller J, Hsu J-C, Greenfield P (2005) matplotlib—a portable python plotting package
6. Ranjani J, Sheela A, Meena KP (2019) Combination of NumPy, SciPy and Matplotlib/PyLab—a good alternative methodology to MATLAB—a comparative analysis. In: 2019 1st international conference on innovations in information and communication technology (ICIICT)
7. Waskom ML (2021) Seaborn: statistical data visualization. *J Open Source Softw* 6(60):3021
8. Pedregosa F et al (2011) Scikit-learn: machine learning in python. *J Mach Learn Res* 12:2825–2830
9. Hao J, Ho TK (2019) Machine learning made easy: a review of Scikit-learn package in python programming language. *J Educ Behav Stat.* 107699861983224. <https://doi.org/10.3102/1076998619832248>
10. Abadi M et al (2016) Tensorflow: large-scale machine learning on heterogeneous distributed systems. arXiv preprint. [arXiv:1603.04467](https://arxiv.org/abs/1603.04467)
11. Girija SS (2016) Tensorflow: large-scale machine learning on heterogeneous distributed systems. Software available from tensorflow. org 39, no 9
12. Imambi S, Prakash KB, Kanagachidambaresan GR (2021) PyTorch. Programming with TensorFlow. Springer, Cham, pp 87–104
13. Goldsborough P (2016) A tour of tensorflow. arXiv preprint. [arXiv:1610.01178](https://arxiv.org/abs/1610.01178)
14. Manaswi NK (2018) Understanding and working with Keras. Deep learning with applications using python. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-3516-4_2
15. Team TTD et al (2016) Theano: a python framework for fast computation of mathematical expressions. arXiv preprint. [arXiv:1605.02688](https://arxiv.org/abs/1605.02688)
16. Bergstra J, Breuleux O, Bastien F, Lamblin P, Pascanu R, Desjardins G, Turian J, Warde-Farley D, Bengio Y (2010) Theano: a CPU and GPU math compiler in Python. In: Proceedings of the 9th python in science conference, vol 1, pp 3–10, 1–5. <https://doi.org/10.1109/ICIICT.1.2019.8741475>
17. Ketkar N, Moolayil J (2021) Deep learning with python. <https://doi.org/10.1007/978-1-4842-5364-9>
18. Bird SG, Loper E (2004) NLTK: the natural language toolkit. Association for Computational Linguistics
19. Loper E (2004) NLTK: building a pedagogical toolkit in Python. PyCon DC 2004
20. Singh A, Ramasubramanian K, Shivam S (2019) Building an enterprise chatbot: work with protected enterprise data using open source frameworks. <https://doi.org/10.1007/978-1-4842-5034-1>

Annual Rainfall Prediction of Maharashtra State Using Multiple Regression



Loukik S. Salvi and Ashish Jadhav

Abstract This paper presents a study of Indian rainfall and prediction of the annual rainfall in the state of Maharashtra and Konkan. The decreasing trends in seasonal rainfall and post-monsoon rainfall and increasing occurrence of the deficit rainfall years indicates the probable intensification of water scarcity in many states and sub divisions of India. Rainfall serves a major source of water only when it is conserved, thus a proper analysis and estimation of rainfall globally is of utmost importance. In an agricultural country like India, where the majority of agriculture is rain dependent rainfall prediction can help to understand the uncertainty in rainfall pattern which may affect the overall agricultural produce. The present study consists a descriptive analysis of annual rainfall in India from 1950–2020, this visualization may prove helpful for deciding the right model for prediction. This study is aimed at finding the most apt model for making accurate prediction for the rainfall dataset. Two machine learning model and a neural network model are implemented and their results are compared. The performance of the results was measured with MSE (mean squared error), RMSE (root mean square error), MAE (mean absolute error). The machine learning models showed high level of deviation as the time series data in use was highly inconsistent, on the other hand the neural network showed better efficiency due to the local dependency in the model which helps it to learn and perform better.

Keywords Rainfall forecasting · SVR · ANN · Linear regression

1 Introduction

The process of analyzing and predicting the probability of precipitation and forecasting of rainfall in future along with estimating the amount of rainfall in specific

L. S. Salvi (✉) · A. Jadhav
Ramrao Adik Institute of Technology, Nerul D.Y. Patil Deemed to be University, Navi Mumbai,
India
e-mail: loukiksalvi96@gmail.com

A. Jadhav
e-mail: ashish.jadhav@rait.ac.in

regions is called Rainfall Prediction. Alongside the likelihood of precipitation in that particular district, it likewise considers the precipitation volume assessment, exactness of forecast, mistake in expectation and. It is ready by the forecasters through social occasion, researching, affirming, showing, reproducing and investigating the different meteorological data and limits available. Precipitation is one of the six intrinsic bits of environment assumption and is moreover the most basic parts in the hydrologic. Storm in India is critical as it gives shape to its farming and economy. Because of the intrinsic intricacy of the actual cycles related with the forecast of Indian Summer Storm Precipitation (ISMR), it is perhaps the most confounding logical errand. However, research related to monsoon has improved and advanced significantly due to the ever increasing amount of data made available from the satellites, improved understanding of the processes, and enhanced computing resources. As an impact of changing climate, the spatiotemporal distribution of precipitation is getting modified since the last few decades. This has resulted into frequent droughts and floods within a spatial distance of a few hundred kilometers. Quantitative forecast of every day precipitation is a difficult assignment and is huge for a considerable length of time and functional applications. Over the most recent thirty years, the precision of rainstorm gauges has improved and the methodical mistake with conjecture length in the medium reach has decreased. In any case, the worry for precipitation conjecture for the Indian rainstorm locale stays as the capacity for jungles is still lower when contrasted with mid-scopes [1]. The expectation of the precipitation can be partition into two sections one being the transient forecast which considers month to month expectation and also the drawn out forecast which considers yearly expectation which ordinarily is troublesome as a result of high unconventionality and reliance of precipitation on different boundaries of precipitation. AI models are typically known to yield better proficiency in expectation in situations where the informational collection has assorted boundaries and has critical degree of consistency, though the precipitation information being a period variation and conflicting which is the reason the AI models normally don't perform well. The coming of profound learning and neural organization has filled in as a well-suited answer for such issues. The different degree of learning and neighborhood conditions in the neural organization model assist them with performing great on such fluctuating dataset with less boundaries. Please note that the first paragraph of a section or subsection is not indented. The first paragraphs that follows a table, figure, equation etc. does not have an indent, either.

2 Literature Review

(Table 1).

Table 1 Review of relevant literature

Sr. No.	Title of the paper/journal/year	Findings
1	Annual and Non-monsoon Rainfall Prediction Modelling Using SVR-MLP: An Empirical Study From Odisha [1], IEEE Access, 2020	In this paper Odisha has been considered as the study area, the monsoon and non-monsoon rainfall prediction has been carried out using a proposed SVR-MLP model. The non-monsoon rainfall being unpredictable has been given utmost importance and proper modelled dataset has been used to ensure higher efficiency
2	Deep Learning Approach for Multi Step Ahead Daily Rainfall Prediction using GCM simulation [2]	In this paper the authors have proposed the Deep learning and neural network models for predicting the rainfall. The paper states that the DL and NN models show better efficiency over the traditional ML models for predicting the rainfall. The suggested model has also integrated the GSM simulator as the rainfall dataset has many meteorological variables
3	Forecasting of nonlinear time series using ANN [3], Future Computing and Informatics Journal (2017)	This paper imparts the importance of fuzzy logic over the common neural networks for recognizing the behavior of nonlinear or dynamic time series, in the proposed study the NARMA and ARNN are deployed and results are compared
4	Multi-model Prediction of Monsoon Rain Using Dynamical Model Selection [4], IEEE Transactions, 2016	In this paper the author has proposed a dynamic ensemble model for rainfall prediction, rainfall data being non-linear and time variant would need a dynamic model to study the irregularities
5	Summer Monsoon Rainfall Variability Over Maharashtra, India [5], Article in Pure and Applied Geophysics, 2012	This article presents a brief study of rainfall in Maharashtra for the last 30 years. The study has been conducted on the basis of the rainfall data from the meteorological stations in Maharashtra and the satellite images of cloud study. It has been observed that the rainfall has been irregular in the past ten years

3 Regression

Regression is module of data science which takes a statistical approach to solve the problems. Regression is categorized under the supervised learning approaches where strategies are made to predict the future variables. While calculating the future (unknown) variables, the known quantities and their relationships are taken into consideration and connections with the unknown variable are build. Simple regression is given by $A = X + X1(B)$, where A_n is the reliant variable, whose worth is to

be anticipated and B is the free factor whose esteem consistent worth and X1 is the improved coefficient.

3.1 Support Vector Machines (SVM) for Regression

The utilization of SVM for relapse is known as Support Vector Regression (SVR) [5]. This calculation consequently changes ostensible qualities over to numeric qualities. Input informational collection must be standardized prior to preparing start, either consequently (by apparatus arrangement or prearranging) or physically by the client (informational collection standardization). SVR observes a best fit line which decreases the mistake of the expense work. Just those occasions (Support Vectors) in the preparation informational collection are picked which are closest to the line with least expense. An edge must be made around the line for better change of forecast and afterward the information might be projected into higher layered space for better expectation and adaptability. The expense work limits the boundaries over the dataset.

Bit Functions are utilized to deal with the high dimensionality of the component space. Appropriate choice of Kernel capacity can create more powerful outcome or exactness in least time in this way expanding productivity of the model. Weka instrument utilizes different parts to accomplish this errand [2].

3.2 Artificial Neural Network

Artificial Neural Organization involves various processors working in equal and organized in levels. The principal level acknowledges the crude information. Rather than crude info, each successive level gets the result from the level going before it. The last level creates the result of the framework. The two significant conversations against ANN is that its asset concentrated and its results are difficult to decipher.

ANN is considered whenever the computational assets are not a constrictive and cost-restrictive. The ANN functions admirably when the informational collection utilized for preparing is colossal which the situation in precipitation dataset is normally. Besides the ANN model has stowed away layers and critical measure of neighborhood conditions which assists it with learning the conflicting examples and perform better on the time series information.

4 Study Area

The area of Maharashtra lies on the western side of India. Maharashtra lies between $19^{\circ} 36' 4.2984''$ N scope and $75^{\circ} 33' 10.7244''$ E longitude. The state is bifurcated

into 35 area and four meteorological locales. These sub divisional bifurcations are (1) Vidarbha (2) Madhya Maharashtra (3) Marathwada and (4) Konkan. The Konkan sub-division lies on the windward side of the Ghats and the other sub-divisions lie on the leeward side. Within a piece of the state is semi-very dry. Colossal assortments in precipitation in different areas of the state achieve a wide extent of climatic conditions. In light of geological Maharashtra gets most outrageous precipitation in July (33% of SW storm precipitation) followed by August (28% of SW rainstorm precipitation). 89% of yearly precipitation gets during southwest tempest precipitation (June–September). Most outrageous precipitation gets during the SW storm season over the districts in Konkan region (2361–3322 mm) while bits of Madhya Maharashtra and Marathwada get least precipitation (454–600 mm). During the entire year there is an immense development in Blustery days in Nandurbar, Jalgaon, Raigarh, Kolhapur and Bhandara regions. However there is a basic reducing in Turbulent days in Pune, Solapur, Kolhapur, Ahmednagar, Aurangabad, Jalna, Beed, Hingoli, Nanded, Yavatmal, Wardha districts. During the period June to September there is an enormous development in Profound precipitation days in Nandurbar, Jalgaon, Raigarh, Kolhapur and Bhandara area. While it is also observed that there is an enormous decrease in significant rainy days in Pune, Solapur, Kolhapur, Ahmednagar, regions. According to the meteorological and climatic variations in.

- (1) Pre-monsoon seasonal rainfall (March–May).
- (2) South-west monsoon season or monsoon season rainfall (June–September).
- (3) Post monsoon season rainfall (October–November).
- (4) Rainfall in winter season (December-February).

In our study we have considered the annual rainfall for the state of Maharashtra from 1950 to 2020, with the annual rainfall we have also taken into consideration the normal monsoon season period which consists of June–September (4 months) since this is the period when the state of Maharashtra receives the majority of rainfall. Along with the above-mentioned period, a time period of further 8 months comprising of 5 months of January–May and 3 months of October–December is considered as the non-monsoon rainfall period.

5 Data Collection

The primary data collected for this study consists of the annual rainfall data for India for a time period between 1950–2020 from IMD. Data-set in use has 36 sub divisions and 19 attributes (individual months, annual, combinations of 3 consecutive months according to the monsoon and non-monsoon period). All the attributes have the amount of rainfall in mm.

6 Analysis of Rainfall Pattern

Many key parts of earth and human existence are subject to precipitation straightforwardly or in a roundabout way. The progressions in worldwide environment have a solid relationship with the yearly precipitation. It has been seen that the precipitation design off of late has been exceptionally unpredictable which has upset the agrarian and water the board universally. It is vital to investigate the precipitation design to comprehend and moderate the conditions of the boundaries which are straightforwardly or by implication impacted by the yearly precipitation. In the introduced study a dataset of Yearly precipitation of India from 1950–2020 has been dissected and the investigation has been addressed in the structure Reference diagrams showing appropriation of measure of precipitation, Conveyance of measure of precipitation yearly, month to month, gatherings of months, Circulation of precipitation in developments, areas structure every month, gatherings of months. The visualization of rainfall helps to make observations which can be used for creating or choosing the right predictive model for further experimentation.

Observations made from the visualization:

1. Histograms in Fig. 1 show the circulation of precipitation over months. It is Seen that there is expansion in measure of precipitation over months July, August, September all through
2. The two charts in Fig. 2 shows that how much precipitation is sensibly great in the long stretches of Spring, April, may in eastern India.
3. Scatter graph in the Fig. 3 shows the conveyance of precipitation on yearly premise, high measure of precipitation is Seen in 1950s.
4. Heat-Map in Fig. 4 shows the co-relation (dependency) between the measures of precipitation over months.
5. From above representations obviously on the off chance that measure of precipitation is high in the long periods of July, august, September then how much precipitation will be high yearly.
6. It is additionally seen that assuming measure of precipitation in great in the long stretches of October, November, December then the precipitation will be great in the general year.

7 Methodology

In the presented study firstly the analysis of rainfall was carried out followed by the prediction of annual rainfall. Three models for prediction were used and tested over the annual rainfall dataset.

1. The input data set originally had monthly and annualized rainfall for each year from 1950–2020. Data integration was performed by adding the Monsoon and Non Monsoon rainfall parameters in form of the Monthly divisions namely Jan–Feb, Mar–May, Jun–Sep, Oct–Dec.

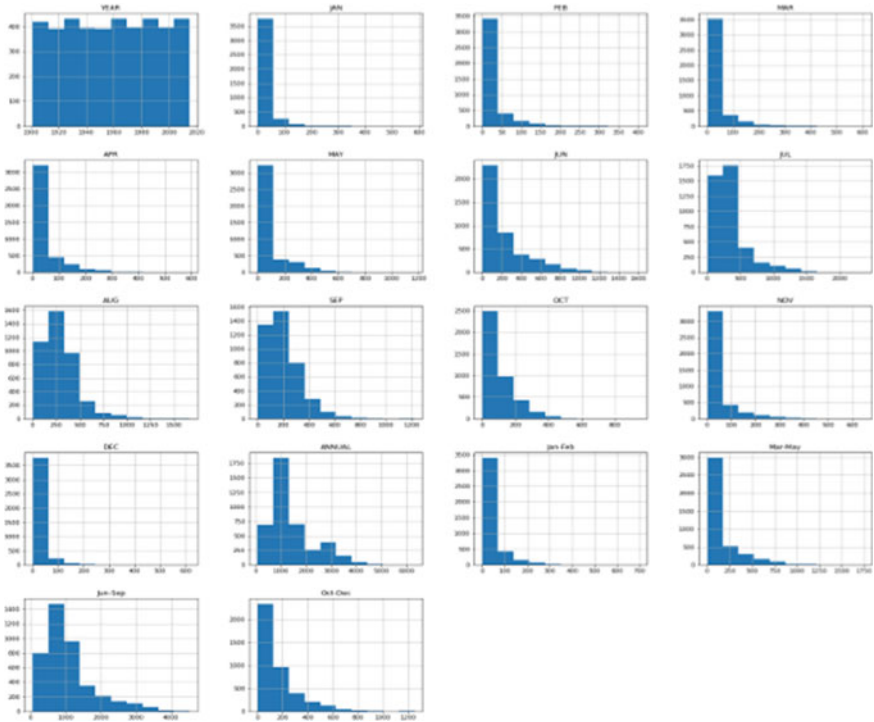


Fig. 1 Total rainfall distribution month wise from 1950–202

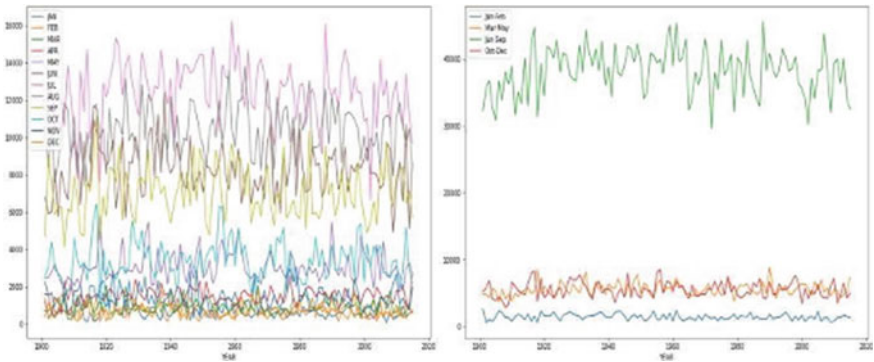


Fig. 2 Monthly rainfall in various sub-divisions of India

2. All the parameters were verified and the the missing data was augmented wherever necessary. The available data was then analyzed to create a visualization of the rainfall trends over the years. The dataset was then split into 80% train and 20% test data.

