Lecture Notes in Electrical Engineering 1105

N. R. Shetty
N. H. Prasad
H. C. Nagaraj *Editors*

Advances in Communication and Applications

Proceedings of ERCICA 2023, Volume 2

Springer

Lecture Notes in Electrical Engineering

Volume 1105

The book series *Lecture Notes in Electrical Engineering* (LNEE) publishes the latest developments in Electrical Engineering—quickly, informally and in high quality. While original research reported in proceedings and monographs has traditionally formed the core of LNEE, we also encourage authors to submit books devoted to supporting student education and professional training in the various fields and applications areas of electrical engineering. The series cover classical and emerging topics concerning:

- Communication Engineering, Information Theory and Networks
- Electronics Engineering and Microelectronics
- Signal, Image and Speech Processing
- Wireless and Mobile Communication
- Circuits and Systems
- Energy Systems, Power Electronics and Electrical Machines
- Electro-optical Engineering
- Instrumentation Engineering
- Avionics Engineering
- Control Systems
- Internet-of-Things and Cybersecurity
- Biomedical Devices, MEMS and NEMS

For general information about this book series, comments or suggestions, please contact leontina.dicecco@springer.com.

To submit a proposal or request further information, please contact the Publishing Editor in your country:

China

Jasmine Dou, Editor (jasmine.dou@springer.com)

India, Japan, Rest of Asia

Swati Meherishi, Editorial Director (Swati.Meherishi@springer.com)

Southeast Asia, Australia, New Zealand

Ramesh Nath Premnath, Editor (ramesh.premnath@springernature.com)

USA, Canada

Michael Luby, Senior Editor (michael.luby@springer.com)

All other Countries

Leontina Di Cecco, Senior Editor (leontina.dicecco@springer.com)

**** This series is indexed by EI Compendex and Scopus databases. ****

N. R. Shetty · N. H. Prasad · H. C. Nagaraj
Editors

Advances in Communication and Applications

Proceedings of ERCICA 2023, Volume 2

Springer

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

N. H. Prasad
Nitte Meenakshi Institute of Technology
Bengaluru, Karnataka, India

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

ISSN 1876-1100 ISSN 1876-1119 (electronic)
Lecture Notes in Electrical Engineering
ISBN 978-981-99-7632-4 ISBN 978-981-99-7633-1 (eBook)
https://doi.org/10.1007/978-981-99-7633-1

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

The Eight International Conference on "Emerging Research in Computing, Information, Communication and Applications," ERCICA 2023 is an annual event organized at the Nitte Meenakshi Institute of Technology (NMIT), Yelahanka, Bengaluru, India, in association with Laghu Udyog Bharati (LUB—Karnataka) and technology partner National Research Development Corporation, Ministry of Science and Technology, Government of India.

ERCICA aims to provide an interdisciplinary forum for discussion among Researchers, Engineers, and Scientists to promote research and exchange of knowledge in computing, information, communications, and related applications. This conference will provide a platform for networking of Academicians, Engineers, and Scientists and also will encourage the participants to undertake high-end research in the above thrust areas.

ERCICA 2023 received more than 400+ papers from all over the world, viz. from China, UK, Africa, Saudi Arabia, and India. The ERCICA Technical Review Committee has followed all necessary steps to screen more than 400 papers by going through six rounds of quality checks on each paper before selection for Presentation/Publication in Springer, Lecture Notes in Electrical Engineering (LNEE) proceedings, is SCOPUS indexed.

The acceptance ratio is 1:4

Kalaburagi, India N. R. Shetty
Bengaluru, India N. H. Prasad
Bengaluru, India H. C. Nagaraj
February 2023

Contents

About the Editors

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

N. H. Prasad is currently working as Professor and Head of the Department of Master of Computer Applications at Nitte Meenakshi Institute of Technology, Bengaluru. He completed his Ph.D. at Jawaharlal Nehru University, India. He has more than 18 years of experience in different roles in both public and private sector enterprises, including the Ministry of Human Resource and Development, Government of India. He has received the prestigious "Dr. Abdul Kalam Life Time Achievement Award" and also received a "Young Faculty" Award at the 2nd Academic Brilliance Awards.

H. C. Nagaraj obtained his B.E. Degree in Electronics Communication Engineering securing first class with distinction from the University of Mysore in 1981. He obtained his M.E. in Communication Systems from P. S. G. College of Technology, Bharathiar University in 1984 and secured first rank. He received his doctoral degree in the area of Biomedical Signal Processing and Instrumentation from the Indian Institute of Technology (IIT) Madras in 2000. He assumed charge as Principal of Nitte Meenakshi Institute of Technology, Bengaluru, in September 2003, and to date, he is serving as Principal. He has about 38 years of experience in teaching, research, and administration. He has published a number of papers in national and international journals, and conferences and delivered invited talks in the field of biomedical signal processing, image processing, mobile communication, etc. He has guided several students in their Ph.D. programs.

Prediction of Airfoil Efficiency by Artificial Neural Network

Chinmayi Siddamsetty, D. Jyothika, Tabindah Saleem, P. Manvitha, H. V. Srikanth, and S. Vijay Kumar

1 Introduction

It is essential and critical to validate the aerodynamic properties of an airfoil in designing the fluid flow application and developing the optimal characteristics in this regard. The aerodynamic properties of fluid flow around an airfoil are accurately predicted by using the Reynolds-averaged Navier–Stokes (RANS) equations. However, this method has very elevated computational costs and very long progression intervals. Several researchers from the aviation industry have validated machine learning techniques for solving fluid flow problems that are less time-consuming and cost-effective. Although recent advances in computational power and efficiency have greatly reduced these costs, performing numerical simulations remains a time-consuming and computationally intensive task for many practical applications. As a result, it was necessary to shorten the computational time and cost for solving problems related to fluid flow.

Several studies, namely incompressible, steady-state flows, predict flow fields, and aerodynamic force coefficients of airfoils, were used to determine the characteristics and performance of the airfoil.

These diversified classes of neural networks have been consistently used to ascertain the visual imagery that these techniques incorporate deep learning techniques, and training can be done by extracting the unique features of an employed convolution layer. For explicit problems, an ANN model is adopted. When this model is used

C. Siddamsetty (✉) · D. Jyothika · T. Saleem · P. Manvitha · H. V. Srikanth
Department of Aeronautical Engineering, Nitte Meenakshi Institute of Technology, Bangalore, India
e-mail: schinmayi72@gmail.com

S. Vijay Kumar
Department of Mechanical Engineering, Nitte Meenakshi Institute of Technology, Bangalore, India

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_1

for the same purpose, it assists in identifying solutions that can be easily validated (Narendra and Parthasarathy 1990; Hunt et al. 1992).

Ahmed and Kamal (2022) proposed that the ability of a BPNN model to predict the significant aerodynamic features Cl and Cd of airfoils was evaluated. The training was carried out for the BPNN model, validated, and tested using aerodynamic data obtained through numerical simulations of 440 cases. In summary, the BPNN produced promising results for predicting the aerodynamic coefficients of airfoils that were contemplated for distinct conditions. Moin and Khan (2021) cited the best geometric coordinates which were determined by training and comparing the network architecture performance of airfoil. In contrast to CFD analysis, the neural network was found to be proficient in capturing aerodynamic properties with sparse geometric information.

Sekar and Khoo (2019) carried out investigations without directly solving the Navier–Stokes equations, where CNNs were used to estimate the flow field over an airfoil as a role of airfoil geometry, Re, and more. Chen et al. (2020) created a dataset of CAI to determine the CL and CD using flow condition convolution. Bhatnagar and Afshar (2019) used CNNs to validate velocity and pressure fields. There are reports that CNN is used to explain the pressure distribution pattern around the airfoils and to predict the nonuniform study laminar flow in the 2D or 3D domain (Hui et al. 2020; Guo et al. 2016). However, simple ANN architectures have been extensively adapted for addressing the problems associated with designing of an inverse airfoil. Rai and Madavan designed turbomachinery airfoils utilizing pressure distribution (Rai and Madavan 2001) and other design variables (Rai and MAdavan 2000) as input to ANNs. Huang et al. (1994) used ANNs to design and evaluate the Eppler method, which explains the velocity distribution of the airfoil.

The authors claimed that the lift coefficients determined by this method are highly accurate in contrast to the claim, and it is observed that there was a deviation between predicted and actual coefficients. Khurana et al. (2008) used an approach with an ANN on PARSEC (a common method for airfoil parameterization) airfoils as a search agent in order to optimize their shapes. Various reports reported the possibility of using supervised machine learning techniques in the field of aerodynamics (Duru et al. 2021). However, in the previous studies, the ANN model was trained for lesser cases with less range of AoA and needs further investigations to prove its adaptability.

In this study, the test cases were increased along with the range of AoA. The numerical simulation was performed in CFD. This resulted in the model being more accurate in predicting the Cl and Cd for different airfoils. The ANN model was trained in MATLAB using a dataset with different operating conditions such as Re and AoA as input data and the corresponding CL and CD as output data. The primary goal of this research was to develop an ANN architecture capable of predicting the most important aerodynamic characteristics Cd and Cl, which were determined for different airfoils under various aerodynamic conditions.

2 Methodology

2.1 Numerical Simulation

The numerical simulation was performed in the CFD software package Ansys Fluent 16.0. The numerical simulations were performed on different airfoil geometries under various flow conditions to obtain the coefficient of lift and drag which then could be used as a training data set to train and validate the ANN model. The CFD analysis on four different NACA series airfoils, i.e., NACA 0012, NACA 0015, NACA 2412, and NACA 4415 was carried out. NACA 0012 and NACA 0015 are symmetrical airfoils and NACA 2412 and NACA 4415 are asymmetric airfoils that have various applications in the field of aeronautics. The simulation was performed for 20 different AoAs on each airfoil varying from $-10°$ to $10°$. The Re varied from 0.5×10^6 to 5×10^6, i.e., the Re used in the simulation was 0.5×10^6, 1×10^6, 1.5×10^6, 2×10^6, 2.5×10^6, 3×10^6, 3.5×10^6, 4×10^6, 4.5×10^6, and 5×10^6. Each airfoil was simulated with each Re at the AoA varying from $-10°$ to $10°$ with a 1° increment between the cases. The meshing details of selected airfoils are shown in Figs. 1a–d.