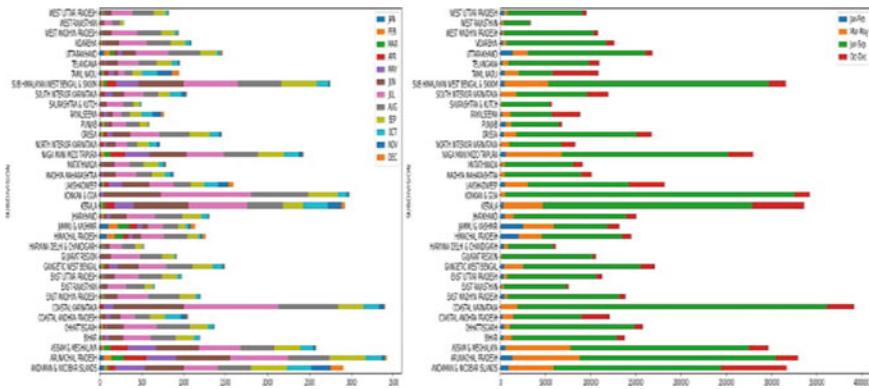


Fig. 3 Scatter plot of rainfall over the months

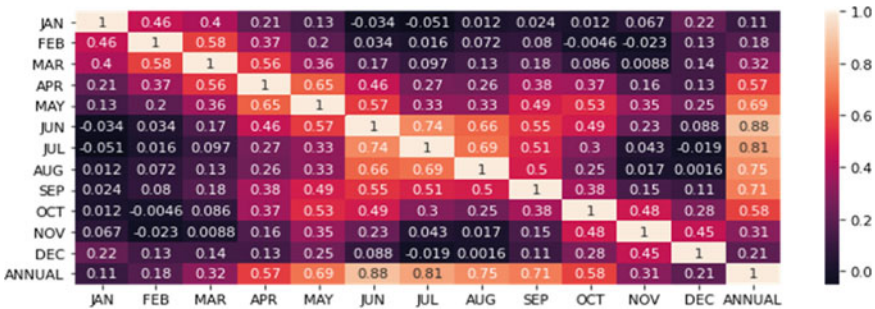


Fig. 4 Heat map for monthly and annual rainfall

3. The Linear regression model, SVR and ANN are used for making annual predictions on the dataset Two types of trainings were done once training on complete dataset and other with training on only Maharashtra and Konkan.
4. The ANN model used has been run with 10 epoch so as to yield better efficiency as it is known to have better accuracy for inconsistent data if executed with more number of epoch than usual.
5. All three models' parameters, MAE, MSE, RMSE and R2_Score are compared to find the best rainfall prediction model for annual rainfall in Maharashtra and Konkan. The models were evaluated on the basis of above-mentioned metrics.

8 Results Analysis and Discussion

The study aims at long term prediction of rainfall in Maharashtra state, which is done with the help of regression modeling. It was found that due to the long-term prediction modelling the accuracies of the models leveraged showed distinct results.

It has been observed that the long-term prediction usually is uncertain because of the data size and time accumulation and lack of information. In the long term time series prediction there are many limitations ranging from the dearth of data to less accurate results, but the deep learning techniques can be leveraged to introduce localization and improve the impact of the long term prediction. The deep learning models leveraged were able to learn directly the complex and arbitrary mapping in the input side and support and supplement the hypothesis on the output end (Table 2).

The tabular comparative result analysis mentioned in the following table shows the parametric performance of the three algorithms. The predictions made were on the basis of the annual rainfall, monsoon and non-monsoon sub-divisions The MAE values for the three models show the deviation of the predicted values from the actual value, which is higher for linear regression and SVR whereas ANN has showed comparatively less deviation. Similarly, the RMSE value tells us that the predictions made by LR and SVR are discrete and away from the best fit line. ANN has a better efficiency of prediction as the even the R2_Score for this model is nearer to 1 which is considered ideal for any predictive model. The rainfall data is discrete and nonlinear which is why the two machine learning models have showed less efficiency in prediction, the efficiency for ANN is better but can't be said satisfactory (Figs. 5, 6 and 7).

Table 2 Prediction performance

Sr. No.	Algorithm (model)	MAE	MSE	RMSE	R2_Score
1	LR	96.324	21,919.821	148.05	0.405
2	SVR	127.160	41,470.85	203.64	-0.1243
3	ANN	76.324	19,490.114	139.44	0.6726

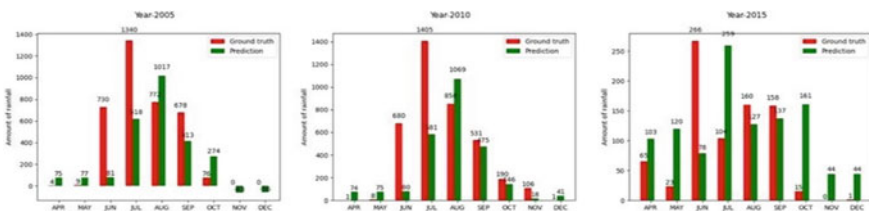


Fig. 5 Prediction of Year 2005, 2010, 2015 for Maharashtra and Konkan using Linear Regression

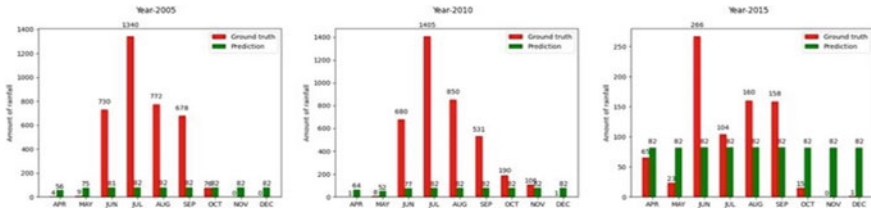


Fig. 6 Prediction of Year 2005, 2010, 2015 for Maharashtra and Konkan using SVR

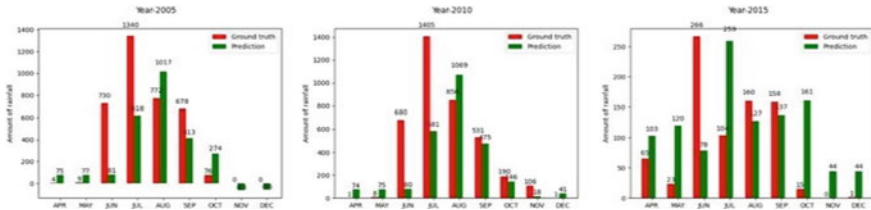


Fig. 7 Prediction of Year 2005, 2010, 2015 for Maharashtra and Konkan using ANN

9 Conclusion

Forecasting rainfall often can be helpful to make decisions regarding agro-based fields, having prior knowledge about the rainfall can help mitigate and foresee problems. In this paper a long term forecasting of rainfall in Maharashtra is done with the help of predictive modelling and deep neural networks, this study also provides a practical proof which supplements the fact that for annual rainfall in Maharashtra and Konkan is variable and unevenly scattered over the monsoon and non-monsoon period. The visualization of Indian rainfall made it clear that there is a lot of variance and discreteness in the rainfall pattern. To produce desired load forecasts, three forecasting techniques i.e. LR, SVR and ANNs were considered for evaluation by using multiple performance metrics. Significant weather profiles from eight different cities were selected to develop a synthetic weather station.. The results yielded by the two machine learning models (linear regression and SVR) depict the incompetency of the machine learning models for prediction of rainfall due to fluctuations in rainfall. It is also observed that the neural network model used for experimentation has performed better on the nonlinear time series data even though the number parameters in the dataset were limited on the basis of the evaluation metrics such as MAE, MSE, R2_score. In future, multiple prospects of this research can be explored for further development. In future experimentation more powerful neural network model such as LSTM (Long Short-Term memory) and RNN can be implemented on the non- linear data in order to get more accurate results for prediction and analysis. Even the dataset can be made more modular by adding some more parameters to obtain better results.

References

1. Zhang X, Mohanty SN, Parida AK, Pani SK, Dong B, Cheng X (2020) Annual and non-monsoon rainfall prediction modelling using SVR-MLP: an empirical study from Odisha. In: IEEE access, vol 8, pp 30223–30233. <https://doi.org/10.1109/ACCESS.2020.2972435>
2. Khan MI, Maity R (2020) Hybrid deep learning approach for multi- step-ahead daily rainfall prediction using GCM simulations. IEEE Access 8:52774–52784. <https://doi.org/10.1109/ACCESS.2020.2980977>
3. Sarma C (2022) Estimation and validation study of soil moisture using GPS-IR technique over a tropical region: variability of SM with rainfall and energy fluxes. IEEE J Sel Top Appl Earth Obs Remote Sens 15:42–49. <https://doi.org/10.1109/JSTARS.2021.3127469>
4. Bhomia S, Jaiswal N, Kishitawal CM, Kumar R (2016) Multimodel prediction of monsoon rain using dynamical model selection. IEEE Trans Geosci Remote Sens 54(5):2911–2917. <https://doi.org/10.1109/TGRS.2015.2507779>
5. Ratna SB (2012) Summer monsoon rainfall variability over Maharashtra, India. Pure Appl Geophys 169:259–273. <https://doi.org/10.1007/s00024-011-0276-4>
6. Gupta R et al (2021) WB-CPI: weather based crop prediction in India using big data analytics. In: IEEE access, vol 9, pp 137869–137885. <https://doi.org/10.1109/ACCESS.2021.3117247>
7. Tealab A et al (2017) Forecasting of nonlinear time series using ANN. Future Comput Inf J. <https://doi.org/10.1016/j.fcij.2017.05.001>
8. Sherpa SF, Shirzaei M, Ojha C, Werth S, Hostache R (2020) Probabilistic mapping of August 2018 flood of Kerala, India, using space-borne synthetic aperture radar. IEEE J Select Top Appl Earth Observ Remote Sens 13:896–913. <https://doi.org/10.1109/JSTARS.2020.2970337>
9. Maitra A, De A, Adhikari A (2019) Rain and rain-induced degradations of satellite links over a tropical location. IEEE Trans Antennas Propag 67(8):5507–5518. <https://doi.org/10.1109/TAP.2019.2911376>
10. Gopalakrishnan D, Chandrasekar A (2018) On the improved predictive skill of WRF model with regional 4DVar initialization: a study with north Indian ocean tropical cyclones. IEEE Trans Geosci Remote Sens 56(6):3350–3357. <https://doi.org/10.1109/TGRS.2018.2798623>
11. Samsudin R, Shabri A, Saad P (2013) A comparison of time series forecasting using support vector machine and artificial neural network model. J Appl Sci 10(11):950–958; Trivedi SK, Dey S (2013) Effect of various kernel and feature selection methods on SVM performance for detecting email spams. Int J Comput Appl 66(21):18–23
12. Üstün B, Melssen W, Buydens L (2006) Facilitating the application of support vector regression by using a universal Pearson VII function based kernel. Chemometr Intell Lab Syst 81(1):29–40; Paras SM, Kumar A, Chandra M (2009) A feature based neural network model for weather forecasting. Int J Comput Intell 4(3):209–216
13. Mishra AK (2013) A new technique to estimate precipitation at fine scale using multifrequency satellite observations over indian land and oceanic regions. IEEE Trans Geosci Remote Sens 51(7):4349–4358. <https://doi.org/10.1109/TGRS.2012.2226733>
14. Kumarasiri AD, Sonnadara DUJ (2006) Rainfall forecasting: an artificial neural network approach. Proc Tech Sess 1–16
15. Chattopadhyay S, Chattopadhyay M (2007) A soft computing technique in rainfall forecasting. [arXiv:nlin/0703042](https://arxiv.org/abs/nlin/0703042) [online]. <https://arxiv.org/abs/nlin/0703042>; Deshpande RR (2012) On the rainfall time series prediction using multilayer perceptron artificial neural network. Int J Emerg Technol Adv Eng 2(1):2250–2459
16. Ko M, Tiwari A, Mehnen J (2010) A review of soft computing applications in supply chain management. Appl Soft Comput 10:661–674
17. Abraham ER, Mendes dos Reis JG, Vendrametto O, de Oliveira Costa Neto PL, Carlo Tolo R, de Souza AE, de Oliveira Morais M (2020) Time series prediction with artificial neural networks: an analysis using brazilian soybean production. Agriculture 10(10):475
18. Salvi L, Patil H Data mining techniques and IoT used for building a futuristic water conservation system. IOSR J Eng (IOSRJEN). ISSN (e) 2250–3021. ISSN (p): 2278–8719

19. Galit S, Koppius OR (2011) Predictive analytics in information systems research. *Mis Quar* 35(3):553–572
20. Seber G, Lee AJ (2012) Linear regression analysis. Wiley Series in Probability and Statistics
21. Montgomery DC, Peck EA, Vining GG (2015) Introduction to linear regression analysis. Wiley Series in Probability and Statistics
22. Kavitha S, Varuna S, Ramya R (2016) A comparative analysis on linear regression and support vector regression. In: 2016 online international conference on green engineering and technologies (IC-GET), pp 1–5. <https://doi.org/10.1109/GET.2016.7916627>
23. Wang Y, Du T, Liu T, Zhang L (2019) Dynamic multi-objective squirrel search algorithm based on decomposition with evolutionary direction prediction and bidirectional memory populations. *IEEE Access* 7:115997–116013. <https://doi.org/10.1109/ACCESS.2019.2932883>
24. Zhang X (2017) Prediction modeling of the population of preschool children based on logistic model. In: 2017 9th international conference on measuring technology and mechatronics automation (ICMTMA), pp 300–305. <https://doi.org/10.1109/ICMTMA.2017.0079>
25. Ma W (2020) Prediction and analysis of population aging based on computer Leslie model. In: 2020 management science informatization and economic innovation development conference (MSIEID), pp 102–105. <https://doi.org/10.1109/MSIEID52046.2020.00027>
26. Wang J, Ma X, Song R, Yan C, Luo H (2021) A novel disease prediction method based on inductive representation learning. In: 2021 IEEE 6th international conference on big data analytics (ICBDA), pp 239–243. <https://doi.org/10.1109/ICBDA51983.2021.9403168>
27. Liu J, Zhang Y, Xiao Z, Qiao T, Tan H (2015) Fast, finite, accurate and optimal WASD neuronet versus slow, infinite, inaccurate and rough BP neuronet illustrated via russia population prediction. In: Sixth international conference on intelligent control and information processing (ICICIP), pp 140–145. <https://doi.org/10.1109/ICICIP.2015.7388158>
28. Firmanuddin G, Supangkat SH (2016) City analytic development for modeling population using data analysis prediction. In: 2016 international conference on ICT for smart society (ICISS)
29. Chakraborty S, Verma P, Paudel B, Shukla A, Das S (2022) Validation of synthetic storm technique for rain attenuation prediction over high-rainfall tropical region. In: *IEEE geoscience and remote sensing letters*, vol 19, pp 1–4. Art no 1002104. <https://doi.org/10.1109/LGRS.2021.3068334>

Automated Healthcare System Using AI Based Chatbot



Akshay Mendon, Megharani Patil, Yash Gupta, Vatsal Kadakia, and Harsh Doshi

Abstract Medical care is vital to having a decent existence. Be that as it may, it is undeniably challenging to get an appointment with a specialist for each medical issue and due to the current global pandemic in the form of Coronavirus, the healthcare industry is under immense pressure to meet the ends of patients' needs. Doctors and nurses are working relentlessly to treat and help the patients in the best possible way and still, they face problems in terms of time management, technical resources, healthcare infrastructure, support staff as well as healthcare personnel. To resolve this problem, we have made a chatbot utilizing Artificial Intelligence (AI) that can analyze the illness and give fundamental insights regarding the infection by looking at the data of a patient who was previously counselled at a health specialist. This will also assist in lessening the medical services costs. The chatbot is a product application intended to recreate discussions with human clients through intuitive and customized content. It is in many cases portrayed as the most moving and promising articulations of communication among people and machines utilizing Artificial Intelligence and Natural Language Processing (NLP). The chatbot stores the information in the data set to recognize the sentence and pursue an inquiry choice and answer the corresponding inquiry. Through this paper, we aim to create a fully functional chatbot that will help the patients/users to know about the disease by simply entering the symptoms they possess. Additionally, they can also get information about certain medicine by simply typing the name of the medicine. Another additional feature is the ability of the bot to answer general questions regarding healthcare and wellbeing.

Keywords Chatbot · Artificial intelligence · Disease prediction · Healthcare

A. Mendon (✉)

Department of Electronics and Telecommunication, Thakur College of Engineering and Technology, Mumbai, India

e-mail: Akshaymendon0911@gmail.com

M. Patil · Y. Gupta · V. Kadakia · H. Doshi

Department of Computer Engineering, Thakur College of Engineering and Technology, Mumbai, India

e-mail: megharani.patil@thakureducation.org

1 Introduction

Starting with a simple yet powerful quote: “Health is Wealth” which in itself is quite self-explanatory stating that without proper and regular management of one’s health everything falls apart. It is tragic to know how a single virus can fail some of the best healthcare services around the globe. There are a ton of large-scale health institutions like Hospitals to small-scale clinics but most patients find it difficult to choose and find the best suitable for them. Computers give us data; they draw in us and help us with plenty of habits [1]. Chatbots, otherwise called jabber robots, savvy bots, conversational specialists, computerized associates, or scholarly specialists, are great representations of AI frameworks that have advanced from ML. The Oxford word reference characterizes a chatbot as a PC program that can discuss with an individual, as a rule over the web [2]. They can likewise be actual elements intended to communicate with people or different robots socially. Foreordained reactions are then created by examining client input, on text or spoken ground, and getting to important information [3]. Chatbots are possibly alluded to as the most encouraging and high-level type of human–machine collaborations. At last, these virtual specialists are engaged in super worldwide areas like medical care, banking, schooling, horticulture, and so forth [4]. In India, the Aarogya Setu, a versatile application has evolved to make consciousness of COVID-19 with the equal association of a chatbot. Notwithstanding, these bots are filling in as clinical advisors of Covid, and not a single one of them features the issues concerning far-off patients about the pandemic [5]. Similarly, the German government fostered a battle COVID courier bot [6], and the Bangladesh-based SAJIDA Foundation fostered a nCOV-19 data bot with a side effect checker and clarifications of preventive measures [7]. The users can gain information about other various diseases and small problems right from a small wound to cancer from our Chatbot.

We are now doing more than ever in the field of Artificial Intelligence in various sectors of our society. Right now the healthcare sector is in the utmost need of it. According to IBM, “Artificial intelligence leverages computers and machines to mimic the problem-solving and decision-making capabilities of the human mind.” One of the reliable and helpful applications of AI is a chatbot [8]. One of the reliable and helpful applications of AI is a chatbot. A chatbot is a conversational specialist who reenacts ongoing connections with patients/clients. It is a program that conducts a conversation via auditory or textual methods. This proposed chatbot is based on a textual method. The dataset which is going to be used will be proficient in answering most of the queries of the users/patients. Dataset is the JSON file with keys such as intention—What patient is expecting answers from the chatbot, responses, and tags (specific disease). After that, data pre-processing will be carried out using Natural Language Processing (NLP) which provides data for fitting it into different machine learning (ML) predefined algorithms. The proposed model will be integrated into a website to make the website more useful for users. Other than this, the fundamental proverb behind the readiness of this proposed framework is to help and rouse the impending youthful cluster of engineers who are intrigued to bring a plunge into the

fields of AI, Machine Learning and NLP. The entire working and use of technology is shared in detail in the following segments to help them understand the basics of the same.

2 Motivation

Issues with the availability of health care as well as mental health care collected broad consideration during the COVID-19 pandemic when admittance to care turned out to be more troublesome. Notwithstanding average boundaries to treatment, limitations, and lockdowns established to alleviate the spread of COVID-19 made inescapable disturbances to close and personal consideration [9]. If we talk about mental health care, the limitations and lockdown have made a powerful coincidence where a generally wrecked framework has fewer assets and more noteworthy interest, causing an expansion in the neglected need for psychological wellness administrations around the world. As anyone might expect, research proposes the neglected requirement for psychotherapy and guidance, notwithstanding the disturbance of conventional administrations, has expanded during the COVID-19 pandemic [10]. Chatbots have been around for a really long time. Nonetheless, the genuine buzz around this innovation didn't begin until the spring of 2016. Explanations behind the unexpected recharged interest in chatbots remember major advances for Artificial Intelligence (AI) and a significant utilization shift from online informal organizations to portable informing applications like Facebook Messenger, Telegram, Slack, Kik, and Viber [11]. As we all know that Artificial Intelligence (AI) is at the very forefront of changing various parts of our lives by adjusting the manner in which we dissect data and further developing dynamic through critical thinking, thinking, and learning. Machine Learning (ML) is a subset of AI that further develops its exhibition in view of the information given to a conventional calculation as a matter of fact as opposed to characterizing rules in customary methodologies [12]. Progressions in ML have given benefits as far as exactness, direction, speedy handling, cost-viability, and treatment of mind boggling information [13].

With digitization of medical care and developing impact of AI, analysts recognized chatbot's capability to work on patients' openness to medication, reinforce doctor patient correspondence, and help in dealing with the constant requests for different related administrations. Chatbots could have been effectively utilized in wellbeing schooling and training, frequently combined with different capabilities, for example, side effect checker, online emergency, intelligent live criticism, etc. [14]. Given these adequate advantages, it isn't business as usual that chatbots have quickly developed throughout recent many years and coordinated themselves into various fields, like amusement, travel, gaming, advanced mechanics, and security. Chatbots have been demonstrated to be especially relevant in different medical care parts that normally include eye to eye communications. With their capacity for complex exchange the board and conversational adaptability, joining of chatbot innovation into clinical

practice might lessen costs, refine work process efficiencies, and work on tolerant results [15].

3 Literature Review

The literature review focuses on articles from conference proceedings, peer-reviewed journals as well as existing chatbots. The inception of the idea of chatbots is attributed to Alan Turing in 1950 when he questioned, “Can machines think?” [16]. The first models of chatbots were intended to clear the Turing test. Turing test is a test in which an individual aimlessly asks questions to the bot as well as to a human and if the interrogator fails to distinguish between the answers from the human and bot, it is said to pass the test. In the mid-1960s, ELIZA was one of the first know chatbots developed at MIT Artificial Intelligence Library as a natural language processing program [17]. Using ‘pattern matching’ and replacement methodology it more or less mimicked how humans converse. The early users felt like they were conversing with somebody who grasped their feedback. ELIZA breezed through the Turing test and so did chatbots like Alice and Mitsuku which were influenced by Joseph Weizenbaum’s ELIZA program [18]. The program of Alice uses similar pattern matching but works on XML schema which is known as artificial intelligence markup language (AIML) which gives the protocol to converse [19]. A resumed interest in artificial intelligence and machine learning has promoted the development and use of chatbots in different fields [20] and some of the articles are discussed below. They build a text-to-text medical diagnosis bot that helps patients into a discussion about their issues and gives a solution for their diagnosis based on their symptoms and their records. But their algorithm accuracy and diagnosis accuracy were very low.

Agrawal et al. [21] built a text-to-text medical diagnosis bot that helps patients to discuss their issues and gives a solution for their diagnosis based on their symptoms and their records. But their algorithm accuracy and diagnosis accuracy were very low. Bali et al. [22] developed “Diabot” a medical chatbot using ensemble learning which is a meta-algorithm that combines a bundle of weaker models and averages them to produce a single accurate model but the accuracy was lower than the random forest model and thus the ensemble model provided less accuracy.

Ghosh et al. [23] built a chatbot to handle complex question-answering tasks, they employed a linked medical knowledge graph (developed internally) to explore the associations between all the potential medical entities recognized by the user input. The selected entities were then ranked by their strength of association with user-selected entities. The ranking of entities was also facilitated by a frequent occurrence of entities attached to the originally extracted entities. Finally, they used this mechanism to select the top-ranked symptom that is sent back using a natural language template response to the user. But this procedure took a very long time to identify the question of the user.

Rarhi et al. [24] presented a design for a virtual doctor that provides diagnoses and remedies based on the symptoms provided to the system. The chatbot they proposed

was extracting the disease from the symptoms provided by the user and based upon the criticalness of the disease the chatbot identifies whether it is a major or minor disease if the disease is major then the algorithm will propose the patient book an appointment with a doctor from the database but this chatbot can only handle the user-specified symptoms if it is present in the database. Divya et al. [25] built a bot that predicts the disease based upon symptoms provided by the patient and provides some information about the disease before consulting the doctor. It's a text-to-text-based algorithmic process which means it's a text-to-text classification model.

Kumar et al. [26] proposed a system that is based on symptom mapping and the chatbot finds out the solution to the patient's diseases by providing them the way to cure and drugs to consume by scanning their symptoms and if the disease is major it redirects to a page where you can have an appointment with the doctor. There is a database of all doctors according to their qualifications from where the data will be fetched and will be sent to users according to their profiles. There are 3 modules of this chatbot: a collection of users' symptoms, matching with the chatbot database, and curing it. In order to achieve an accurate diagnosis finite graph is used. The string searching algorithm is used to extract symptoms however most of the time the user needs to say exact symptoms in order to get the desired diagnosis otherwise the conversation will go into a loop and start the conversation all over again. The dataset and mapping are accurate but simple and small at the same time. Datasets trained are well and the use of NLP is pinpoint.

4 Methodology

The general layout for the development of any chatbot is relatively very straightforward. There are 4 fundamental stages for all types of chatbots. Processing the input, understanding the input, generating a response and selection of a response are the 4 stages [1]. The architecture used for the development of the chatbot is illustrated in Fig. 1. The user has to first make a request in text format which is processed and interpreted by the chatbot. After the request is processed, the chatbot will store this information or it will request for more information for clarity of the user's request. Once the chatbot understands the request, the data concerned to the request is recovered from the data set [2]. Our proposed chatbot would predict diseases when the user enters symptoms. The user has to enter symptoms one by one followed by comma and the chatbot will predict the most accurate disease affiliated to those symptoms. Secondly, if the user wants to get information about a particular drug then simply providing the name of the medicine would yield the user it's full name, price in Indian Rupees, side effects as well as what exactly is the drug used for. Thirdly, the user can also ask general question answers relating to health concerns as well. For example, the user can ask "How do you treat bruises?" and get answered with the most appropriate answer by the chatbot.

The flow chart illustrated in Fig. 2 explains the methodology for the detection of symptoms and prediction of diseases. The flow of the work can be broken down

Fig. 1 General chatbot architecture

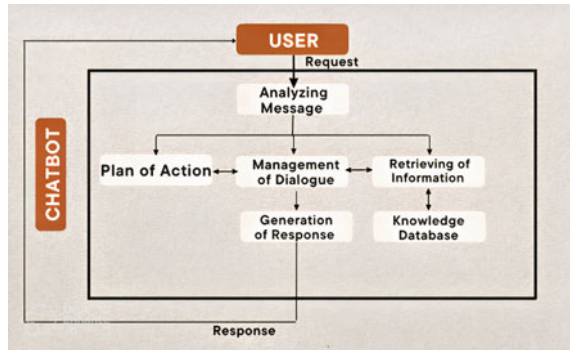
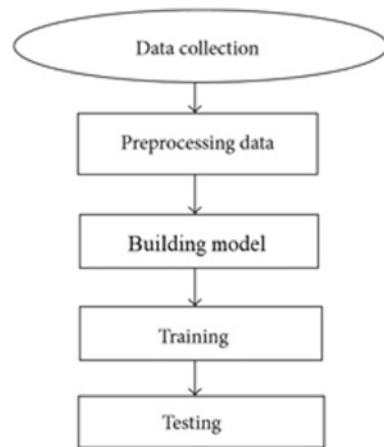


Fig. 2 Flowchart to build a model using machine learning



into 3 stages. Getting the appropriate dataset, pre-processing of text using natural language processing and feeding the data to machine language model for training and testing are the three essential stages that would be further elaborated in the later section.

4.1 Data Collection

The major purpose of datasets related to healthcare is to identify the data elements to be collected for each patient and to provide uniform definitions for common terms. There are different datasets used for this paper for different functions of the chatbot. The datasets were solely prepared and extracted from different website where they conducted different surveys and interviews with health professionals as well as patients. First dataset involves questions and answers type of conversation between the chatbot and the user which helps to know what the user wants and

Table 1 Description of intent, tag and response

Keyname	Description
Intent	Queries asked by the user
Tag	Corresponding disease with respect to Intent
Response	Answers provided by the chatbot

how the user wants to solve it. The dataset is the json file with different keys such as INTENT-Which are different ways of asking questions to chatbot or in simple words the intention of patient; RESPONSE-Which stores the responses of particular questions asked by the patient related to particular disease; TAG-Tag represents the disease name as shown in Table 1. The dataset contains 80 different tags thus capable of answering questions of 80 different diseases. This dataset is used for answering basic queries of patient's related to common/basic remedies of a disease and detailed information of parameters used in various health tests such as blood test, urine test, vitamin test, etc. The dataset contains 10–12 questions of a particular tag thus giving the model sufficient amount of data to correctly predict the intention of users.

The second dataset is an open source dataset which can be acquired on Kaggle website name as Disease Symptom Prediction. This dataset is used to detect disease based on the symptoms provided by the user. The symptoms are marked as binary values i.e. 0 and 1. If a bunch of symptoms map to a particular disease then it is marked as 1 else it is marked as 0. This dataset has 132 parameters on which 42 different types of diseases can be predicted. Since different users can have different symptoms for a particular disease, hence we have different rows for the same prognosis to reduce the incorrectness in predicting the disease. Thirdly, for the function to display information about drugs we used a dataset available on Kaggle that was made by scraping drug list from Drugs.com.

4.2 Pre-processing the Data Using NLP

For a data or text in our case to be predictable and analyzable for different task we need to pre-process it. Preprocessing text is a method to transform text so that it can be in an efficient format ready to be used with a machine learning model for better accuracy. To preprocess the data, we have used techniques like lower casing the text, tokenization, removing stop-words and lemmatization (Fig. 3).

Lower Casing the sentences

One of the simplest and most efficient way of text preprocessing is to convert all the text to lowercase. Words like "Pain" and "pain" mean the same but are represented as different words in the vector space which results in increase in dimension, hence lowercasing the data is important. Table 2 illustrates an example of solving sparsity issue through lowercasing, where same word with different meaning are mapped to single lowercased word.