Altogether, the CFD analysis of 850 such cases was performed. The K-epsilon turbulence model was used which is a general description of turbulence by means of PDE. The standard k-epsilon model was used for a much more practical approach which helps in minimizing the unknowns and presents a set of equations that could be applied to various turbulent applications (Li et al. 2020). The C-type domain was used with the boundary conditions given as velocity inlet and pressure outlet at the inlet and outlet, respectively. The velocity specification was selected as magnitude and direction where magnitude was changed according to the Re. The velocity components x and y are changed with respect to the AoA as $\cos\alpha$ and $\sin\alpha$.

When an airfoil is subjected to flow, its streamlined shape enables the flow to split between the airfoil's upper and lower surfaces. The pressure in the upper surface of the airfoil decreases as the flow stretches over the curved upper surface, whereas in the flat lower section, the flow speed and pressure are constant. At the leading edge of the airfoil, the pressure distribution is maximum, whereas at the trailing edge, flow separation occurs. The static pressure distribution is predicted using CFD, and the contour plots are shown in Figs. 1e–h. The conditions for numerical simulations for airfoil flow analysis are mentioned in Table 1.

2.2 Feedforward Backpropagation Neural Network

A feedforward backpropagation neural network (ANN) is a form of ANN that has three layers: the input layer, the hidden layer, and the output layer. The substantial number of hidden layers is determined by the nature of the challenge. The number of neurons or nodes in the input layer is determined by the quality of the input characteristics, and the number of nodes or neurons in the layer that produces output

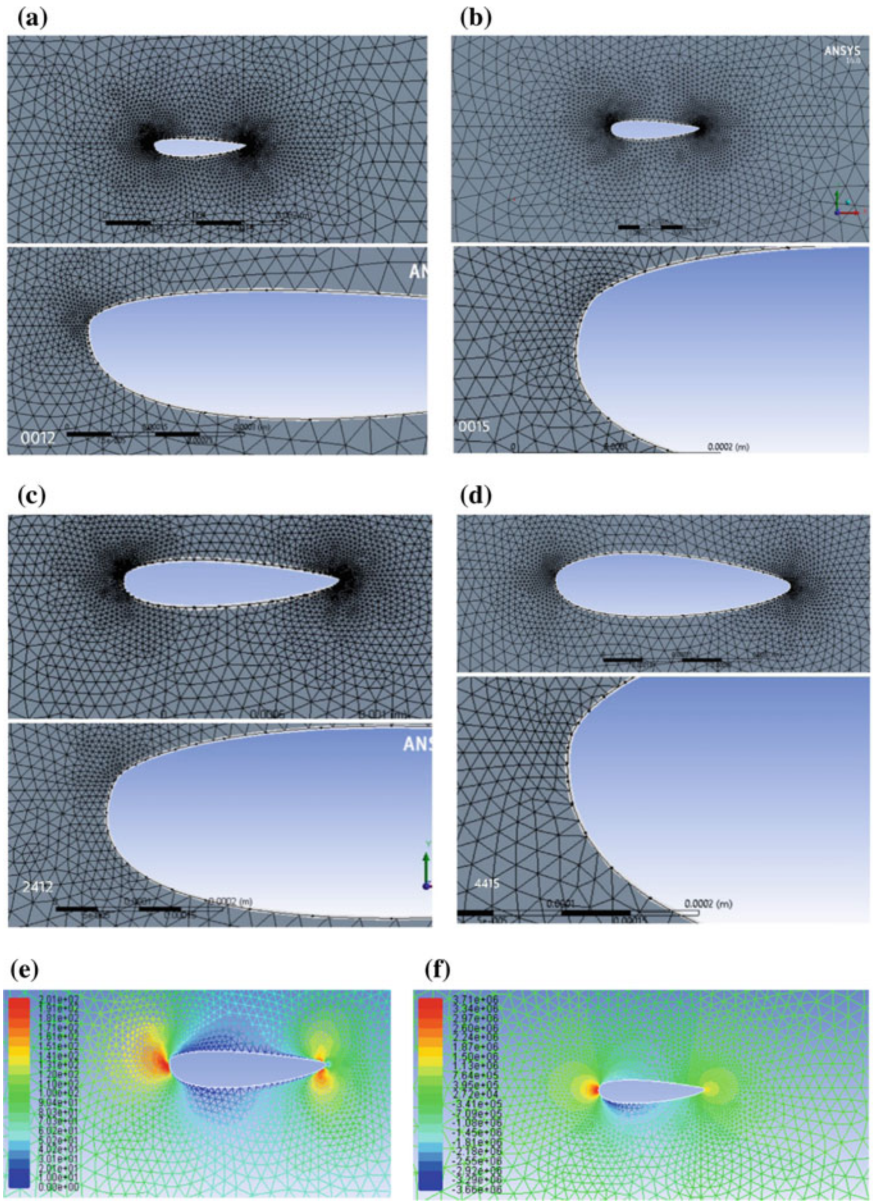


Fig. 1 **a** Meshed airfoil NACA 0012, **b** meshed airfoil NACA 0015, **c** meshed airfoil NACA 2412, **d** meshed airfoil NACA 4415, **e** static pressure plot of NACA 0012, **f** static pressure plot of NACA 0015, **g** static pressure plot of NACA 2412, **h** static pressure plot of NACA 4415

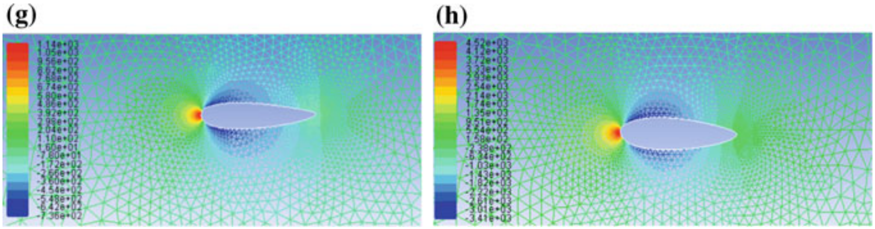


Fig. 1 (continued)

is determined by the number of outputs. Weighted connections transmit all processing items to the next layer, such as neurons or nodes. The weights of the corresponding neurons have been multiplied through the hidden layer's inputs and aggregated by the hidden layer to generate a summed output. The output is then processed using the sigmoid, tan-sigmoid, or threshold activation functions (Yuan et al. 2018). This type of neural network is used because of its speed of functioning, accuracy, and due its ease of implementation. The general architecture of the model used is shown in Fig. 2.

The delta rule is known as the WIDROW-HOFF learning rule using a supervised learning technique. Our model was trained by the adaption learning function called gradient descent (LEARNGDM) which was adopted by the following procedure.

1. By giving all the random values each weight w_{ij} and b_j where $i = 1$ to n and $j = 1$ to m.
2. Input and output datasets are fed into the BPNN model, and the output of each layer is calculated by the following equation:

$$y_{jp}^{[lc1]} = f\left(\sum_{i=1}^{N1} w_{ij}^{[l+1]} y_{ip}^{[l]} + b_j^{[l+1]}\right).$$

3. Error at the output layer is computed by the below formula:

$$\text{err}_{jp}^{[L]} = f\prime\left(y_{jp}^{[L]}\right)\left(d_p - y_{jp}^{[L]}\right),$$

in the i_{th} hidden layer ($i = L - 1, L - 2, \ldots 1$).

4. The following equations are used to calculate the bias between the input and output layers and changes in weights

$$\begin{aligned} b_{ij}^{[l]}(n+1) &= b_i^{[l]}(n) + \text{NG.err}_{jp}^{[l]}, \\ w_{ij}^{[l]}(n+1) &= w_i^{[l]}(n) + \text{NG.err}_{jp}^{[l]}.y_{jp}^{[i-1]}. \end{aligned}$$

Error terms are back propagated into the neurons of the previous layer while calculating the changes in weights and biases.

Table 1 Numerical simulation for various flow conditions

Airfoil NACA	Reynolds number	Angle of attack																				
0015	500,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	1,000,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	1,500,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	2,000,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	2,500,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	3,000,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	3,500,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	4,000,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	4,500,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	5,000,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
0010	500,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	1,000,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	1,500,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	2,000,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	2,500,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	3,000,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	3,500,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	4,000,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	4,500,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	5,000,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
2412	500,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10

(continued)

Table 1 (continued)

Airfoil NACA	Reynolds number	Angle of attack																				
	1,000,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	1,500,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	2,000,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	2,500,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	3,000,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	3,500,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	4,000,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	4,500,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	5,000,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
4415	500,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	1,000,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	1,500,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	2,000,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	2,500,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	3,000,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	3,500,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	4,000,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	4,500,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10
	5,000,000	– 10	– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10

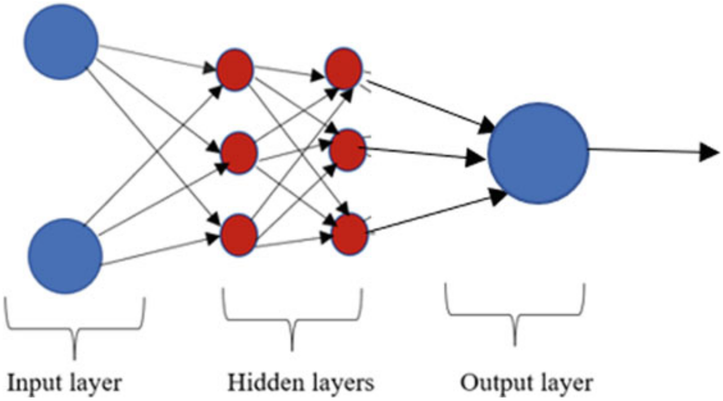


Fig. 2 General block diagram of BPNN

5. Steps 2–5 are repeated till the errors are lesser than the minimum specified error.
6. The present study involves the usage of the sigmoid activation function and is given by the following equation:

$$F(x) = 1/\left(1 + \mathrm{e}^{-x}\right).$$

2.3 *Prediction of Aerodynamic Coefficients*

The block diagram (Fig. 3) depicts the methodology used in this neural network which is used as a regression analysis tool to determine airfoil aerodynamic coefficients such as coefficient of lift and drag (Li et al. 2019). The dataset used in our model consists of 850 cases obtained through CFD analysis in the ANSYS fluent software on four different NACA airfoils: NACA 0015, NACA 0012, NACA 2412, and NACA 4415. Each of these NACA series was analyzed with ten different Re ranging from $5 * 10^5$ to $5 * 10^6$ for various AoA ranges.