Fig. 3 Process of pre-processing

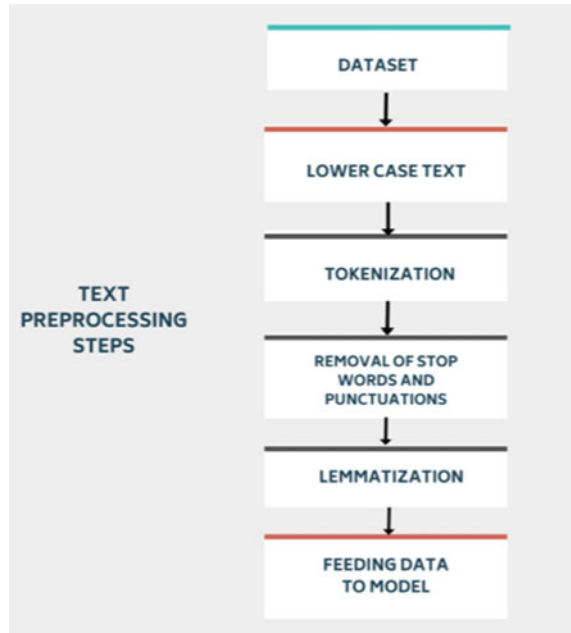


Table 2 Example of lower casing a word

Raw word	Lowercased word
Fever FeVeR FEVER	fever

Tokenization

Tokenization refers to breaking a sentence into single words whenever a white space or a punctuation is encountered. This step is very essential as it provides list of words to be dealt with for the next steps involved in text preprocessing. For example if input sentence is “I am having a stomach pain”, then the sentence is tokenized as “I”, “am”, “having”, “a”, “stomach”, “pain”.

Removal of Stop-words and Punctuations

In NLP, stop words are generally called as “useless” data that is they are not useful for the model. A stop word is a commonly used word (such as “the”, “a”, “an”, “in”) which are not at all useful for classification model and can create a noise which will effectively decrease the accuracy model. Hence all the stop words are removed from our data. For removal of punctuations Regular Expression (Regex) are used that identify punctuations and remove them. Punctuations makes the text noisy and provides no efficiency hence it’s essential to eliminate them.

Table 3 Example of lemmatization

Original word	Lemmatized word
Itching Itchy Itches	Itch
Sting Stingers Stings	Sting

Lemmatization

Doing things properly with the use of vocabulary through grouping different inflected forms of words to its base root mode with same meaning is known as lemmatization. Lemmatization extracts the correct lemma of each word. Lemmatization is a slower process but it is useful for analysis. It’s a dictionary based approach and produces a better accuracy if used. Lemmatization transforms the word without chopping the letters that gives the word a definition. Table 3 shows an example of the words ‘Itching’, ‘Itchy’ and ‘Itches’ mapped to its lemma ‘itch’.

4.3 Machine Learning Model

Naïve Bayes algorithm

Naïve Bayes model is a fast and simple classification algorithm used for a very high dimensional dataset. The classifier is based upon Bayesian classification method, which can be shown by Eq. 1 below.

$$P(A|B) = P(B|A) * P(A)/P(B) \tag{1}$$

For prediction, the algorithm uses likelihood probability. In simple term likelihood is known as reverse probability. Each word from the question asked by the user are considered as features and tag and with respect to those tags, questions are considered as labels. Now every input word is compared with all words there in the dataset and based on above Bayesian formula the different labels are associated with input question. The labels having the highest probability are considered and in this way, the algorithm works. For example, let’s say if the user asked a question as “What do I do if I am having mild fever?”, so now every word will form a feature vector space. Every word will be compared with different words in the bag and based upon that the class labels are assigned with certain probabilistic measure. Class labels assigned are ‘mild fever’ with probability of 95.67% and an—other class label is assigned as ‘cough’ with 92.78%, then class label with highest probability is considered that is “mild fever”.

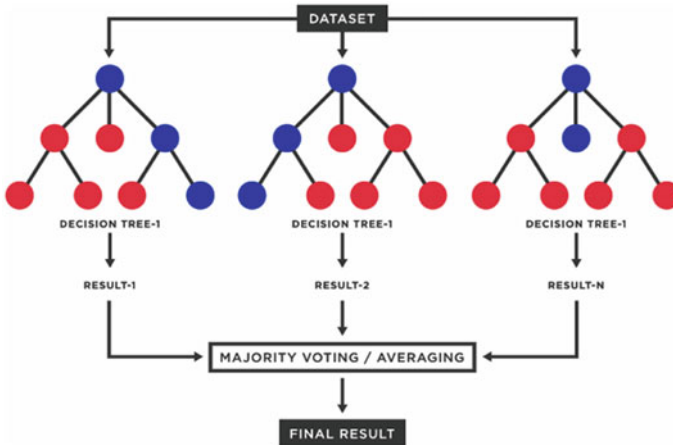


Fig. 4 Process of random forest classifier

Random Forest Classifier

Random forest classifier algorithm is a supervised learning algorithm which is used for classification as well as regression. As a name suggest it is a forest where there are trees which carries correct labels within them. So now random forest classifier algorithm selects the best tree from forest based on voting. Generally, it is considered that if there are more trees in the forest than the accuracy of the algorithm increases, which is indeed true in real life. The main advantage of using random forest classifier is saving the model from overfitting, because it itself takes the average of the prediction. As given in Fig. 4, random subsets are decision tree which consist of root node, leaf node, leaves and correct prediction of each decision tree is stored on its leaf and after through majority voting correct labels are predicted.

5 Evaluation Metrics

To build an effective machine learning model, certain evaluation metrics are used. Evaluating helps to get the useful insights whether the labels are correctly or incorrectly predicted. There are various metrics for evaluating a model such as mean squared error (MSE), Area under Curve (AOC) and confusion matrix. In our case, we would be evaluating our model on the basis of confusion matrix which is the metric used most for classification problems. In confusion matrix, the correct and incorrect predicted labels are segregated or broken with each class. Confusion matrix also gives the type of error the model is making. Figure 5 is an example of confusion matrix that uses terms like True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). TP and TN refers to the model accurately predicting and classifying whereas, FP and FN refers to the model inaccurately predicting and

Fig. 5 Example of confusion matrix

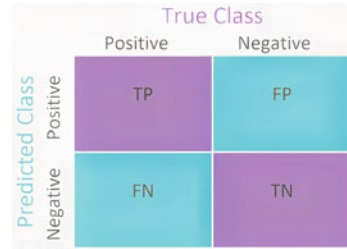


Table 4 Formulae of evaluation metrics

Metric	Formula
Accuracy	$(TP + TN)/(TP + FP + FN + TN)$
Recall	$TP/(TP + FN)$
Precision	$TP/(TP + FP)$
F1 score	$\frac{TP}{TP + \frac{1}{2}(FP + FN)}$

placing the data in wrong classes. Using these terms from the confusion matrix we can calculate the accuracy, precision, recall and F1 score that would be the evaluation metric for our model.

Accuracy is the metric which gives us the ratio of correct predictions (TP and TN) to the number of classes (TP, TN, FP, FN). But we know that, accuracy is not the best option to evaluate a model if the dataset is unbalanced. Hence, we have to use precision and recall as well. Precision is fraction of correct positive predictions (TP) out of all the positive patterns in the class (TP and FP). It is called as positive predictive value. Recall is the true positive rate and gives the fraction of positive labels that are predicted positively. Preferably, the best case of a model would be if the precision and recall having value as one which also implies that FP and FN should be zero. Hence, we would be also using the metric F1 score that uses both precision as well as recall to evaluate the result. Table 4 illustrates the formulae of all the evaluation metrics using confusion matrix terms.

6 Results and Discussion

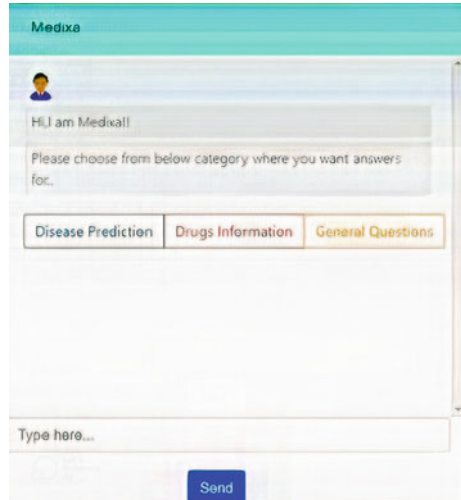
The dataset used for disease prediction is divided into 70% of the training dataset and 30% of the testing dataset. As discussed in the section Evaluation Metrics, we used confusion matrix to evaluate our classification model and evaluated the metrics like accuracy, precision, recall and F1 scores.

From Table 5 we can conclude that the Random Forest Classification model is a better model for our chatbot with accuracy of 86% and an F1 score of 82% as compared to Naïve Bayes. With this inference we proceeded with the development of our chatbot with Random Forest algorithm as the model to be used.

Table 5 Results of classification models

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Naïve Bayes	52	52	80	65
Random Forest	86	84	88	82

Fig. 6 User Interface of our chatbot



We were successful in creating the chatbot with the planned architecture and modeling as discussed in Methodology section. The program of our chatbot can be integrated with websites that are built for healthcare concerns in order to make the website more useful and give all round support to users. Figure 6 illustrates beginning of the conversation with the chatbot where the user would get to choose the type of conversation from three choices namely.

- Disease prediction.
- Drugs information.
- General question.

Figure 7 illustrates the conversation of the user and chatbot concerning information of medicine ‘Adaferin Gel’ when the user clicks on Drug information section it provides the user with the medicine’s full name, price in Indian Rupees, side effects and the usage of the medicine.

Chatbot accurately predicts the infection Impetigo in Fig. 8 when the user enters a list of symptoms. General questions concerning health was also successfully answered by the chatbot and it has been illustrated in Fig. 9.

Fig. 7 Chatbot describing about a medicine

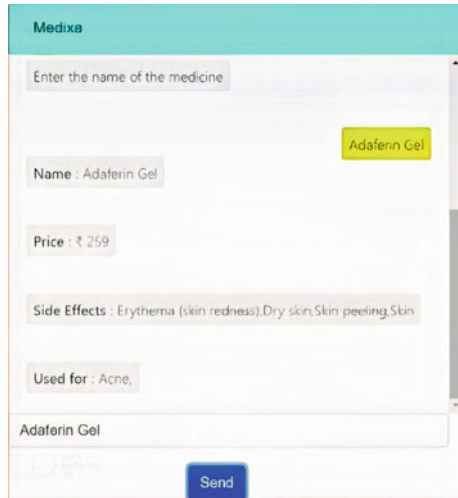
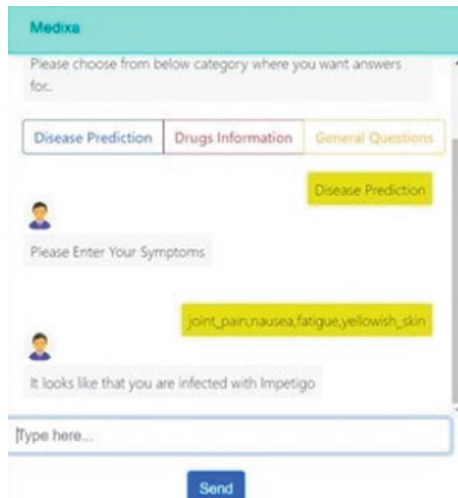


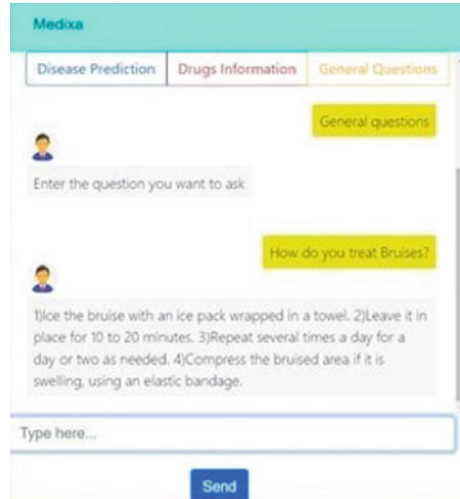
Fig. 8 The chatbot predicting a disease



7 Conclusion

Many individuals are utilizing various sorts of chatbots that work in the field of fashion, medical care, and so on and are helpful for business as expressed by mindtree.org, a web-based website that stays aware of the number and kinds of chatbots all over the planet and has revealed around 1400 chatbots presently underway. A central justification for the spike in fame is the expansion of man-made consciousness to visit applications giving an efficient and compelling response to questions. Chatbots are modified to satisfy the expectation of inquiries posed by the client while

Fig. 9 The chatbot answering general questions



at the same time noting them rapidly. Likewise, chatbot joining in any site or portable application gives the client a remunerating experience and saves a great deal of time.

References

1. Mental Health and Substance Use. The impact of COVID-19 on mental, neurological and substance use services. World Health Organization, 2020
2. Kaiser family foundation [Internet]. Unmet Need for Counseling or Therapy Among Adults Reporting Symptoms of Anxiety and/or Depressive Disorder During the COVID-19 Pandemic. State Health Facts. 2021. <https://www.kff.org/other/state-indicator/unmet-need-for-counseling-or-therapy-among-adults-reporting-symptoms-of-anxiety-and-or-depressive-disorder-during-the-covid-19-pandemic/?currentTimeframe=0&sortModel=%7B%22colId%22:%22Location%22,%22sort%22:%22asc%22%7D>
3. Gopi Battineni Nalini Chintalapudi and Francesco Amenta AI Chatbot Design during an Epidemic like the Novel Coronavirus; MDPI Open Access Journals
4. Chatbot by Oxford Learner's Dictionary online: <https://www.oxfordlearnersdictionaries.com/definition/english/chatbot#:~:text=chatbot-,noun,person%2C%20usually%20over%20the%20internet>
5. Dahiya M A tool of conversation: chatbot. Int J Comput Sci Eng 5(5):158. http://www.ijcseonline.org/pub_paper/27-IJCSE-02149.pdf
6. Luo X, Tong S, Fang Z, Qu Z (2019) Frontiers: machines versus humans: the impact of artificial intelligence chatbot disclosure on customer purchases. Mark Sci
7. Aarogya Setu Mobile App/MyGov.in. <https://www.mygov.in/aarogya-setu-app>
8. How Governments Worldwide are Using Messaging Apps in Times of COVID-19'. <https://www.messengerpeople.com/governments-worldwide-covid-19/#Germany>
9. SAJIDA Foundation and Renata Ltd. Team up to Tackle the COVID-19 Pandemic/Dhaka Tribune'. <https://www.dhakatribune.com/feature/2020/04/06/sajida-foundation-and-renata-ltd-team-up-to-tackle-the-covid-19-pandemic>
10. ARTIFICIAL INTELLIGENCE by IBM Cloud Education online: <https://www.ibm.com/cloud/learn/what-is-artificial-intelligence>

11. Chatbots: Changing user needs and motivations by Petter Bae Brandtzaeg, Asbjørn Følstad: <https://interactions.acm.org/archive/view/september-october-2018/chatbots>
12. Kersting K (2018) Machine learning and artificial intelligence: two fellow travelers on the quest for intelligent behavior in machines. *Front Big Data* 19(1):6. <https://doi.org/10.3389/fdata.2018.00006>
13. Sathya D, Sudha V, Jagadeesan D (2020) Application of machine learning techniques in health-care. In: *Handbook of research on applications and implementations of machine learning techniques*. IGI Global, Hershey, PA
14. Tang H (2019) The hindrance and motivations of chatbot services in healthcare [Blog post]. <https://ai-med.io>
15. Laranjo L, Dunn A, Tong H, Kocaballi A, Chen J, Bashir R, Surian D, Gallego B, Magrabi F, Lau AY, Coiera E. Conversational agents in healthcare: a systematic review. *J Am Med Inform Assoc.* 25(9):1248–1258. <https://doi.org/10.1093/jamia/ocy072>. <http://europepmc.org/abstract/MED/30010941.5052181>
16. Machinery C (1950) Computing machinery and intelligence—AM turing. *Mind* 59(236):433
17. Weizenbaum J (1966) ELIZA—a computer program for the study of natural language communication between man and machine. *Commun ACM* 9(1):36–45. <https://doi.org/10.1145/365153.365168>
18. Hwerbi K (2020) An ontology-based chatbot for crises management: use case coronavirus. arXiv preprint [arXiv:2011.02340](https://arxiv.org/abs/2011.02340)
19. AbuShawar B, Atwell E (2015) ALICE chatbot: trials and outputs. *CyS* 19(4) <https://doi.org/10.13053/cys-19-4-2326>
20. Shriver B, Smith B (1998) *The anatomy of a high-performance microprocessor: a systems perspective*. Wiley-IEEE Computer Society, Hoboken, New Jersey
21. Agrawal M, Cheng JL, Tran C (2017). What’s up, doc? a medical diagnosis bot
22. Bali M, Mohanty S, Chatterjee S, Sarma M, Puravankara R (2019) Diabot: a predictive medical chatbot using ensemble learning. *Int J Recent Technol Eng* 08(02):6334–6340
23. Ghosh S, Bhatia S, Bhatia A (2018) Quro: facilitating user symptom check using a personalised chatbot-oriented dialogue system. *Stud Health Technol Inform.* 252:51–56 PMID: 30040682
24. Rarhi K, Bhattacharya A, Mishra A, Mandal K (2017) Automated medical chatbot (December 20, 2017). Available at SSRN: <https://ssrn.com/abstract=3090881> or <https://doi.org/10.2139/ssrn.3090881>
25. Divya S, Indumathi V, Ishwarya S, Priyasankari M, Kalpanadevi S (2018) A self-diagnosis medical chatbot using artificial intelligence
26. Anil Kumar S, Vamsi Krishna C, Nikhila Reddy P, Rohith Kumar Reddy B, Jeena Jacob I (2020) Self-diagnosing health care chatbot using machine learning. *Int J Adv Sci Technol* 29(05):9323–9330. <http://sersc.org/journals/index.php/IJAST/article/view/19027>

Winner Prediction of Football Match Using Machine Learning



Shailja Jadon , Aman Jain , Prathamesh Bagal , Kunal Bhatt, and Manish Rana

Abstract Over the course of this article, a simple machine learning model for the prediction of a football match winner will be discussed. An in-depth analysis and insight of this model is presented further below. And we would go about the process of building it, the relevance of the project will also be mentioned along with its business implications. Finally, the merits and flaws of the project will be discussed along with ways in which it can be improved in future.