The dataset contained two input parameters, velocity and AoA, as well as output parameters, Cl and Cd 70% of the data in the dataset was used for training the network, while 30% was used for testing and validation (Michos et al. 1983). Later, a neural network was trained using a backpropagation algorithm such as the Levenberg–Marquardt algorithm, and training continued until it reached a stopping criterion, which was based on the validation error, i.e., the error should reach a minimum value (Yildiz et al. 2415).

The architecture of the BPNN used consists of two inputs with ten neurons in the hidden layer and two outputs as shown in Fig. 4.

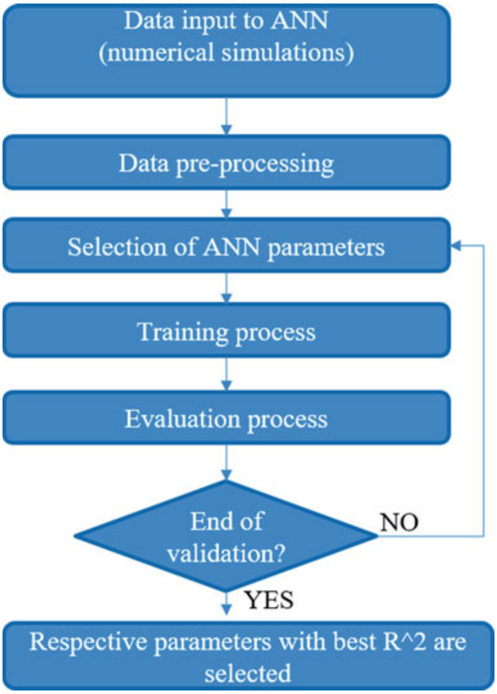


Fig. 3 Flowchart of the steps followed

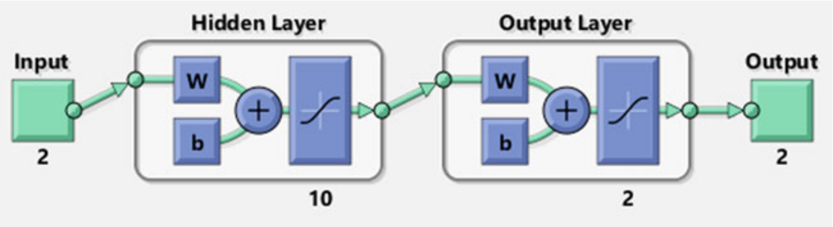


Fig. 4 Architecture of BPNN model used

2.4 *Performance Evaluation*

The following parameters are used to evaluate the performance of the BPNN model.

1. RMSE: Lower the value of RMSE, higher the accuracy of the regression of the model, and it is calculated by the following formula:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=0}^{n}|Pi - Mi|^2}.$$

2. Pearson correlation coefficient (R): R-value varies from − 1 to 1; the accuracy of the regression model is said to be high if it has R-value near to 1. R I calculated by the following formula:

$$R = \sqrt{1 - \frac{\Sigma(Mi - Pi)^2}{\Sigma\left(Mi - \overline{Mi^{-}}\right)^2}}.$$

n number of data points in the testing subset.
P_i predicted values for the ith aerodynamic coefficient.
M_i mean of all measured values for the aerodynamic coefficients.

3 Results and Discussions

The multiple numbers of neurons or nodes that are in the hidden layer are used to train the ANN model to get the best RMSE value. The RMSE values associated with number of neurons in the hidden layer are summarized in Table 2. Ten number of neurons in the hidden layer gave the best RMSE value, i.e., 0.0076318 at 27 epochs. It was discovered that the correlation (R) value was close to one for all training, validation, and testing cases indicating high accuracy of the predicted value. From the table, it may be inferred that changes in the number of neurons affect the RMSE value in zig-zag fashion indicating that RMSE value is sensitive to several neurons.

Figure 5 describes the relationship between the RMSE and neurons in the hidden layer. A minimum value of 0.0076318of RMSE with accurate prediction was achieved, but there was no continuous trend. As the number of neurons increases, RMSE value for neuron 10 has increased, and then, it decreased for neuron 14;

Table 2 Correlation table for different neurons

No. of neurons	Epochs	RMSE * 10^{-3}	Corr-coeff. for training	Corr-coeff. for validation	Corr-coeff. for testing	Corr-coeff. overall
10	27	7.6318	0.97557	0.98447	0.98303	0.97812
14	24	6.5816	0.98402	0.98656	0.93697	0.97815
18	12	9.553	0.97581	0.97877	0.98238	0.97716
22	9	5.9927	0.97708	0.98695	0.97691	0.97839
26	6	9.5672	0.9747	0.98147	0.98265	0.97691
30	13	9.3398	0.98386	0.98061	0.94836	0.97769
34	13	10.746	0.98359	0.97696	0.95456	0.97792
36	5	11.265	0.97838	0.97377	0.99939	0.97757
40	7	8.5035	0.9727	0.98105	0.97739	0.97503
44	8	8.4694	0.9768	0.98165	0.9791	0.97809

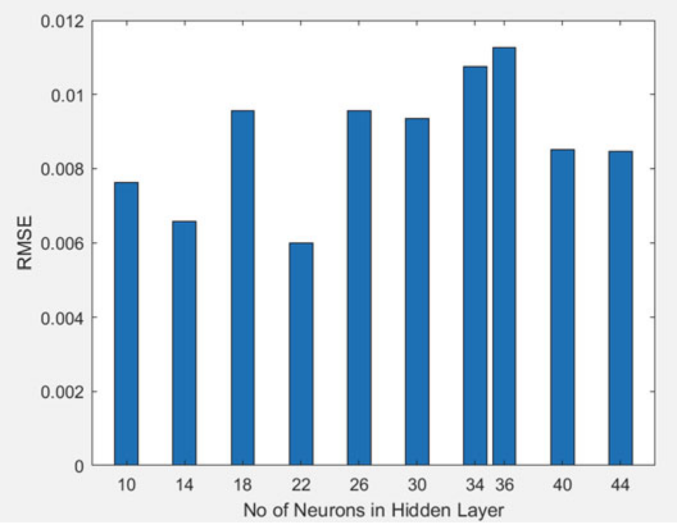


Fig. 5 Neurons in hidden layer versus RMSE

later for neurons 18 and 22, RMSE value has increased and decreased respectively. RMSE value for the networks with neurons 26–36 was seen to increase, whereas RMSE value has again decreased with an increase in neurons to 44.

Figure 6 depicts the relationship between the RMSE value and the network used for the training, validation, testing, and prediction. The graph shows that it was proportionately high at the beginning of training and moderately decreased as training progressed. The RMSE value remained constant after 27 epochs at 0.0076318, and it did not decrease further. This performance graph only shows the best-case scenario. Compared to testing and validation data, RMSE value in the present study is large which may be due to the limited number of training data points in 850 instances.

From Fig. 7, we can conclude that the regression value for training, validation, and testing was near one which indicates high accuracy in the predicted values. BPNN model was trained and validated using the data which was obtained from the historical data. As each of the data points falls on the regressions line, the graphs demonstrate an excellent correlation between the target and projected values. In this work, the trained BPNN model indicates aerodynamic parameters with excellent accuracy and a relatively small number of inaccurate predictions when compared to other current models. Because machine learning techniques require a high number of data sets, model performance is directly related to data size. Only 850 simulated datasets were used to train the BPNN model in this investigation. Because 850 is such a small figure, it could indicate a constraint of the model's performance. To improve the model's dependability, more data from CFD simulations of various airfoils under various conditions must be collected. This will allow us to use this model to estimate aerodynamic coefficients of lift and drag for undiscovered airfoils under varied conditions.

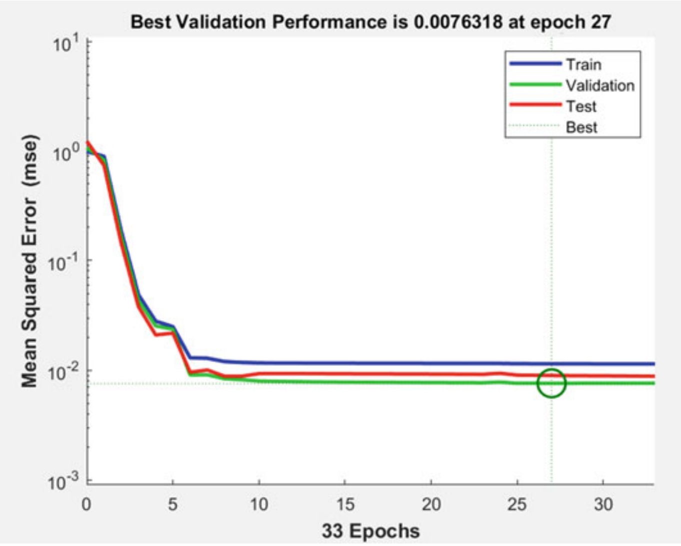


Fig. 6 Performance plot for the best RMSE case

4 Conclusions

Validation of the BPNN model was done on basis of its efficiency to predict aerodynamic parameters. It is evident in Fig. 7 that there is a correlation between predicted values and the dataset as R is near to one. This research has contributed to the development of a model that can be used to determine the aerodynamic coefficients (CL and CD) of any airfoil and evaluate its performance. About 850 instances in the dataset obtained from numerical simulations are used to train, test, and validate the developed ANN model. Regression plots showed a nearly perfect fit between the actual and predicted values. At epoch 27, with ten neurons in the hidden layer, the RMSE for the best validation performance obtained was 0.0076318. The model supports any Mach number between 0 and 0.7 and any Re between 0.5×10^6 and 5×10^6. To summarize, the developed ANN model produced promising results in efficiently predicting the aerodynamic coefficients of various airfoils.