Keywords Machine Learning · Multivariate linear regression · Football prediction · Match outcome prediction

1 Problem Description

Machine Learning has become a rather sought—after technology among young students and even industries. Machine Learning is a key solution that can answer questions of the future. Its predictive nature has appealed the masses as it removes the ambiguity from various situations where the future is unknown. Humans tend to rely on such predictions for variety of their tasks.

The prediction of the winner of a football match is a curious problem of machine learning. Here the objective is to apply certain machine learning models on existing data such that we can predict the outcome with precision. The solution of this problem

S. Jadon · A. Jain (✉) · P. Bagal · K. Bhatt · M. Rana
Thakur College of Engineering & Technology, Mumbai 400101, Maharashtra, India
e-mail: jainamanr@gmail.com

S. Jadon
e-mail: shailjajadon2001@gmail.com

P. Bagal
e-mail: prathameshbagal2908@gmail.com

K. Bhatt
e-mail: bhatakunal04@gmail.com

M. Rana
e-mail: manishrana23@gmail.com

is not just anticipated by mathematicians interested in the sport but also huge organizations that dabble in the region of betting. Even, news rooms look forward to such statistics for their audience. Thus, proving the words in the above paragraph to be true in their intent.

For this paper, we shall be discussing the problem of prediction of match outcome. We will be presenting a detailed solution in form of a project that was built using certain information about the previous seasons of the English Premier League. The given information contains various parameters such as match date, goals made etc. This dataset can be obtained from online resources.

2 Literature Survey

Machine Learning (ML) approaches have been increasingly popular for forecasting sports results over the last two decades. In this paper, the authors provide a review of studies that have used ML for predicting results in team sport, covering studies from 1996 to 2019. They have sought to answer five key research questions while extensively surveying papers in this field. This research examines which machine learning algorithms have been most often utilized in this discipline, as well as others that are beginning to emerge with promising results. Their research highlights defining characteristics of successful studies and identifies robust strategies for evaluating accuracy results in this application domain. Their study considers accuracies that have been achieved across different sports and explores the notion that results of certain team sports could be more difficult to predict than others. Finally, their study uncovers common themes of future research directions across all surveyed papers, looking for and proposing gaps and opportunities for future researchers in this domain [1].

Several efforts are targeted towards increasing the accuracy of the prediction results of the soccer match. The researchers planned numerous models via implement completely different ML algorithms. Razali et al. [2] prepared a theorem stratified model that predicts soccer results. Their model relied on the goals that each groups scored in every match. Min et al. [3] provides a dynamic system for predicting the results of football matches. This dynamic structure called the FRES system comprises of two main components: theorem supported rules and therefore the theorem network element. Therefore, the FRES methodology could be a mixture of 2 ways that job along to predict the outcomes of soccer matches.

Moreover, the FRES methodology has conjointly been introduced in-game time-series approach, that permits prediction additional sensible. Nonetheless, the FRES program needs decent professional experience so as to be controlled. Constantinou [4] has established a soccer prediction model called pi-rating to supply projections on the result of football matches, whether or not home win, draw or away winfor EPL matches throughout the 2010/2011 seasons, which mixes objective data and subjective data like team strength, team form, psychological impact and fatigue.

Jan and Lit [5] area unit increasing work by Maher on the statistical distribution, demonstrating the offensive and defensive power of the goal score distribution. Koopman and Lit area unit developing an applied math model for the study and estimation of the outcomes of soccer matches, which assumes a quantity distribution of Poisson with coefficients of intensity that adjust at random over time.

Predicting Outcome of Indian Premier League (IPL) Matches Using Machine Learning by Rabindra Lamsal and Ayesha Choudhary. Cricket, particularly the Twenty20 style, has the highest level of unpredictability, with a single over having the potential to radically swing the game's momentum. With millions of people watching the Indian Premier League (IPL), creating a model for forecasting match outcomes is a real-world challenge [6].

A cricket match is influenced by a variety of circumstances, and the elements that have a major impact on the result of a Twenty20 cricket match are discovered in this study. The total weight (relative strength) of the team is determined by each player's performance on the field. To calculate points for each player in the league, a multivariate regression-based approach is provided, and the total weight of a team is computed based on the historical performance of the players who have appeared the most for the club. Finally, a dataset is modelled based on the identified seven factors which influence the outcome of an IPL match. Six machine learning models were built and utilized to forecast the outcome of each 2018 IPL match 15 minutes prior to the start of the game, just after the toss. Three of the trained models accurately predicted more than 40 matches, with the Multilayer Perceptron topping all others with a remarkable accuracy of 71.66%.

Machine learning algorithms are employed in this work to predict the outcome of soccer matches.

Although it is impossible to account for all factors that impact match outcomes, an attempt is made to identify the most important factors, and several classifiers are evaluated to tackle the problem.

Below, the literature study is summarized in form of a comparative study (Table 1).

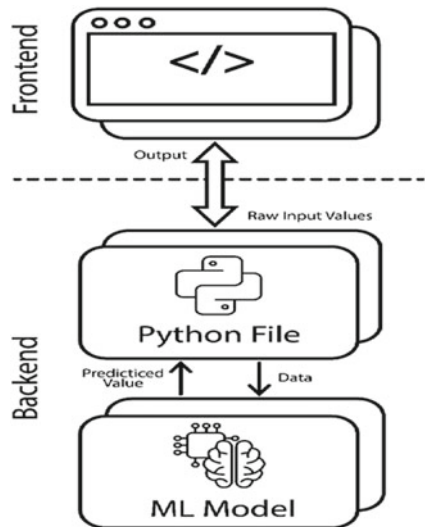
3 Methodology

To describe the approach that we have adopted for this project, it is imperative to understand the tools and resources that were available and put to use. There are two parts to this project, the frontend that was built using the simple and basic technology of HTML, CSS and bootstrap. Connecting to this is the backend, that makes it functional. It comprises of the machine learning model and some additional files to enable the model to process the input from the frontend and then provide predicted output back to the frontend. This section tailed a detail study of the same. The diagram below facilitates the understanding of how different component of the project interact with each other (Fig. 1).

Table 1 Comparative study of technical papers

Paper Title	Authors	Takeaway
Predicting outcome of soccer matches using machine learning	Albina Yezus	The study takes into account one particular league for its evaluation. It states that a simple regression model can give the results that are as good as the results given by some complex models. This study claims that a model will accuracy of 60% can be deemed a good model
The Application of Machine Learning Techniques for Predicting Results in Team Sport: A Review	Rory Bunker and Teo Susnjak	This paper claims that ANN algorithms are superior to other for predicting the outcomes of a sports match. It contains a deep analysis of various classification algorithms. It also focuses on the importance of a good dataset and mentions the challenges of finding one in the field of sports
Predicting Outcome of Indian Premier League (IPL) Matches Using Machine Learning	Rabindra Lamsal and Ayesha Choudhary	This study brings to light a solution of predicting match outcomes in IPL using multivariate regression. This is key because, this is the algorithm, that will be further used by our purpose too

Fig. 1 A visual representation of how different components of the project interact with each other



4 Dataset

4.1 Description

The dataset that we are using for building this project contains information that was partly derived from online resources and partly built by intuition and common knowledge of the given situation. The available dataset included fields such as XYZ. However, this plainly would not be very sufficient to predict the match outcomes. Before we move onto pre-processing the data and use it to derive any meaningful result, we must first be fully aware of the various parameters that are present in the raw dataset (Table 2).

4.2 Exploratory Data Analysis

Even before we start building out model, it is imperative that we analyse the details in the dataset. It is important to understand the data and how well it is structured before we pre-process the data to make it useful. Given below are the screenshots of the plots that were produced by python for the given dataset. These plots eventually helped us to understand the structure of the data and give us an insight regarding the kind of pre-processing that will be required to make the dataset efficient in its use (Figs. 2 and 3).

4.3 Data Pre-processing

Data pre-processing is defined as a process of preparing and making the raw data suitable for a machine learning model. It is very rare to come across a clean and formatted data when creating a machine learning project. And while doing any operation with data, it is vital to clean it and arrange it in a formatted way.

It is necessary that we filter our raw unstructured data into a format which can be fed to models and be processed to derive meaningful results. There are various steps involved in data pre-processing such as:

- Getting the dataset
- Importing libraries
- Importing datasets
- Finding Missing Data
- Encoding Categorical Data
- Splitting dataset into training and test set
- Feature scaling

Table 2 Description of features of dataset before pre-processing

Sr. No	Feature name	Meaning
1	Div	League Division
2	Date	Match Date (dd/mm/yy)
3	Home Team	Home Team
4	Away Team	Away Team
5	FTHG	Full Time Home Team Goals
6	FTAG	Full Time Away Team Goals
7	FTR	Full Time Result (H = Home Win, D = Draw, A = Away Win)
8	HTHG	Half Time Home Team Goals
9	HTAG	Half Time Away Team Goals
10	HTR	Half Time Result (H = Home Win, D = Draw, A = Away Win)
11	Referee	Match Referee
12	HS	Home Team Shots
13	AS	Away Team Shots
14	HST	Home Team Shots on Target
15	AST	Away Team Shots on Target
16	HHW	Home Team Hit Woodwork
17	AHW	Away Team Hit Woodwork
18	HC	Home Team Corners
19	AC	Away Team Corners
20	HF	Home Team Fouls Committed
21	AF	Away Team Fouls Committed
22	HO	Home Team Offsides
23	AO	Away Team Offsides
24	HY	Home Team Yellow Cards
25	AY	Away Team Yellow Cards
26	HR	Home Team Red Cards
27	AR	Away Team Red Cards
28	HBP	Home Team Bookings Points (10 = yellow, 25 = red)
29	ABP	Away Team Bookings Points (10 = yellow, 25 = red)

In our data, we have performed importing libraries, importing datasets, encoding categorical data and splitting the dataset for train and test purposes. In The figure given below we have demonstrated the dataset before pre-processing, the code that was used to pre-process the dataset and finally the outcome dataset from the code (Figs. 4 and 5).

Now the dataset comprises of fewer columns therefore making the regression easier to perform. The pre-processed dataset contains the information stated in the table below (Table 3).

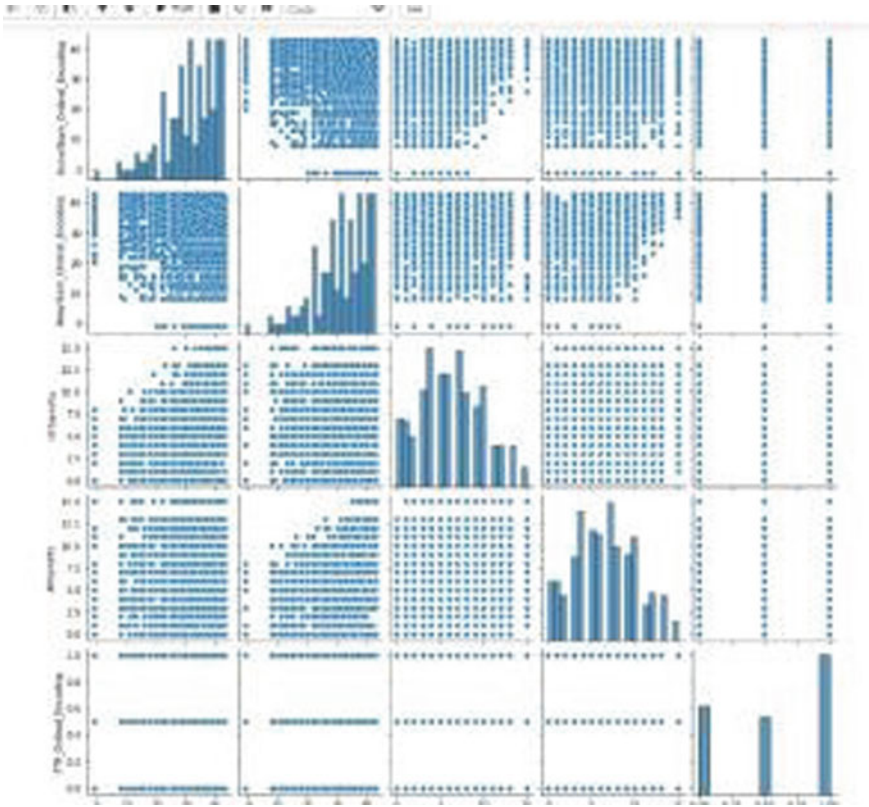
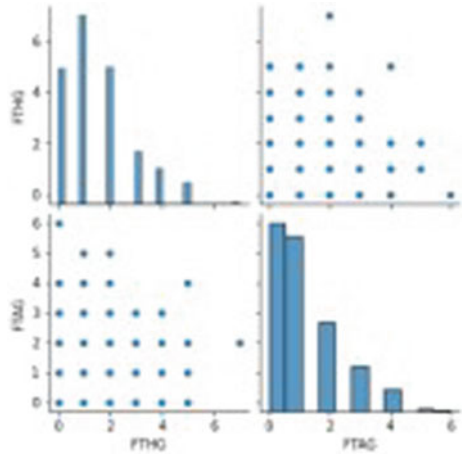


Fig. 2 Exploratory jointplot of dataset

Fig. 3 Exploratory pairplot of dataset



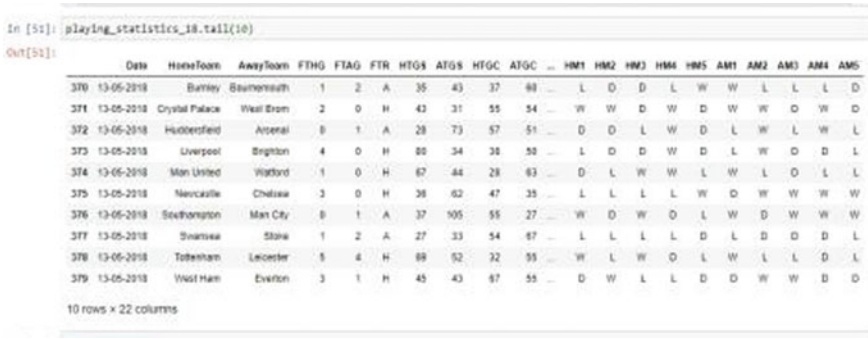


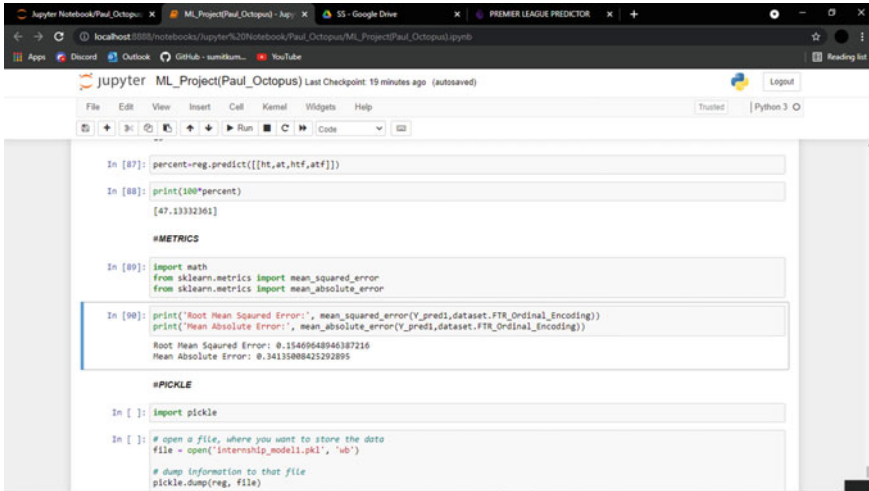
Fig. 4 Screenshot of the raw dataset prior to pre-processing



Fig. 5 Screenshot of the dataset after preprocessing

Table 3 Description of features of pre-processed dataset

Sr. No	Feature name	Meaning
1	HomeTeam_Ordinal_Encoding	Categorical Encoding for the Home Team
2	AwayTeam_Ordinal_Encoding	Categorical Encoding for the Away Team
3	HTFormPts	Recent winning/losing form of the team in the last 5 matches for the Home Team
4	ATFormPts	Recent winning/losing form of the team in the last 5 matches for the Away Team
5	FTR_Original_Encoding	Full Time Result



```
In [87]: percent-reg.predict([[ht,at,htf,atf]])

In [88]: print(100*percent)

[47.1333236]

#METRICS

In [89]: import math
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error

In [90]: print('Root Mean Squared Error:', mean_squared_error(Y_pred1,dataset.FTR_Ordinal_Encoding))
print('Mean Absolute Error:', mean_absolute_error(Y_pred1,dataset.FTR_Ordinal_Encoding))

Root Mean Squared Error: 0.15409648946387216
Mean Absolute Error: 0.34135808425292895

#PICKLE

In [ ]: import pickle

In [ ]: # open a file, where you want to store the data
file = open('interhip_model1.pkl', 'wb')

# dump information to that file
pickle.dump(reg, file)
```

Fig. 7 Code snippet

6 Testing and Training (Evaluation Metrics)

Coding is not sufficient when we talk of predictive Machine Learning models. We also need to train the model so that it can identify the relationship between the dependent and independent variables. Then we have to test out model to determine its accuracy.

$$Accuracy = \frac{No. of Correct prediction}{Total No. of Prediction Made}$$

However, in our model we do not make a train test split. This is because the relatively small amount of data available to train the model. Thus, testing is done for real time data (Fig. 7).

7 Result and Discussion

In the snippet below, we see the front-end of the project also we see a sample of how we can use the webpage for predicting the winner of a football match.

The model gives us a result that is accurate 70% times. This can be considered as a good model since it is not only simple but it also takes into account various factors of a match. The data is condensed into five columns from an overwhelming number of columns initially. The features were well used to predict the results (Fig. 8).

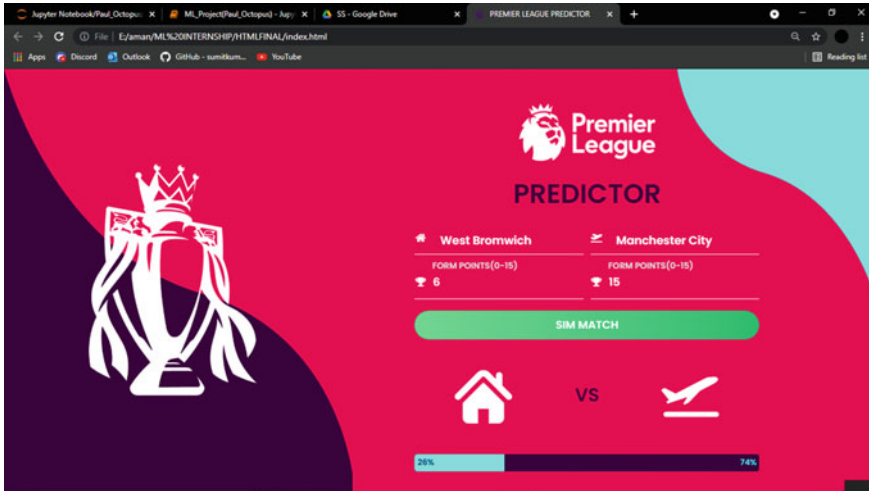


Fig. 8 Demo of the software

8 Future Scope

No project can ever be truly called complete. There is always room to do more and to do better. Such is the case of our project. In our project we have implemented a Linear regression however, other models can also be tried. We can also adopt a more advanced dataset that can help us to make better predictions. Despite several attempts we couldn't find a dataset that included the players of each team and hence we would have to work further to form a dataset in which the key players of each team are encoded and that would become an influential factor in determining the winning chances of each team. We can also develop the predictor to also predict the number of goals that would be made by each team. However, for such computation a larger dataset would be required.

9 Conclusion

To conclude, this paper has presented deep insights into what goes into making a project that amalgamates machine learning and web development. We discussed various studies to justify our choice of model which is multivariate regression. This study also presents all the steps undertaken in pre-processing, model development and training and testing of the model.

Acknowledgements Developing this project wouldn't have been possible without the support of the faculties at TCET. We are thankful to our mentor Dr. Manish Rana who has guided us throughout the technical paper. We are grateful to receive your valuable input that has enhanced this paper above

the mediocrity. We grateful for the opportunity, to present this paper. We are also thankful to the authors and resources which have provided a strong base for our research and project development.

References

1. Yezus A Predicting outcome of soccer matches using machine learning. Term paper at Saint-Petersburg State University
2. Razali N, Mustapha A, Clemente FM, Ahmad MF, Salamat MA (2018) Pattern analysis of goals scored in Malaysia super league. *Indonesian J Electr Eng Comput Sci* 11(2):718–724
3. Min B, Kim J, Choe C, Eom H, McKay RB (2008) A compound framework for sports results prediction: a football case study. *Knowl-Based Syst* 21(7):551–562
4. Anthony C, Fenton N, Neil M (2012) Pi-football: a Bayesian network model for forecasting association football match outcomes. *Knowl-Based Syst* 36:322–339
5. Jan KS, Lit R (2015) A dynamic bivariate Poisson model for analysing and forecasting match results in the English Premier League. *J Roy Stat Soc Ser A (Stat Soc)* 178(1):167–186
6. Azeman AA, Mustapha A, Mostafa SA, Abu Salim SWG, Jubair MA, Hassan MH (2020) Football match outcome prediction by applying three machine learning algorithms. *Int J Emerg Trends Eng Res* 8(1)

RaktaSeva—An App for Civilians and Blood Banks



Akash Singh, Vidhi Punjabi, Samiksha Bedekar, and Anand Khandare

Abstract According to the WHO i.e., World Health Organization, a target of 10–20 donors per 1,000 persons in any country is required to ensure adequate blood supplies. Traditionally, it is identified and observed that whenever a person has a requirement of blood, they either approach a blood bank or a blood donor with the same blood group. However, it becomes difficult to find a suitable blood donor during the time of emergency requirement of blood. Moreover, availability of the suitable blood group is not guaranteed even in a blood bank. We aim to propose an app that connects the recipient of the blood to its donor in the time of crisis and provides the flexibility of finding the blood banks near them based upon their location. The app can help to increase the possibility for a patient to get a blood donor as the requestor will be connected to all eligible donors sharing the same blood group in the same city. Thus, providing an expanded search space to the person who is in the need of blood. The application makes sure that the important crucial information of the registered users is kept private and confidential before the confirmation from both parties. The application can also be used by organizations such as blood banks or non-profit service organizations that aim to search for blood donors for their blood donation camps and create awareness to a broader mass by creating digital campaigns for their blood donation drives.

Keywords Blood · Blood banks · Donors · Requestors · Location

1 Introduction

Even with a huge population, our nation stares right into the demand–supply crisis of blood units and this prevails in several medical facilities in the country. In 2012, WHO reported that despite the demand on blood units being 12 million, only 9 units are available annually. Unavailability of proper facilities for handling the blood stock in various localities have led to wastage of blood in some regions whereas at the same

A. Singh · V. Punjabi · S. Bedekar · A. Khandare (✉)
Department of Computer Engineering, Thakur College of Engineering and Technology, Mumbai, India
e-mail: anand.khandare@thakureducation.org

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
V. E. Balas et al. (eds.), *Intelligent Computing and Networking*, Lecture Notes in Networks and Systems 632, https://doi.org/10.1007/978-981-99-0071-8_17 219

time there is lack of blood donors which causes scarcity of blood stocks in other regions of the country. Hence, with the help of our app we instill some motivation in the minds of people to voluntarily donate. There already exist a few blood donor finder applications but they require time and manpower as they just provide details and donors are needs to be contacted manually. The Problem definition is, “to develop a blood donor connect app which can help a person to connect with the potential blood donors in the vicinity of a particular hospital without the need of contacting every person individually.” Registered user (Requestor) finds all available and eligible donors in the vicinity.

2 Literature Survey

The author presented an application for voluntary blood donors in [1], the main goal of which is to notify Rh++ of the donor location on a regular basis. Rh++ is a smart information system with the goal of regulating blood donation and supply. The donors were initially confirmed, which was one of the paper’s merits. The user can donate blood after receiving authorization. The possible shortcomings which could be considered was that GPS capabilities were not used by this particular application for donations.

Eahtesam and Raaz [2] was a general health centric application that kept record of medical history of patients. With the help of Global Position System i.e., GPS it found the donors in the locality where the patient was currently present. The location of the donor was kept up to date, making it easy for the patient to find donors. This reduced the amount of memory used and the volume of contact with the user were both reduced.

In [3] when blood is required in an emergency, we can use GPS to locate a nearby blood donor. Whenever any user enters his/her blood group, the application automatically locates a nearby blood donor and send an alert request message to the potential donor. And if the first donor is unavailable, the system will automatically search for the next available potential donor available in the queue. If the donor accepts the request, the donor will receive a one-time password (OTP) to validate his or her identity (Table 1).

3 Existing Systems

3.1 *Friend2Support.org*

This application is very popular in India. Also has support for Nepal, Bangladesh. Sri Lanka. It stores the record of all the donors registered on the platform. Whenever a patient needs a donor, they can access the details of donors in their vicinity and

Table 1 Comparison of carious tables

Paper	Findings/Merits	Gaps identified/Demerits
[1]	<ul style="list-style-type: none"> – Users are verified – History of the patient is maintained – Secure 	<ul style="list-style-type: none"> – No tracking features – Accessibility issue concerned issue for the application is limited to social media
[2]	<ul style="list-style-type: none"> – Tracking is available – Minimized memory consumption and user interaction – Checks the medical records of the donor 	<ul style="list-style-type: none"> – Not secure as it gives out information – Available only as an Android Application – The donor is in contact with the hospital, but a patient in need cannot request for blood individually
[3]	<ul style="list-style-type: none"> – Use of GIS (Geographic information System) technology – Users are verified – Secure 	<ul style="list-style-type: none"> – No method of checking authenticity of users – Strenuous on the admin

contact the donors accordingly. Donors can be contacted through SMS, call, or email. But the details of the user are always accessible which can also jeopardize the user’s personal information. The users also have to be manually contacted through a given medium and there is no automatic notification option. It is not a viable option during times of emergency.

3.2 *Save Life Connect*

This app works on similar grounds of the above app. It displays fundraisers going on and people willing can contribute to the cause. The person in need of a donor creates a request which is posted on the application feed. Unless the user is connected to the internet the request is not sent or received. After receiving the request, the donor can contact the person in need. It enables a GPS tracker to make it easier for the user to track down the request. However, the user’s medical records/history is not maintained. Due to this donor receive requests even after recent donations. The reach of this application is minimum in India.

3.3 *e-RaktKosh*

This is a government approved website which focuses on Blood Banks. It stores details of blood banks all over the country and displays the updated blood unit stocks. This enables people in need of blood to access their nearby blood banks and blood availability according to their blood group. Furthermore, there is also an option for live tracking the blood bank location with the use of maps. This makes it easier for

the people to reach the blood bank in stipulated time. A blood donor can also look for camps and blood banks for blood donation through the portal.

4 Proposed Work

The system is proposed to work for mainly two stakeholders: civilians i.e., normal users and the second i.e., blood banks and service organizations. Whenever a patient need blood, he can search for registered users in his immediate area. When a person agrees to help, he or she provides their one-time location with the requestor so that the requestor can connect with the nearest donor in the shortest span of time. The system must also ensure that the donors on the list of donors are both available and eligible, implying that the user has made himself available for any such assistance and is qualified to donate blood on medical grounds. The requestor only needs to provide the patient's information, and the information about potential eligible donors of the city will be retrieved from the database. Apart from this user's data which is used to identify and fetch potential donors the database must also be leveraged with the data of blood banks across country with their complete details i.e., city, address, contact information and blood bank category. This data will be helpful for the users in order to get the details of potential blood banks in their city and in their respective locations where they can look for blood.

In order to use the portal, it is taken into consideration that important medical or private details are not asked from the users so that there could be minimal hesitation among the users to register on the portal. Only details which are asked here are only their blood group, contact number, and address, which would be requested from everyone registering on the portal.

While requesting for blood, the requestor has to submit the details of the patient and the admitted hospital's location and with a single click of a button, a system-generated SMS and a post will be distributed to all recipients' feeds. Once the recipient receives the request, he can confirm it using the options given to him, and the request can be recorded as served on the platform once the blood donation process is completed.

Blood banks can utilize the portal to create donation requests for their diminishing blood stocks and to build campaigns for their blood donation drives directly through the portal. There are various service groups that work in the field of blood donation awareness and the implementation of blood donation camps in addition to blood banks (Fig. 1).

5 Implementation

The implementation for user side includes the development of an in order to build the functionality for the user side a web application is developed. The system which

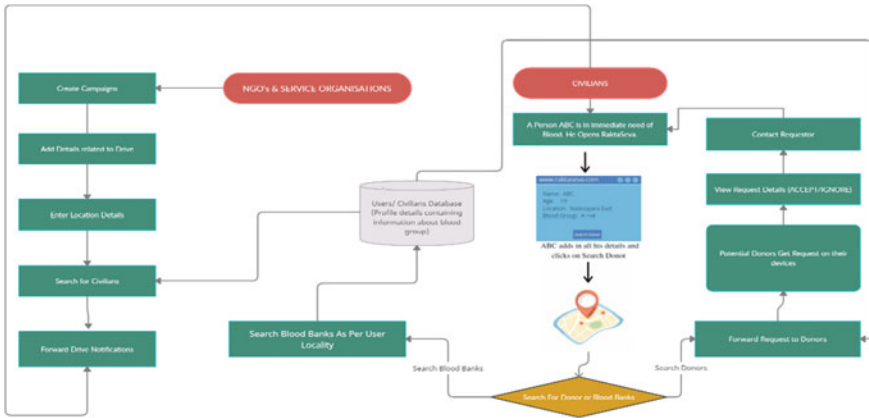


Fig. 1 Architecture overview for working principle of the proposed system

is developed registers the users on the portal and ask for the basic details of the user which includes some of the medical details like blood group and age. Now these details are used to create the database of the potential of the donors registered on the portal. These details are further used to find the donors in the closest vicinity of the hospital (Fig. 2).