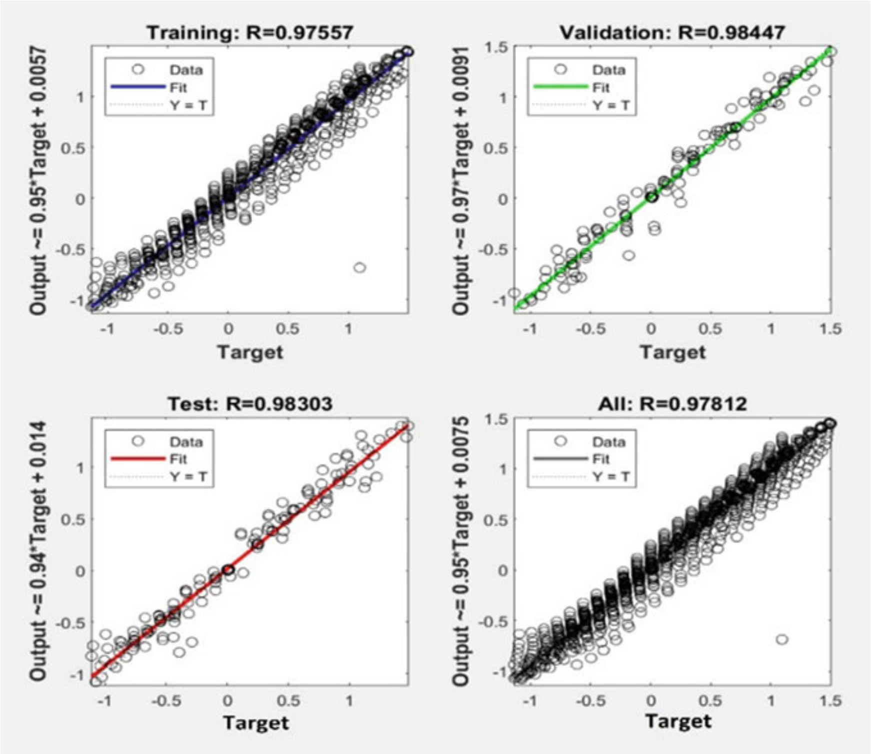


Fig. 7 Regression plots for the best RMSE value

Nomenclature

ANN	Artificial neural network
Re	Reynolds number
M	Mach number
AoA	Angle of attack
Cl	Coefficient of lift
Cd	Coefficient of drag
Cm	Coefficient of moment
BPNN	Backpropagation neural network
RMSE	Root Mean Square Error
PDE	Partial Differential Equations

References

Ahmed S, Kamal K (2022) Aerodynamic analyses of airfoils using machine learning as an alternative to RANS simulation. National University of Sciences and Technology, Islamabad 44000, Pakistan. https://doi.org/10.3390/app12105194

Bhatnagar S, Afshar Y (2019) Prediction of aerodynamic flow fields using convolutional neural networks. Comput Mech 64:525–545. https://doi.org/10.1007/s00466-019-01740-0

Chen H, He L, Qian W, Wang S (2020) Multiple aerodynamic coefficient prediction of airfoils using a convolution neural network. Computational Aerodynamics Research Institute, China Aerodynamics Research and Development Center, Mianyang 621000, China. https://doi.org/10.3390/sym12040544

Duru C, Alemdar H, Baran U (2021) CNNFOIL: convolutional encoder decoder modeling for pressure fields around airfoils. Neural Comput Appl. https://doi.org/10.1007/s00521-020-05461-x

Guo X, Li W, Iorio F (2016) Convolutional neural networks for steady flow approximation. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. https://doi.org/10.1145/2939672.2939738

Huang S, Miller L, Steck J (1994) An exploratory application of neural networks to airfoil design. In: 32nd aerospace sciences meeting and exhibit, p 501

Hui X, Bai J, Wang H, Zhang Y (2020) Fast pressure distribution prediction of airfoils using deep learning. School of Aeronautics, Northwestern Polytechnical University, 127 Youyixi Road, Xi'an, 710072, China.https://doi.org/10.1016/j.ast.2020.105949

Hunt K, Sbarbaro D, Zbikowski R, Gawthrop P (1992) Neural networks for control system

Khurana M, Winarto H, Sinha A (2008) Application of swarm approach and artificial neural networks for airfoil shape optimization. In: 12th AIAA/ISSMO multidisciplinary analysis and optimization conference

Li K, Kou J, Zhang W (2019) Deep neural network for unsteady aerodynamic and aeroelastic modeling across multiple Mach numbers. Nonlinear Dyn 96:2157–2177. https://doi.org/10.1007/s11071-019-04915-9

Li J, Zhang M, Martins JRRA, Shu C (2020) Efficient aerodynamic shape optimization with deep-learning-based geometric filtering. AIAA J 58(10):4243–4259. https://doi.org/10.2514/1.J059254

Michos A, Bergeles G, Athanassiadis N (1983) Aerodynamic characteristics of NACA 0012 airfoil in relation to wind generators. Wind Eng 7:247–262. https://www.jstor.org/stable/43749000

Moin H, Khan HZI (2021) Airfoils aerodynamic coefficients prediction using artificial neural network. Department of Aeronautics & Astronautics, Institute of Space Technology, Islamabad, 44000, Pakistan. https://doi.org/10.48550/arXiv.2109.12149

Narendra KS, Parthasarathy KK (1990) Identification and control of dynamical systems using neural networks. IEEE Trans Neural Netw

Rai MM, Madavan NK (2000) Aerodynamic design using neural networks. AIAA J 38(1):173–182

Rai MM, Madavan NK (2001) Application of artificial neural networks to the design of turbomachinery airfoils. J Propul Power 17(1):176–183

Sekar V, Khoo BC (2019) Fast flow field prediction over airfoils using deep learning approach. Phys Fluids 31:057103. https://doi.org/10.1063/1.5094943

Sun G, Sun Y, Wang S (2015) Artificial neural network based inverse design: airfoils and wings. Aerosp Sci Technol 42:415–428

Yildiz F, Turkmen AC, Celik C, Sarac HI (2015) Pitch angle analysis of NACA 2415 airfoil. In: Proceedings of the world congress on engineering, London, UK, 1–3 July 2015, p 5

Yuan Z, Wang Y, Qiu Y, Bai J, Chen G (2018) Aerodynamic coefficient prediction of airfoils with convolutional neural network. In: Zhang X (ed) Proceedings of the 2018 Asia-Pacific international symposium on aerospace technology (APISAT 2018), vol 459, Chengdu, China, 16–18 Oct 2018. Springer, Singapore, pp 34–46. https://doi.org/10.1007/978-981-13-3305-7_3

Predicting Holding Days for Equity Gain Using Wavelets

Mala Sundaram, Gitapremnath Raja, and Vimal Pant

1 Introduction

The stock price movements and prediction of future stock values are widely studied in various fields including finance, machine learning, and the trading market. This research attempts to establish that stock can be bought and sold by finding an improved equity gain model at profitable positions. Stock price prediction and indices' movements have been previously investigated by several traders, professionals, and researchers. Many prediction methods for the stock price have been proposed in literature based on various factors such as news and sentiment analysis. Since the stock price movements are volatile and non-stationary, i.e., non-stationary stock prices do not remain constant over time. Therefore, stock market prediction becomes a challenging and difficult task. The stock investor's interest is the motivation for this research work. This work signifies stock price movements using national stock exchange data in the Indian capital market. Further, this work finds the trading interval days using wavelet analysis, which can lead to equity gain in the future.

1.1 Mathematical Preliminaries on Wavelets

The proposed work utilizes wavelets for the interval-based stock price and indices' movements. Thus, the mathematical preliminaries on wavelets with their related work are presented. This work examines the theoretical foundations of wavelet analysis and various stock market researches based on wavelets.

M. Sundaram (✉)
Nehru Memorial College, Tiruchirappalli, India
e-mail: ursmala@gmail.com

G. Raja · V. Pant
NIFTEM, Sonipat, India

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_2

1.2 Wavelets

"A wavelet transform is a tool that cuts up data, functions/operators into different frequencies, and then studies each frequency component with a resolution matched to its scale" (Daubechies 1990). Wavelet refers to a small oscillation with zero mean. Wavelet has frequency- and time-localized functions to analyze the signals identifying the abrupt changes accurately. Wavelets exist for a defined duration and come in different sizes and shapes. Wavelength analysis is strengthened by the availability of a wide range of wavelets. The two main concepts in wavelets are: scaling and shifting. The mathematical representation of equivalent frequency is defined in Eq. 1.

$$F_{\mathrm{eq}} = \frac{C_{\mathrm{f}}}{S\delta t}, \tag{1}$$

where

C_{f} Center frequency.
S Scale factor.
δt Sampling interval.

A significant-scale factor results in a stretched wavelet corresponding to low frequency. In contrast, a shorter scale factor results in shrinking/compressed wavelets correspond to high frequency (use narrow window for better frequency resolution). A stretched wavelet helps in capturing slowly varying signals, whereas a compressed wavelet captures abrupt changes. Shifting of a wavelet means delaying (or) advancing the wavelet's onset along the length of the signal. The given signal features are identified by shifting and scaling the wavelets used for the specified application. The scaling and shifting of wavelets are represented in Fig. 1.

1.3 Discrete Wavelet Transform

Continuous and discrete wavelet transform differs by how the wavelet is shifted and scaled. DWT is easy to implement when compared with CWT. The critical application of discrete wavelets is denoising and compression of signals. The idea in denoising is to keep the most significant coefficients. The DWT requires the signal's length to be dyadic (power of 2) and highly dependent on the origin of the signal analyzed. Any signal ($S(t)$) can be decomposed by a sequence of projections onto a wavelet basis, shown in Fig. 2.

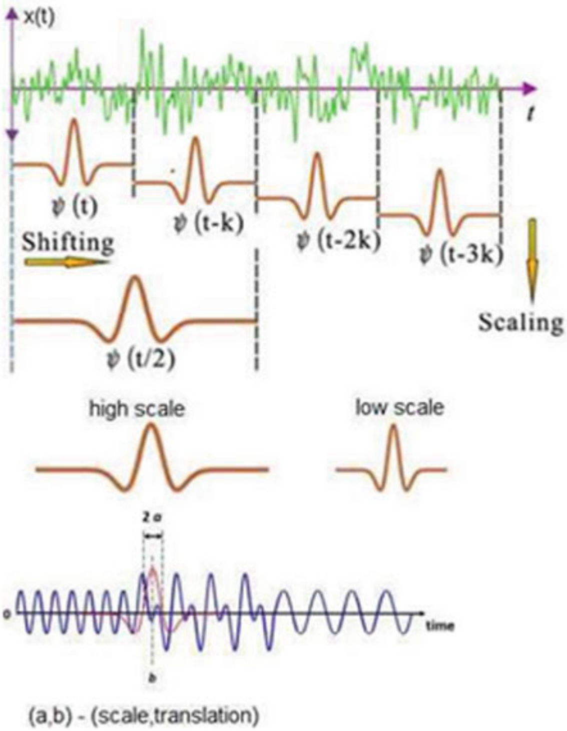


Fig. 1 Wavelet at various scales and translations

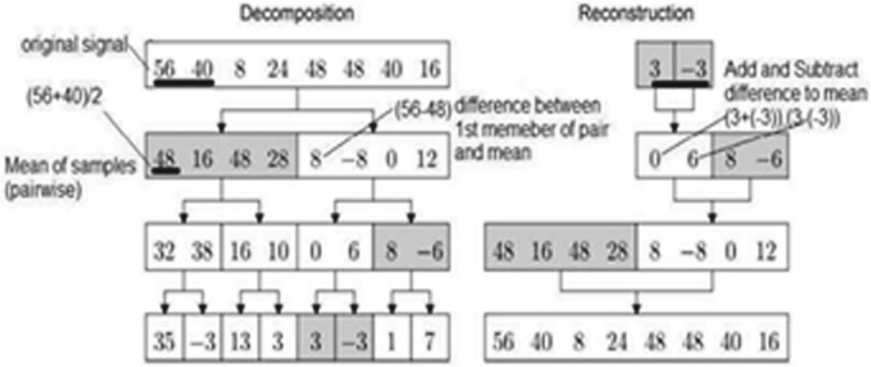


Fig. 2 Wavelet decomposition and reconstruction sample

1.4 Haar Wavelet

Haar Wavelet is the oldest and simplest of wavelets and captures fluctuations between adjacent observations (Huang and Wu 2008; Lahmiri 2013). It resembles a square form. The wavelet function, the scaling function of the Haar Wavelet, is given in Eq. 2.

$$\phi(t) \underset{\leftarrow}{\longrightarrow} \underset{\leftarrow}{\longrightarrow} \underset{\leftarrow}{\longrightarrow} \underset{\leftarrow}{\longrightarrow} = \left\{ \begin{array}{ll} 1 & \text{for} \quad 0 < t < \frac{1}{2} \\ -1 & \text{for} \quad \frac{1}{2} < t < 1 \\ 0 & \text{otherwise} \end{array} \right\},$$

$$\varphi(t) = \left\{ \begin{array}{l} 1 \text{ for} \quad 0 \le t1 \\ 0 \text{ otherwise} \end{array} \right\}. \tag{2}$$

2 Literature Review

This part of the work has focused on studies that have examined the wavelets, wavelets applications, data mining techniques, machine learning, and deep learning implementation in various stock market analyses. Multiple studies show the importance of wavelets and their applications in varieties of fields. This work mainly focuses on wavelets and their applications in stock price and indices' movements.

Wavelets were proposed for various applications (Graps 1995; Gencay et al. 2002; Lee and Yamamoto 1994; Fama 1991). The wavelets to the analysis of financial data and economical data (Ramsey 1999) and wavelets for finding trading interval days are also presented (Mala and Saravanan 2018). The wavelet transform (Daubechies 1990) successfully applied to both time-series financial (Li et al. 2006; Huang and Wu 2008; Lai and Huang 2007; Hsieh et al. 2011; Wang et al. 2019, 2011; Huang 2011; Wang et al. 2011; Kao et al. 2013; Lahmiri et al. 2013; Shah and Debnath 2017) and engineering problems (De and Sil 2012; Rikli 2012) because of its simultaneous analysis in frequency and time domains. Wavelet decomposes given stock price into high- and low-frequency components (Crowley 2005). The wavelets can capture stock price discontinuities in high frequency. In low frequency, the wavelet characterizes the stock data's coarse structure to identify the long-term trends (Lahmiri 2014). Wavelet analysis can extract the hidden temporal features of the stock price. Various factors in the stock market cause stock market instability. The stock market is a complex system that takes much information from news to investors, essential information to socio-political events, and its behavior produces an output (Novak and Veluscek 2015). "Though most of the wavelet applications are in time-series analysis, only very few focuses on stock market data analysis. Deployed various wavelet methods were deployed in stock series with multi-resolution analysis, denoising stock price series, characterizing abrupt changes in the stock prices, and detecting stock series's self-similarity" (Lai and Huang 2007). The discrete wavelet transform is applied to analyze the stock index's prediction on National Stock Exchange Fifty Index (Jothimani et al. 2015). The wavelet analysis improves the forecasting performance and provides promising indicators in stock market analysis. Stock market prediction using hybrid of wavelet transform and Artificial Neural Network has also implemented (Mahadevan et al. 2016). "Theoretical foundations for fundamental research, the Discounted Dividend Model (DDM), and finally the Discounted Cash Flow Model (DCFM), all these models have strengths, despite the lack of accuracy, because it is required financial efficiency market" (Wafia et al. 2015). A new approach for stock returns prediction is presented by applying data mining techniques with possible features based on the filter-based clustering (Barak 2015). This work used a feature selection algorithm and data mining techniques for stock risk and return prediction. In data mining techniques classification, various classifiers for stock price direction prediction and classification were evaluated (Ballings 2015). Data mining methods for analyzing financial reports with quantitative and qualitative data by combining data and text mining techniques were proposed by Kloptchenko (2004). Using context capturing features based on economic news by automated news reading, the stock price was given by Hagenau (2013). The stock price prediction using R, a data mining tool in stock market analysis, is proposed (Angadi and Kulkarni 2015; Khanna 2014). Compared with various models to use, a unidirectional perspective of the quality of volatility of returns forecasts provided by GARCH, without any cost component (Matei 2009). For stock price prediction, various methods have been implemented. Most of the researches focus on predicting the stock value of future trading days. For financial time-series forecasting, the combined prediction model, with principal component analysis (PCA) based on Artificial Neural Networks (ANNs), has also been presented (Hu et al. 2021). Various research works analyzed the stock/forex market prediction through different deep learning methods and identified deep learning to improvise stock value prediction with more accuracy. Notable researches used machine learning techniques. For large and small capitalizations and in different markets, the support vector (SV) predicts stock prices, employing prices with daily and up-to-the-minute frequencies (Miranda 2018; Patel et al. 2015). To find stock price prediction efficiently, optimization methods were implemented (Stankovic et al. 2015). An integrated system used an artificial bee colony (ABC) algorithm for forecasting stock markets using wavelet transforms and recurrent neural networks (Hsieh et al. 2011). Stock trends prediction by wavelet transform and multistage fuzzy inference optimized by the genetic algorithm implemented by Kishikawa and Tokinaga (2000).

The Brazilian stock market model called the increasing–decreasing linear neuron (IDLN) with time-phase adjustment, gradient-based training algorithm, for high-frequency stock market forecasting presented to solve financial time-series problem (Araújo 2015). Efficient Market Hypothesis (EMH) in the Indian stock market in the post-liberalization era (1991–2013) through a series of tests was performed. The research outcomes suggest that despite substantial progress in the Indian stock market, EMH remains weak. Therefore, there remains a distinct possibility by investors in Indian stock markets to generate super-normal returns (Gupta and Gedam 2014). In the Indian Pharmaceutical sector, the technical analysis of selected stocks was performed. The result interprets whether to buy or sell them (Boopalan 2014; Sudheer 2015; Umaprabha and Malavika 2015). The Colombo Stock Exchange (CSE) is examined by its stock price behavior and leads to the result that CSE prices do not display any month-specific behavior (Chkraborty 2006). For Tehran Stock Exchange, a compound model of ARIMA and neural network was proposed to predict the finance market trend (Mousazadeh Abbasi et al. 2015). Few types of research demonstrated can improve the African equity price formation process if African stock markets integrate their operations (Simons and Laryea 2006; Ntim et al. 2011).

Apart from predicting stock price movements and its improvement in the stock index's accuracy, another area that comforts the investors is the optimal timing for buying and selling the stock security toward equity gain. This paper is illustrated to guide the investors to make buying and selling decisions based on trading interval days. This paper presents the basic ideas of interval findings in the stock market using discrete wavelet analysis, and a confusion matrix analyzes the performance of stock value. This work deployed a decision tree classifier for predicting stock price movement and an accuracy measure for validating the stock price movement.