Now whenever a concerned person needs blood, he/she will just fill the basic minimal details of the person for whom the blood is requested along with the hospital details to find the closest potential donors. The system will further look for the potential donors with the help of the database maintained already. Next the requester just needs to click a Message button using which an auto generated Text SMS which

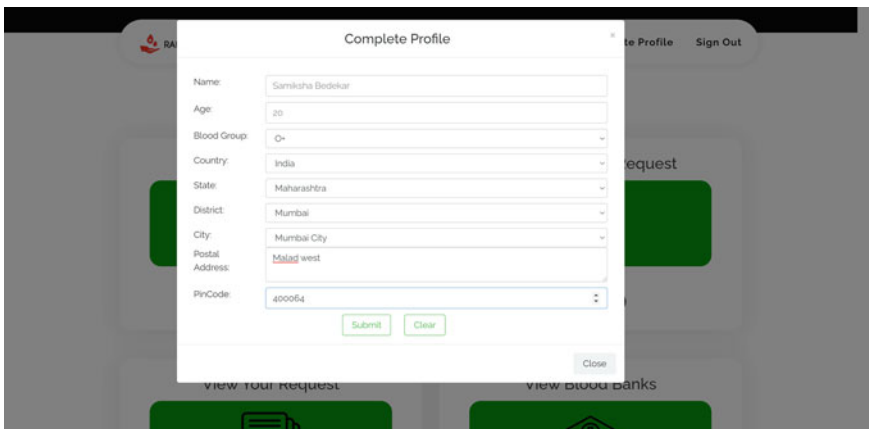


Fig. 2 Profile details requested from a registered user

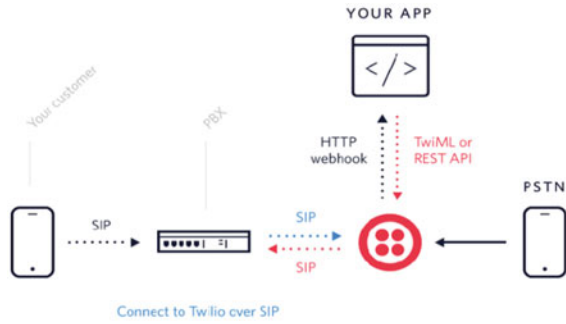
Fig. 3 A minimal help form to fill necessary details for making request

will be sent to all those users whose details are matching with the detailed specified in the request (Fig. 3).

Doing this the requester will get to know how many people were connected through his request however the details of the users are not revealed to him/her as long as authenticity of the request is validated. The users who have received the messages get the hospital details mentioned in the message through which they can check for the authenticity of the request.

The system was implemented with a python backend where the business logic was developed using Python's Django Framework based upon MVT i.e., Model View Template Architecture. MVT is a software design for developing web applications. Model acts as an interface for our data and it is the logical structure of our system and is represented by database such as MySQL, PostgreSQL. In our case we have dB SQLite which is default database setup for Django Framework. The Views in this MVT architecture are used to create link between Template files and Model Data. Views in our case plays the same role which is played by Controller in MVC Architecture. The API used here is Twilio API which is a Programmable SMS API which helps us send and receive SMS messages. The Twilio provides this API service on premium basis however as long as we use it on trial basis, we need to verify each and every person registered on portal on Twilio's portal as well (Fig. 4).

Fig. 4 Twilio API architecture



6 Result and Discussion

The implementation resulted into the development of important modules which forms the entire architecture for the proposed system. Whenever a request is placed, a text SMS consisting of all required details are sent over potential donors registered phone numbers so that any request originated through the portal reaches the potential donors even though they are not using their internet connectivity. This makes a major improvement in existing systems which are totally an online model and every aspect of it depends on online connectivity of users (Fig. 5).

Along with the text SMS received on recipient mobile numbers. A detailed request ticket is also created on the portal for recipient side where they can view the entire details, and can show their willingness to help if they want. The ticket consists of an accept request button which notifies the user that their one-time location and mobile number will also be sent along with their acceptance consent so that requestor can initiate further communication or not depending upon location feasibility of donors (Table 2).

The requestor can view the distance between their current location and donor via using google maps API which helps the requestor to understand whether the potential donor can reach to them in optimal time or can choose among the multiple volunteer donors depending upon their fast reach.

Fig. 5 Text SMS received by the potential donor

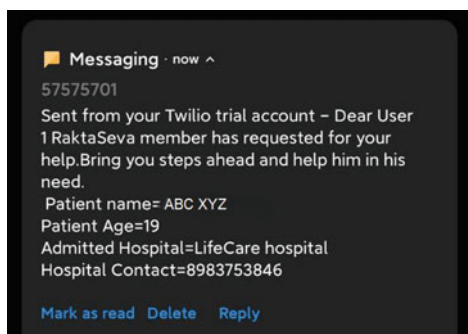
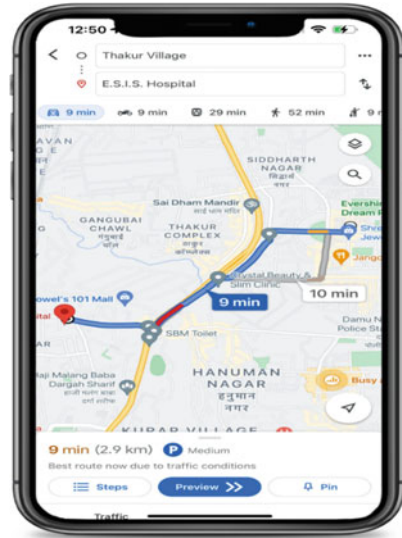


Table 2 Comparison of system functionality

Parameters	RaktaSeva	Save life connect	e-RaktaKosh
Responsive	✓	×	✓
GPS	✓	✓	✓
Blood bank data	✓	×	✓
Data privacy	✓	✓	✓
Auto requesting	✓	✓	×

Fig. 6 Estimating donor reach using Google Maps Route based feature



The median latency for the APIs like Places API, Maps JavaScript API and Direction API is well within the optimal range indicating that the request is fulfilled quickly. The Geocoding API provides a higher latency value. The 95-percentile latency indicates a higher range for the results generated by the API.

Along with this request module the blood bank list can also help the requestor to connect to their nearest blood bank and check for the required blood stock availability. The database created for the blood banks helps for quickly fetching out the blood banks details throughout and connect with them as fast as possible (Fig. 6).

7 Conclusion

In conclusion, we foresee that the dedicated applications and platforms which can help the requestors get in touch with potential donors, nearby blood banks can bring about a revolutionary change in the blood donation trends in India. Also, as the

system can work in collaborations with civilians, blood banks as well as service organizations it can provides much wider possibility for the application to scale. It will facilitate finding donors in emergencies when the blood bank is out of reach and blood banks with complete details. Furthermore, the use of GPS will enable the patients to find potential donors in their vicinity so that the process can be speedy. With one click a person in need can contact all the donors in the neighborhood. While doing so the privacy of app users is always ensured. As a portal is free to use it can reach maximum people and benefit them. Overall, this application will play a major role in saving people's lives.

8 Future Scope

The system can also be seen working with the inclusion of artificial intelligence and machine learning where donors learning patterns can be learnt and understood in order to enhance the capabilities of the system in order and connect them with the most suitable donors.

References

1. Turhan S (2015) An android application for volunteer blood donors. In: Fifth international conference on computational science, engineering and information technology
2. Eahtesam S, Raaz S (2016) An efficient android app from blood donation process. *Int J Innov Eng Technol (IJJET)*
3. Hamlin A, Mayan A (2016) Blood donation and life saver-blood donation app. Institute of Electrical and Electronics Engineers (IEEE)
4. Shinde A, Gharat A, Sakhalkar V, Chapke R (2018) RED DONATE: a blood bank android application. *J Netw Commun Emerg Technol (JNCET)*
5. Tatikonda VK, El-Ocla H (2017) BLOODR: blood donor and requester mobile application. *mHealth*
6. Chakrabarti R, Darade A, Jadhav N, Chitalkar SM (2020) Lifesaver E-blood donation app using cloud. *Int Res J Eng Technol (IRJET)*
7. Gaikwad A, Mulla N, Wagaj T, Ingale R, Gupta B, Reddy K (2018) Smart blood finder. *Int J Trend Sci Res Dev*
8. Agrawal SA, Chavan SB (2014) EMS: an android application for emergency patients. *Int J Comput Sci Inf Technol (IJCSIT)*
9. Talapatra S, Kabir R, Bappy AS (2019) Development of an online blood management system. In: *Proceedings of the international conference on industrial engineering and operations management*
10. Alfouzan N (2014) Knowledge, Attitudes and motivations towards blood donation among King Abdulaziz medical city population. *Int J Fam Med*
11. Al-Rashdi YDA, Alharbi SH, Alateeq FA, Ahmed IB, Alsogair ARAA, Aldugieman TZ, Ahmed HG (2018) Factors influencing the levels of recruitment for blood donations. *Int J Med Res Health Sci*
12. Ekanayaka EM, Wimaladharma C (2015) Blood bank management system. Technical session-computer science and technology & industrial information technology.

13. Priya P, Saranya V, Shabana S, Subramani K (2014) The optimization of blood donor information and management system by Technopedia. *Int J Innov Res Sci Eng Technol*
14. Kulshreshtha V, Maheshwari S (2012) Benefits of management information system in blood bank. *Int J Eng Sci*
15. Snigdha, Anabhavane V, Lokhande P, Kasar S, More P (2016) Android blood bank. *IJARCCCE. Int J Adv Res Comput Commun Eng*
16. Clemen Teena A, Sankar K, Kannan S (2014) A study on blood bank management system. *Middle-East J Sci Res* 19
17. Gupta N, Gawande R, Thengadi N (2015) A life saving application. *Int J Res Emerg Sci Technol*
18. Selvamani K, Ashok Kumar Rai (2015) A novel technique for online blood bank management. *Procedia Comput Sci*
19. Spyropoulos B, Botsivaly M, Tzavaras A, Spyropoulou P (2009) Towards digital blood-banking. *ITU-T Kaleidoscope: Innovations for Digital Inclusions*
20. Lunawat NM, Kshirsagar CD, Gawhande AA, Rathod RM, Thool AD, Chumble SC (2016) Blood and organ for patient using android application. *IJRET: Int J Res Eng Technol*
21. Shashikala BM, Pushpalatha MP, Vijaya B (2018) Web based blood donation management system (BDMS) and notifications
22. Kulshreshtha V, Maheshwari S. Blood bank management system in India. *Int J Eng Res Appl (IJERA)*
23. Hazzari D, Wijesekera D, Hindawai S (2014) Formalizing and verifying work flows used in blood banks. *Procedia Technol*
24. Sugijaro DP, Safie N, Mukhtar M, Sulaiman R (2016) Telehealth model information flow: a case study on laboratory information system. *Procedia Technol*
25. Sulaiman S, Hamid AAKA, Yusri (2015) Development of blood bank management system. *Procedia Soc*
26. Bhowmik A, Nabila NA, Imran MA, Rahaman MAU, Karmaker (2015) An extended research on the blood donor community as a mobile application. *Int J Wirel Microw Technol*
27. Kalem G (2015) Mobile technology in the healthcare industry for disease management and wellness. *Procedia Soc Behav Sci*
28. Chinnaswamy A, Gopalakrishnan G, Pandala KK, Venkat SN (2015) A study on automation of blood donor classification and notification techniques. *Int J Appl Eng Res*
29. Snaigdha, Anabhavane V, Lokhande P, Kasar S, More (2015) Android blood bank. *Int J Adv Res Comput Commun Eng*
30. Jenipha TH, Backiyalakshmi R (2014) Android blood donor life saving application in cloud computing. *Am J Eng Res (AJER)*

Prediction of Anemia Disease Using Machine Learning Algorithms



Aditya Dixit, Rahul Jha, Raunak Mishra, and Sangeeta Vhatkar

Abstract As we know, Red Blood Cells are the main part of blood that is responsible for the circulation of blood in the human body. Anemia is a well-known disease that is caused due to the deficiency of healthy red blood cells. Due to anemia, red blood cells are unable to supply oxygen throughout the body. This sickness can be lethal to the human body if not treated promptly. We are using machine learning techniques such as Random Forest, SVM, and others to detect anemia in a patient in this study. We can detect anemia in a patient using machine learning methods. As a result, we intend to create a classification-based ML model in which we provide the essential CBC test values for our model to predict whether a patient is anemic. With the help of machine learning techniques, we are automating the process for detecting anemia in this study work. We compared the statistical analysis of all algorithms we've utilized to predict anemia in this paper.

Keywords Anemia · Machine learning · Random forest · SVM · Naive Bayes

1 Introduction

There has been an exponential increase in the data generated through the health industry because of the remarkable advances in Technology used. Using this data, we can extract all the useful information which can then be used for analysis, recommendation, prediction and decision making. In medical science, disease prediction at the right time is important for prevention and effective treatment plan. Anemia

A. Dixit (✉) · R. Jha · R. Mishra · S. Vhatkar
IT Department, Thakur College of Engineering and Technology, Mumbai, India
e-mail: 1032190284@tcetmumbai.in

R. Jha
e-mail: 1032190307@tcetmumbai.in

R. Mishra
e-mail: 1032190325@tcetmumbai.in

S. Vhatkar
e-mail: sangeeta.vhatkar@thakureducation.org

is a disease which is caused by the deficiency of healthy red blood cells which are unable to deliver oxygen throughout the body. Anemia is highly prevalent in India. The third National Family Health Study (NFHS-3) conducted during 2005–6 found that amongst children aged 6–59 months, the prevalence of anemia is 69.5%; in rural India, the prevalence is 71.5%. The prevalence of anemia is maximum among younger children between the age of 12–17 months and 18–23 months. The prevalence of anemia in rural areas appeared to have risen since the previous NFHS (in 1998–9) [1].

Hence, it is important to take some measures to prevent the spread of anemia as much as possible using the latest advancements happening in the Tech Industry. In our study, we found out using various classifier algorithms like Random Forest, SVM, Naïve Bayes etc., we can predict the early stage of anemia so that patients can take required medicine on time and prevent anemia [2]. This project is important as, using the latest advancement in the field of machine learning, we can also make solutions in the field of medical science. This technology can be used in many areas like rural areas where health care systems are still not developed to the extent that of urban areas [3, 4].

Anemia is a disease, which needs timely treatment and early diagnosis, using machine learning we can achieve this. Machine Learning can help us overcome many different problems faced by our country in the field of medicine. Using this project, we will be able to detect whether a person or patient is suffering from anemia or not in a matter of seconds [3].

2 Problems Faced

Anemia is a growing problem amongst young children living in rural India. In Rural areas, there is a lack of proper medical treatment and experienced doctors. This leads to patients traveling long distances to visit experienced doctors for treatment. This delay ultimately leads to the disease becoming more fatal.

Also, many people avoid going to the doctor because they are scared or they can't afford it. Also, due to the lack of trained or experienced doctors in rural areas, they misdiagnose the symptoms resulting in Anemia becoming more fatal.

Anemia, also goes quite unnoticed in many people especially children, which can go unnoticed at first but suddenly become fatal in nature. To identify this, a doctor needs to go through the CBC blood test report thoroughly to identify the early stages of Anemia. Once identified, it is quite easy to cure the disease.

To tackle all these problems, we are planning to create a Machine Learning Model, using which we would make use of multiple algorithms like Naive Bayes, Random Forest, SVM, etc. and select the best algorithm using which we will create a website, where the user can simple put in their blood test parameters in our machine learning model which would then predict whether the user is suffering from Anemia or not.

Our machine learning model can predict and alert the user if the user is suffering from anemia and using which the user can be treated on time without the need of any experienced medical staff.

3 Methodology

We followed the below methodologies to make our project:

1. Taking Input Data

- Firstly we collect the dataset [5].
- Dataset should be in csv format (Fig. 1).
- We import the dataset using various python libraries like Pandas.
- Above, in our dataset, we have considered five parameters—[3, 6, 7].
 1. Gender—Gender is a very important parameter as the blood parameters and limits for both Male and Female are different and vary, so it is important to also consider this factor.
 2. MCV—MCV stands for mean corpuscular volume. Basically this blood test measures the average size of the red blood cells. Using this test we can get to know whether our red blood cells are too small or too large which can depict any blood disorder such as anemia [8].
 3. MCH—MCH is short for “mean corpuscular hemoglobin.” It’s the average amount in each of your red blood cells of a protein called hemoglobin, which carries oxygen around your body [9].
 4. MCHC—MCHC is a similar measure to MCH, MCHC stands for “mean corpuscular hemoglobin concentration”. MCHC checks the average amount of hemoglobin in a group of red blood cells. A doctor might use both MCHC and MCH in order to diagnose Anemia [10].
 5. Hemoglobin—This parameter tells us about the amount of oxygen present in our blood. It is basically a protein which has the capacity to carry oxygen throughout the body from the lungs. It is also a very important parameter in prediction of anemia. For men, anemia is typically defined as a hemoglobin level of less than 13.5 g/dl and in women as hemoglobin of less than 12.0 g/dl [11].