3 Methodology

Data Collection

The data implemented in this research are collected from the portal Yahoo Finance which has one of the richest collections of publicly available stock indices and price data on the internet. By using the fix_yahoo_finance Package in Python, this work downloaded the required data directly from Yahoo Finance. From January 1, 2010, to December 31, 2019, there were a total of 2460 trading days and this work has considered the data from August 31, 2016, to December 31, 2019 (~ 818 trading days), for testing purpose.

Interval-Based Framework

To access and gain stock market knowledge and guide investors in terms of when to buy or sell or hold the shares, interval analysis is needed. This work utilized a simple wavelet-based model for stock interval calculation, which in future related to interval-based trading days. The issues related to the data preprocessing technique, architecture design, wavelet selection, and performance measures are considered carefully. Using Google Colab, the methods were all implemented online. The framework for stock market analysis using trading interval days' calculation is depicted in Fig. 3.

Stock Price Collection and Preprocessing

This work is analyzed using various stock sectors, namely banking sector, pharmaceutical sector, automobile sector, information technology sector, and food sector,

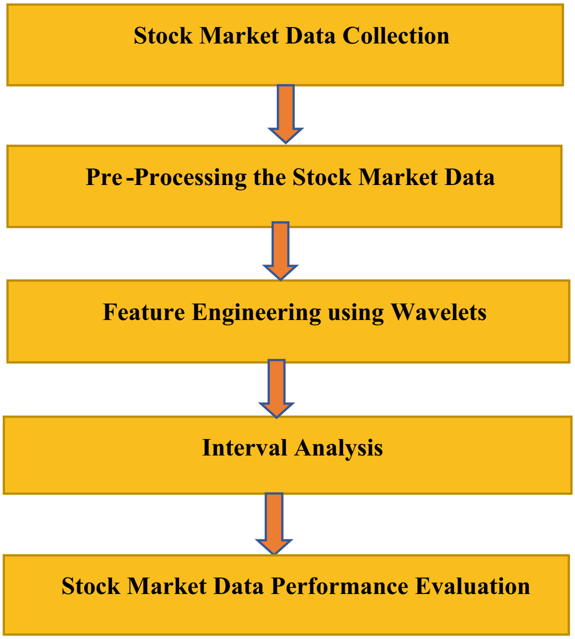

Stock Market Data Collection
Pre-Processing the Stock Market Data
Feature Engineering using Wavelets
Interval Analysis
Stock Market Data Performance Evaluation

Fig. 3 Framework for stock market analysis based on trading interval days

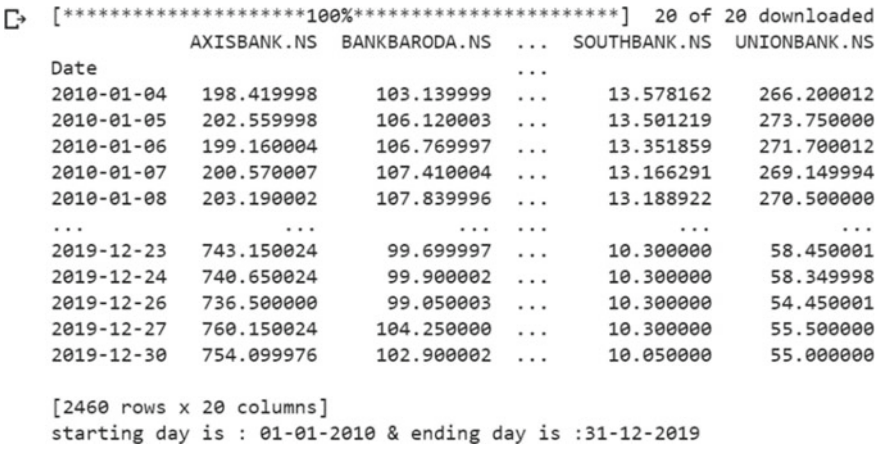
and for all, the same periods of stock prices have been taken for analysis. This work used 50 stocks from five different sectors from January 1, 2010, to December 31, 2019. In total 2460 trading days were taken for analysis. This work excluded those stocks containing any missing values between the periods of study. The missing values, also known as null values, were removed to avoid inconsistencies and handle the data effectively and efficiently. It is verified if the stock contains price values during the trading days; otherwise, it has been omitted. This research work included only the closing price of the stock for each trading day for further analysis.

Banking Sector

Banking sector—January 1, 2010, to December 31, 2019 (2460 trading days)—was taken for analysis. From August 31, 2016 to December 31, 2019 is considered for testing (~ 818 days). The sample dataset for the banking sector is depicted in Fig. 4. It is an online process that directly downloads the data from the internet using the Python package fix_yahoo_finance.

Various stock market data for the banking sector are listed in Table 1, and the ten best-performing bank stocks are chosen for analysis.

The closing price for different banking sector stocks is shown in Fig. 5. This figure indicates that price movements are present in all the stocks for the period of study (from January 2010 to December 2019). The closing price for different pharmaceutical sector stocks is shown in Fig. 6. This figure indicates that price movements are present in all the stocks for the period of study (from January 2010 to December 2019).



```
[*********************100%***********************]  20 of 20 downloaded
              AXISBANK.NS  BANKBARODA.NS  ...  SOUTHBANK.NS  UNIONBANK.NS
Date                                      ...
2010-01-04     198.419998     103.139999  ...     13.578162    266.200012
2010-01-05     202.559998     106.120003  ...     13.501219    273.750000
2010-01-06     199.160004     106.769997  ...     13.351859    271.700012
2010-01-07     200.570007     107.410004  ...     13.166291    269.149994
2010-01-08     203.190002     107.839996  ...     13.188922    270.500000
...                   ...            ...  ...           ...           ...
2019-12-23     743.150024      99.699997  ...     10.300000     58.450001
2019-12-24     740.650024      99.900002  ...     10.300000     58.349998
2019-12-26     736.500000      99.050003  ...     10.300000     54.450001
2019-12-27     760.150024     104.250000  ...     10.300000     55.500000
2019-12-30     754.099976     102.900002  ...     10.050000     55.000000

[2460 rows x 20 columns]
starting day is : 01-01-2010 & ending day is :31-12-2019
```

Fig. 4 Stock price collection for bank sector from Yahoo Finance

Table 1 List of few bank stocks used in this research work

S. No.	Stock symbols
1	AXISBANK.NS
2	BANKBARODA.NS
3	BANKINDIA.NS
4	CANBK.NS
5	CENTRALBK.NS
6	CUB.NS
7	FEDERALBNK.NS
8	HDFC.NS
9	ICICIBANK.NS

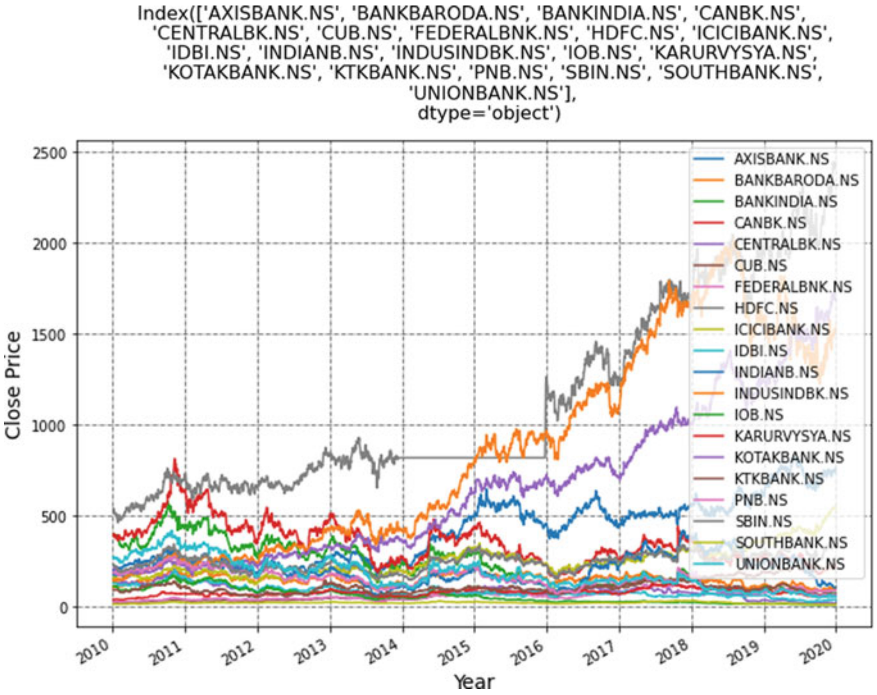
The closing price for different automobile sector stocks is shown in Fig. 7. This figure indicates that price movements are present in all the stocks for the period of study (from January 2010 to December 2019).

The closing price for different food sector stocks is shown in Fig. 8. This figure indicates that price movements are present in all the stocks for the period of study (from January 2010 to December 2019).

The closing price for different information technology sector stocks is shown in Fig. 9. This figure indicates that price movements are present in all the stocks for the period of study (from January 2010 to December 2019).