2. Pre-processing and Cleaning Dataset

	Gender	Hemoglobin	MCH	MCHC	MCV	Result
0	Male	14.9	22.7	29.1	83.7	Not Anemia
1	Female	15.9	25.4	28.3	72.0	Not Anemia
2	Female	9.0	21.5	29.6	71.2	Anemia
3	Female	14.9	16.0	31.4	87.5	Not Anemia
4	Male	14.7	22.0	28.2	99.5	Not Anemia
5	Female	11.6	22.3	30.9	74.5	Anemia

Fig. 1 Picture of anemia CBC dataset [5]

- For data cleaning and preprocessing, we have imported the required dataset using the pandas dataset.
- After importing the dataset and making it a dataframe, we have first converted all values into integers. Checked for null values, we didn't find any null values in our dataset.
- Next, we went ahead and checked all the number of entries and removed all duplicates.
- Now, after cleaning the data, we went ahead for data visualization (Figs. 2 and 3).

3. Feature Extraction/Feature Selection

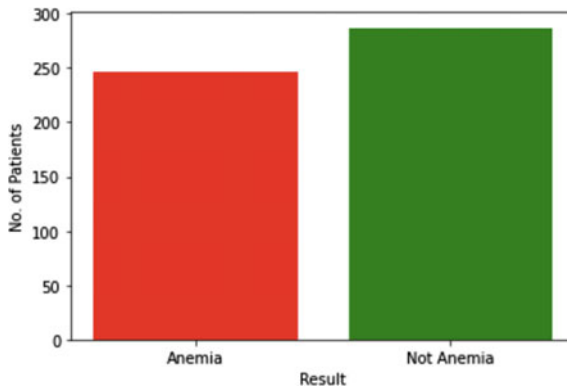


Fig. 2 Split of results in dataset [5]

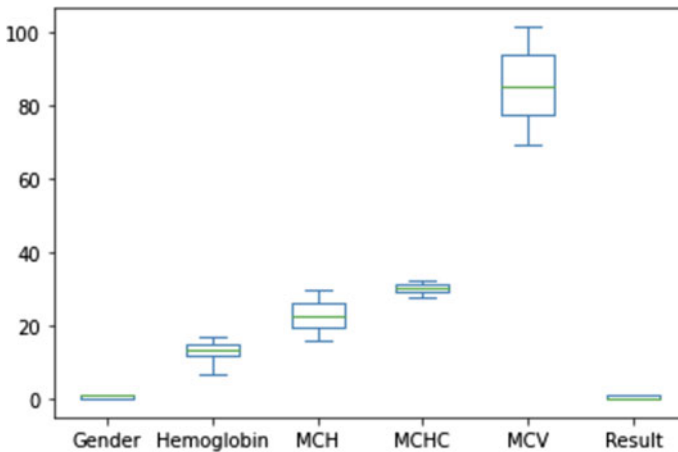


Fig. 3 Boxplot of all the parameters [5]

- As discussed above, we are using 5 features to predict whether a user/patient is suffering from anemia or not.
- We are using Gender, Hemoglobin, MCH, MCHC and MCV from the blood test reports to predict whether a user is suffering from anemia or not [3].
- After cleaning all the data, we will then Normalize the data using MinMaxScaler. MinMaxScaler transforms all the features between 0 and 1.

Here we extracted features that are required for model training (Fig. 4).

4. Apply Classification Algorithms

- After feature extraction now comes to model training.
- First of all we have divided the dataset into training and testing using a method called train_test_split(). We have divided our dataset into a 75–25% train-test split.
- Now select the classification algorithm and import it from respective libraries.

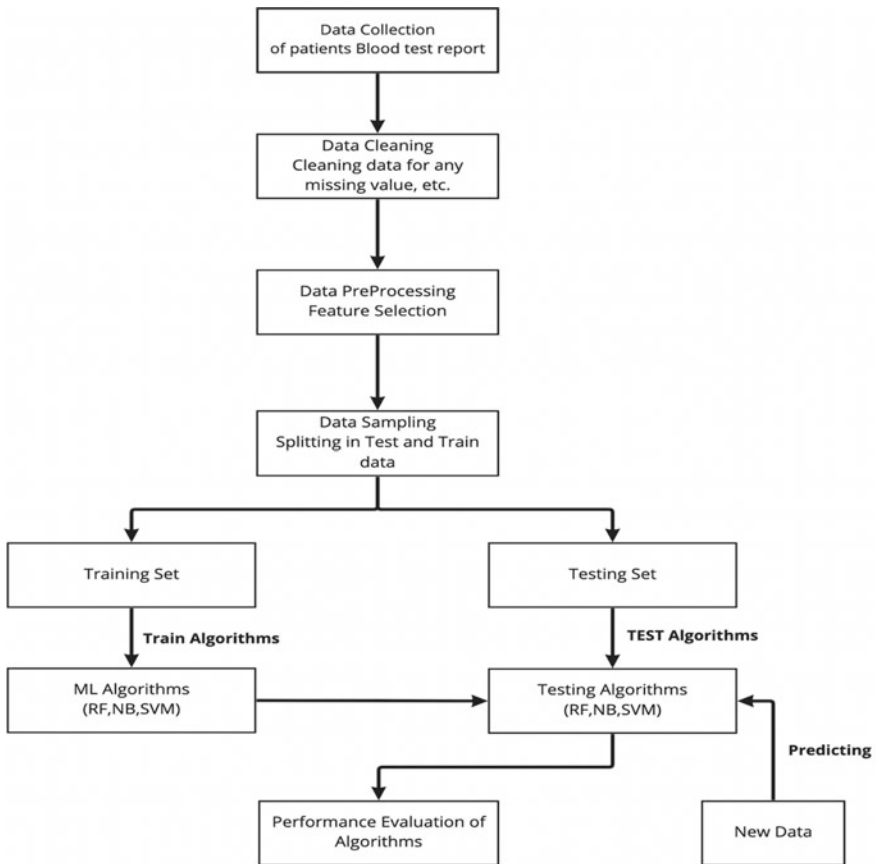


Fig. 4 Flowchart [2, 4]

- Algorithms that we are going to use are Random Forest, SVM, Naïve Bayes etc. [2, 3].
- Below are the detailed study of our algorithms

1. *Naive Bayes Algorithm*—Naive Bayes Algorithm is a supervised machine learning algorithm which is based on the famous bayes theorem and is used mostly to solve classification problems. It is one of the easiest and effective classification algorithms. It basically predicts the output based on the basis of the probability of the object [12].

Now, defining the formula as per our project

$P(A|B)$ is Posterior probability: Probability of hypothesis A on the observed event B.

$P(B|A)$ is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

$P(A)$ is Prior Probability: Probability of hypothesis before observing the evidence.

$P(B)$ is Marginal Probability: Probability of Evidence [12].

$$P(A|B) = P(B|A) P(A)/P(B) \tag{1}$$

As per our problem, We define the formula:

$P(\text{YES}|\text{Anemia})$ is Probability of having Anemia Disease in a person.

$P(\text{Anemia}|\text{YES})$ is the value of patients having parameters outside the normal range having anemia.

$P(\text{Anemia})$ is the value of people having Anemia.

$P(\text{YES})$ is the value of total people having blood parameters out of range.

So, we can rewrite the Naive Bayes algorithm as

$$P(\text{YES}|\text{Anemia}) = P(\text{Anemia}|\text{YES}) * P(\text{Anemia})/P(\text{YES})$$

We will then compare this value with the normal or parameter of people not having Anemia

$$P(\text{NO}|\text{Anemia}) = P(\text{Anemia}|\text{NO}) * P(\text{Anemia})/P(\text{NO})$$

After this calculation, we will in the end compare both these, and the one greater will be the final answer

If $P(\text{NO}|\text{Anemia})$ is greater than $P(\text{YES}|\text{Anemia})$, then the person is not suffering from Anemia, else vice-versa.

2. *Random Forest*—Random forest is a simple to use machine learning algorithm that delivers a good result much of the time, it also does not require us to use hyper parameter tuning. It is also one of the most commonly used algorithms due to its simplicity and versatility which can be used as both regression and classification algorithms [13].

Why use Random Forest?

Random Forest is one of the most popular machine learning algorithms used for both classification and Regression problems. It is used because of its speed, that is it works very fast even for very big datasets. It

also provides a very high accuracy in comparison with the other machine learning algorithms [13].

How does the Random Forest algorithm work?

Random Forest as the name suggests, is an algorithm created by the use of multiple decision trees. In this, Random Forest algorithm creates multiple decision trees, and then as per the input, the decision tree shows the output. In random forest, the algorithm for a classification problem takes all the majority classes predicted by all decision trees and average of all predicted outputs for a Regression Problem [13].

Now, lets see the working of Random Forest Algorithm

Step-1: Firstly, we select random data points from the training data set.

Step-2: Next, we build a decision tree for each of the respective data points.

Step-3: Next, we decide the number of decision trees we want.

Step-4: Repeat Step 1 and 2.

Step-5: Now, for predicting, compile all the outputs of all decision trees and take the majority of all outputs for the final output.

3. *SVM*—Support Vector Machine is one of the best machine learning algorithms when it comes to classification problems. This is exactly what SVM does! It tries to find a line/hyperplane (in multidimensional space) that separates the two classes. It then classifies the new point as to whether it lies on the positive or negative side of the hyperplane, depending on the classes to be predicted [14].

Steps to implement support vector regression in python:

- Import the library
- Read the dataset
- Feature Scaling
- Fitting SVR to the dataset
- Predicting a new result
- Visualizing the results of support vector regression

Support vector regression is the counterpart of a support vector machine for regression problems. In our project we are using different attributes of the dataset and predicting the result using this support vector machine [14].

5. Real Time Implementation of Project

- Here comes the main part where we have to map our project with the real world problems.
- For this purpose we are trying to reach the various resource persons which are pathologists/doctors and provide them with solutions that our model is giving.
- We have decided to provide our service to NGOs or Social work bodies or organizations or medical bodies or rural clinics or hospitals where there is a lack of experienced medical staff.

- Patients can, on our website, just put in their blood test parameters and our machine learning model will predict whether the patient is suffering from anemia or not.

4 Technology Used

Technology and Tools that we are going to use in our project:

We are using one of the most useful and powerful languages i.e. Python. Python also has robust library support for Machine Learning.

1. Google Collab—This is a Jupyter notebook IDE where we can easily run and also see the output of each cell simultaneously. We will use Google Colab as it already has many of the required libraries installed.
2. Pandas—This is one of the most important libraries for data science applications. It is used for cleaning and perfecting our dataset before inserting it in the machine learning model.
3. Scikit—It is a machine learning library containing many models like classification, regression and clustering algorithms. It also has a metrics module which is used for checking the accuracy of the models.
4. Matplotlib—It is a library used for data analysis. It is a library used to create various types of graphs.
5. Seaborn—It is a library used for creating many types of graphs.
6. Flask—It is a library which we will use to create our website where the user enters their CBC parameters.

5 Result and Discussion

- After implementation of all the above steps, we have come up with the accuracy we have achieved using the Random Forest, Naive Bayes and SVM algorithms (Table 1).
- Above is the accuracy we have achieved from our algorithms after training them and then testing them with the test data.
- We have also below attached the True Positive, True Negative, False Positive and False Negative of each algorithm.

Table 1 Algorithm accuracy

Algorithm	Accuracy (%)
Random forest	99.38
Naive Bayes	95.65
SVM	97.52

Algorithms	True positive	True negative	False positive	False negative
Random forest	99	61	1	0
Naive Bayes	95	59	5	2
SVM	97	60	3	1

- Using TP, TN, FP, FN we have found the accuracy using the formula [15]

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad (2)$$

- As we see, our results are up to standards and the accuracy of each algorithm is very good, even exceeding our expectations.

Acknowledgements The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have completed the project successfully. We would like to thank everyone for their guidance.

We sincerely thank our Principal, Dr. B. K. Mishra, Vice Principal, Dr. Kamal Shah and HOD IT, Dr. Sangeeta Vhatkar for always encouraging us to do our best. We are highly indebted to our guide Dr. Sangeeta Vhatkar who supported and constantly supervised us through this project and helped us in not only completing this project but also provided us with a sample amount of knowledge that was really beneficial to us.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all teaching staff of the IT Department who helped us in successfully completing our project work. Also, we would like to extend our sincere thanks to all staff in the laboratory for their timely support.

We would like to express our gratitude towards our parents for their kind cooperation and encouragement which helped us in completion of this project.

References

1. Pasricha SR, Biggs BA, Prashanth NS, Sudarshan H, Moodie R, Black J, Shet A (2011) Factors influencing receipt of iron supplementation by young children and their mothers in rural India: local and national cross-sectional studies. *BMC Public Health* 3(11):617. <https://doi.org/10.1186/1471-2458-11-617>. PMID:21810279;PMCID:PMC3171369
2. Jaiswal M, Siddiqui TJ (2018) Machine learning algorithms for anemia disease prediction: select proceeding of IC3E 2018. <https://www.researchgate.net/publication/329484705>
3. Pavan B, Chandra YH, Yeruva S, Shradhah M, Jain S, Kumar AR, Kondaveeti S (2020) Prediction of anemia disease using classification methods. *EasyChair Preprint* April 13, 2020
4. Yıldız TK, Yurtay N, Öneç B (2021) Classifying anemia types using artificial learning methods. *Eng Sci Technol Int J* 24(1):50–70. ISSN 2215-0986. <https://doi.org/10.1016/j.jestch.2020.12.003>. (<https://www.sciencedirect.com/science/article/pii/S2215098620342646>)
5. <https://www.kaggle.com/code/rahulsarkar221/anemia-predictive-analysis/data>
6. Barpanda SS (2013) Use of image processing techniques to automatically diagnose sickle-cell anemia present in red blood cells smear. Department of Electrical Engineering National Institute of Technology Rourkela-769008 (ODISHA), May-2013. <https://core.ac.uk/reader/5318955>
7. Abdullah M, Al-Asmari S (2016) Anemia types prediction based on data mining classification algorithms. *Communication, management and information technology*. CRC Press, pp 629–636. https://www.researchgate.net/publication/311107778_2016

8. <https://medlineplus.gov/lab-tests/mcv-mean-corpuscular-volume/>
9. <https://www.webmd.com/a-to-z-guides/what-are-mch-levels>
10. <https://www.healthline.com/health/low-mchc>
11. <https://medlineplus.gov/lab-tests/hemoglobin-test/>
12. <https://www.javatpoint.com/machine-learning-naive-bayes-classifier>
13. <https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/>
14. <https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm>
15. <https://www.javatpoint.com/confusion-matrix-in-machine-learning>

Author Index

A

Aayan Khan, 143
Abhinav Shambharkar, 35
Abhishek Gupta, 15, 25
Aditya Dixit, 229
Akash Singh, 219
Akshay Mendon, 191
Aman Jain, 207
Amruta Bodhankar, 157
Anand Khandare, 157, 219
Ankur Kulkarni, 157
Anmol Bajaj, 123
Anuja Somthankar, 1
Anuj Goenka, 143
Ashish Jadhav, 179
Ayush Chandak, 45

D

Dakshal Dhere, 143

E

Eliganti Ramalakshmi, 85

G

Gurleen Pannu, 123

H

Harsh Doshi, 191
Himanshu Soni, 25
Huma Hussain, 85

I

Ishaan Mane, 157

K

Kritika Agarwal, 85
Kshitij Taley, 35
Kunal Bhatt, 207

L

Loukik S. Salvi, 179

M

Mahima Churi, 123
Manish Rana, 207
Megharani Patil, 1, 123, 191

N

Naman Chandak, 45
Nandan Kanvinde, 15
Neelam Sunil Khasgiwala, 111
Niki Modi, 73
Nimavat Dhaval, M., 61
Nipun Agarwal, 157

O

Om Deshpande, 45

P

Pinky Gerela, 15
Prathamesh Bagal, 207

Pratik Fandade, 45
Premanand Ghadekar, 35
Priyadarshan Dhabe, 45

R

Rahul Jha, 229
Raiyani Ashwin, G., 61
Raunak Joshi, 15, 25
Raunak Mishra, 229
Rishabh Chopda, 143
Ronald Laban, 25

S

Samiksha Bedekar, 219
Sangeeta Vhatkar, 229
Sanika Salunke, 35
Sedamkar, R. R., 111
Shailja Jadon, 207

Shivanshu Shrivastava, 1
Shiwani Gupta, 143
Shreyas Mendhekar, 35

T

Tanmayi Nagale, 95

V

Vallabh Niturkar, 35
Vatsal Kadakia, 191
Vedant Pandya, 1
Vidhi Punjabi, 219

Y

Yash Gupta, 191
Yash Oswal, 45