Wavelet Implementation (Interval Finding)

The stock closing price is the significant input in this processing. First, the interval days are calculated using Haar Wavelet. The numerous features like positive peaks using a threshold in the wavelet features are then analyzed. Significant features and



```
Index(['AXISBANK.NS', 'BANKBARODA.NS', 'BANKINDIA.NS', 'CANBK.NS',
       'CENTRALBK.NS', 'CUB.NS', 'FEDERALBNK.NS', 'HDFC.NS', 'ICICIBANK.NS',
       'IDBI.NS', 'INDIANB.NS', 'INDUSINDBK.NS', 'IOB.NS', 'KARURVYSYA.NS',
       'KOTAKBANK.NS', 'KTKBANK.NS', 'PNB.NS', 'SBIN.NS', 'SOUTHBANK.NS',
       'UNIONBANK.NS'],
      dtype='object')
```

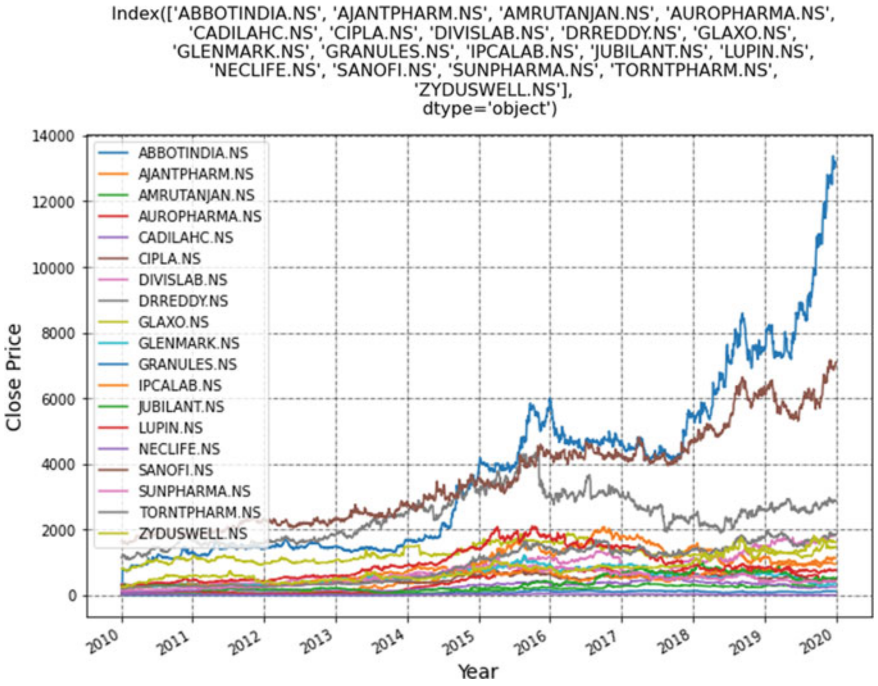
Fig. 5 Various bank sector stock price movements

intervals provide positive peaks in the wavelet, the trading interval days to hold a stock for equity gain. The calculated interval trading days are given as input to the classifier, where the interval for positive stock market prediction is calculated. The results are then evaluated by using a confusion matrix.

The closing price for the stock SBI and wavelet approximation with reconstructed stock price normalized values are shown in Fig. 10. From the wavelet approximation coefficients, the positive and negative peaks are identified first. The positive peaks indicate the equity gain of the stock as in Fig. 11. The difference between the positive peaks is noted for each stock. The average interval days are calculated from the peaks for finding the holding period of the particular stock. After finding the average interval days for equity gains, the stock is analyzed with the test data to see whether the particular interval predicts the gain or not. If the expected and calculated values are positive, then it is noted as true and positive prediction. With this, the accuracy of the stock is calculated and tabulated for each sector stock.

Algorithm for Interval Finding

STEP 1: Collect the stock price from historical data.
STEP 2: Split the collected data into training and test data.
STEP 3: Create the model with training data.



```
Index(['ABBOTINDIA.NS', 'AJANTPHARM.NS', 'AMRUTANJAN.NS', 'AUROPHARMA.NS',
       'CADILAHC.NS', 'CIPLA.NS', 'DIVISLAB.NS', 'DRREDDY.NS', 'GLAXO.NS',
       'GLENMARK.NS', 'GRANULES.NS', 'IPCALAB.NS', 'JUBILANT.NS', 'LUPIN.NS',
       'NECLIFE.NS', 'SANOFI.NS', 'SUNPHARMA.NS', 'TORNTPHARM.NS',
       'ZYDUSWELL.NS'],
      dtype='object')
```

Fig. 6 Various pharmaceutical sector stock price movements

STEP 4: Deploy the Haar Wavelet, find the positive peaks using threshold in wavelet features.

STEP 5: Calculate the Average Interval days between the positive peaks.

STEP 6: Evaluate the result with test data.

Performance Evaluation

This work deployed a decision tree classifier for predicting stock price movement and an accuracy measure for validating the stock price movement. The decision tree classifier is a tree-based classifier used to classify the stock data into profit or loss class. The actual value for the stock is taken from the dataset. The classifier predicted class (profit/loss) is analyzed by using confusion matrix.

The values of the errors (generally known as type 1 and type 2) are calculated with a confusion matrix in Python, Google Colab. The results against the actual value and machine predicted value are shown in Fig. 12. Figure 12 indicates the stock taken for analysis is SUNPHARMA; the stock prices are downloaded online from NSE at dynamic time (SUNPHARMA.NS). The true positive predictions are 478 out of 818.

Shown equity gain is predicted correctly by a decision tree classifier. Using the decision tree classifier, the loss value 337 is predicted correctly, but the trading

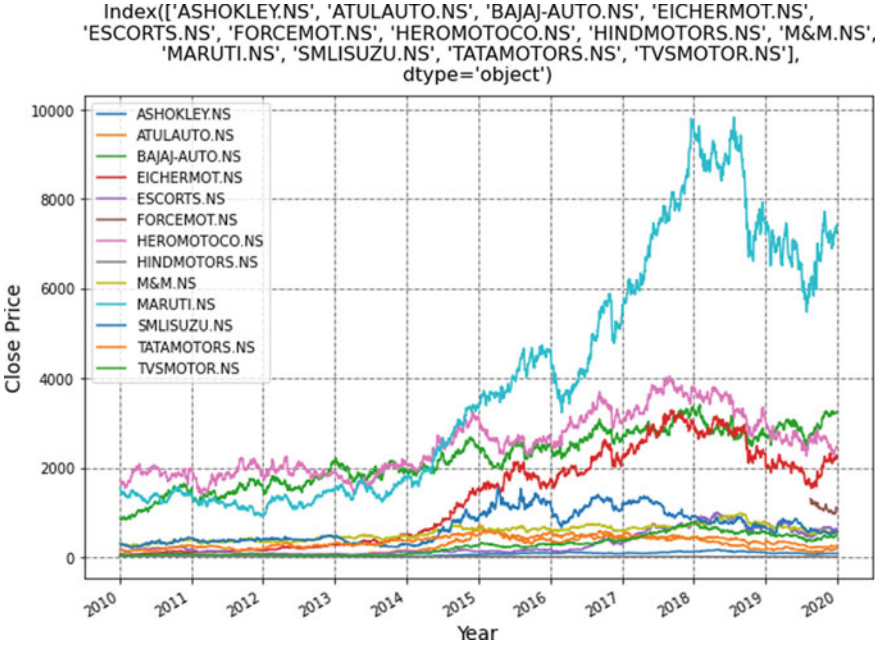


Fig. 7 Various automobile sector stock price movements

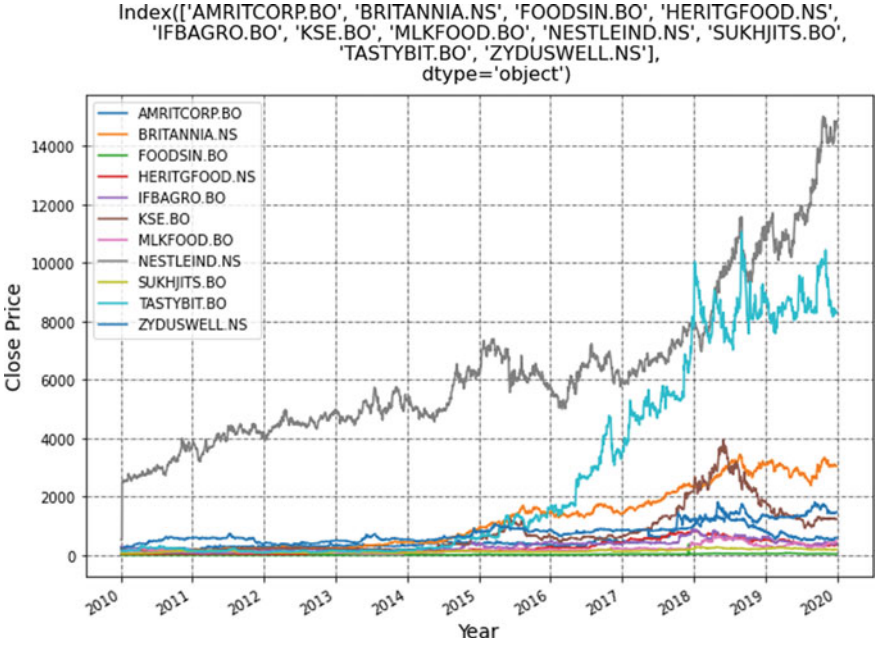


Fig. 8 Various food sector stock price movements

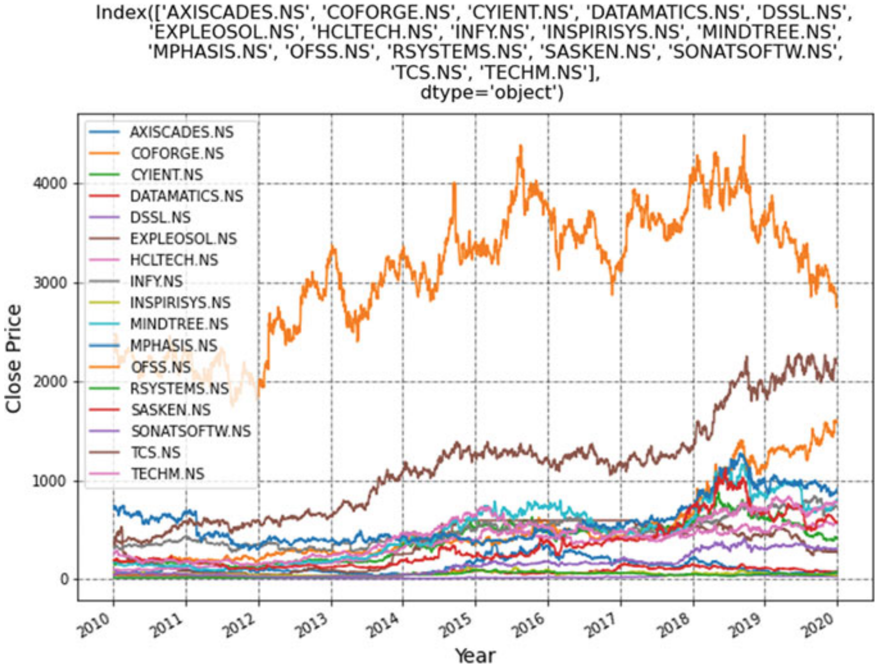


Fig. 9 Various IT sector stock price movements

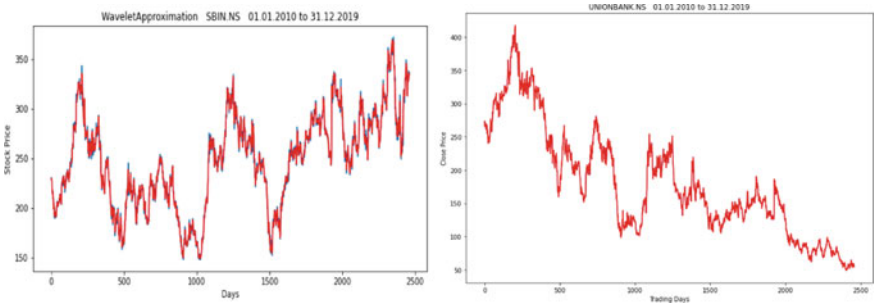


Fig. 10 Various bank sector stock price movements with wavelet features

intervals show its gain. Therefore, it is also taken with type 1 and type 2 errors. Only true positive predictions are considered as correct for the given trading interval.

The analysis of stock prices NSE closing price of trading days between January 1, 2010, to December 31, 2019, resulted in an accuracy measure of 43% (food sector)–60% (pharma sector). The lower accuracy in the food sector may be attributed to either lower trading volumes of the stocks taken and/or the price movements of the stocks failing to reach the given threshold to be considered as profit/equity gain. It

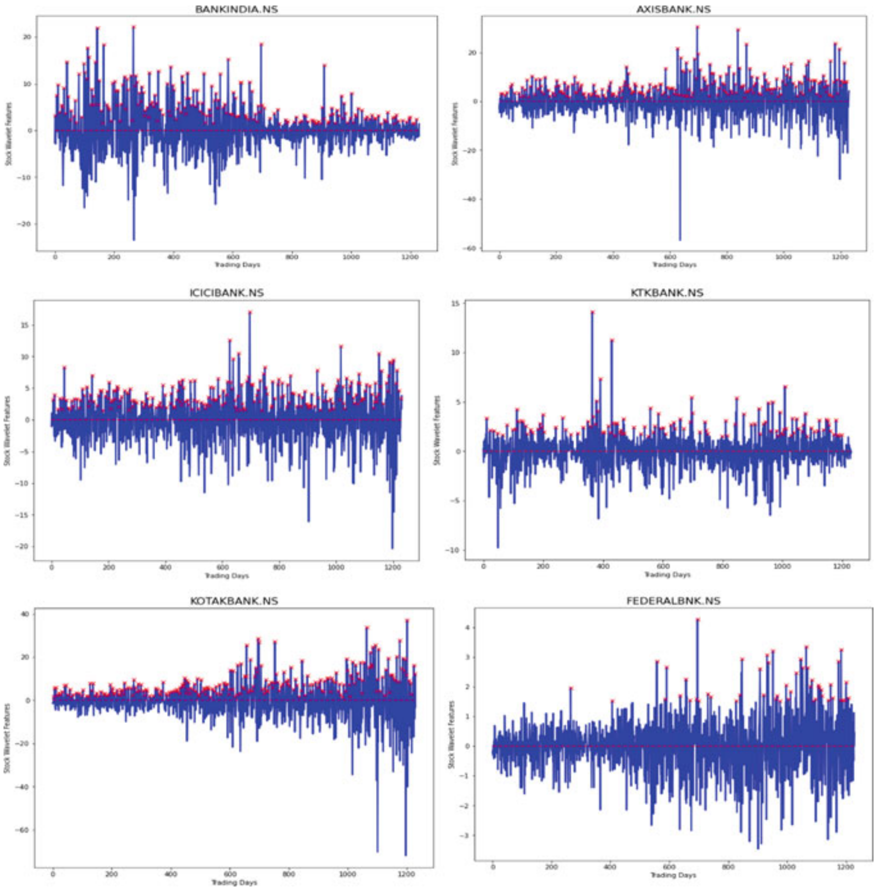
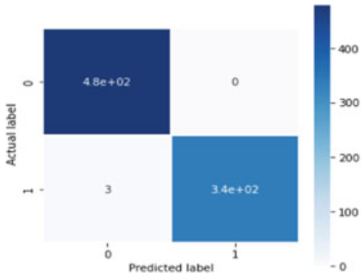


Fig. 11 Bank sector stock price movements—positive peaks predictions



SUNPHARMA.NS
Classification report - Confusion Matrix -
Actual
ProfitLoss
Predicted**Profit** [**478**0]
Loss[3 **337**]

Accuracy 0.58

Total 818, True Positive Predictions 478 (After the predicted interval days 4, equity gain obtained)

Fig. 12 Stock price movements' accuracy calculation for SUNPHARMA

can be inferred that if the stocks from this sector were held for approximately five trading days, the stock investor may get an equity gain notably with 52% accuracy. Hence, the forecasted trading interval days for five sectors considered for analysis are approximately five trading days.

Result and Analysis

In this theme of five different sectors, namely banking, information technology, pharmaceutical, food, and automobiles, ten stocks were taken for each industry for equity gain analysis. The movement of stock is considered from January 1, 2010, to December 31, 2019. All these stocks were analyzed by using Haar Wavelet to find trading interval days. By utilizing these interval days, the stocks for equity gain were analyzed. The empirical results show that by holding on an average of 5 days, we may get equity gain with a probability of 52.92%. According to the effective market hypothesis theory, the stock market will follow a random movement. The accuracy of predicting these movements cannot exceed 50%. Even if a model was to predict the movements in a particular financial market with much higher accuracy, it might be by chance. The same model cannot predict the movements in other financial markets where the movements are random, and the price movements are based on other factors and not just historical prices.

The results section is analyzed by two themes of maximum accuracy and minimum interval days.

Food sector		
Stock name	Interval days	Accuracy (%)
ITC.NS	5	57
GRMOVER.BO	7	56
NESTLEIND.NS	4	56
GODFRYPHLP.NS	**4**	**43**
BRITANNIA.NS	4	53
ATFL.NS	4	52
DFM.BO	5	52
TATACOFFEE.NS	5	52
SUKHJITS.BO	4	51
KRBL.NS	5	50

The bold indicates the accuracy associated with minimum interval days and the interval days associated with maximum accuracy

IT sector		
Stock name	Interval days	Accuracy (%)
DSSL.NS	8	58
HCLTECH.NS	5	56

(continued)

(continued)

IT sector		
Stock name	Interval days	Accuracy (%)
INFY.NS	4	54
RSYSTEMS.NS	**8**	**54**
TCS.NS	4	54
EXPLEOSOL.NS	7	52
SASKEN.NS	4	52
MINDTREE.NS	4	51
MPHASIS.NS	4	51
COFORGE.NS	4	50

The bold indicates the accuracy associated with minimum interval days and the interval days associated with maximum accuracy

Pharma sector		
Stock name	Interval days	Accuracy (%)
AJANTPHARM.NS	**5**	**60**
LUPIN.NS	4	59
SUNPHARMA.NS	4	57
DRREDDY.NS	4	56
GLAXO.NS	**4**	**56**
DIVISLAB.NS	4	55
GLENMARK.NS	4	55
IPCALAB.NS	4	54
CADILAHC.NS	4	53
SANOFI.NS	4	53

The bold indicates the accuracy associated with minimum interval days and the interval days associated with maximum accuracy

Automobile sector		
Stock name	Interval days	Accuracy (%)
EICHERMOT.NS	**4**	**55**
BAJAJ-AUTO.NS	4	54
TATAMOTORS.NS	4	53
HEROMOTOCO.NS	4	51
ASHOKLEY.NS	**5**	**50**
M&M.NS	4	50
MARUTI.NS	4	49

(continued)

(continued)

Automobile sector		
Stock name	Interval days	Accuracy (%)
TVSMOTOR.NS	4	49
ESCORTS.NS	4	48
HINDMOTORS.NS	17	44

The bold indicates the accuracy associated with minimum interval days and the interval days associated with maximum accuracy

Banking sector		
Stock name	Interval days	Accuracy (%)
CUB.NS	**5**	**56**
INDUSINDBK.NS	4	55
AXISBANK.NS	4	53
FEDERALBNK.NS	5	53
HDFC.NS	**6**	**53**
KARURVYSYA.NS	5	53
BANKBARODA.NS	4	52
CANBK.NS	4	52
ICICIBANK.NS	4	52
PNB.NS	5	52

The bold indicates the accuracy associated with minimum interval days and the interval days associated with maximum accuracy

Each sector highlights the research work by each theme, and the accuracy value and minimum interval days serve the objective of this research work. However, this work excluded the stock if it contains any missing values between January 1, 2010, and December 31, 2019. This research work included only the closing price of the stock for each trading day for further analysis.

From the sample that was for analysis, in the pharmaceutical sector, the stock AJANTAPHARM.NS performed the best having a 5-day trading interval with 60% accuracy for getting an equity gain. The stock CADILAHC.NS with a trading interval of 4 days had the lowest accuracy of 53% in the sample. This work predicted an average of five trading interval days for equity gain with an accuracy of 52.9%.

In the sample taken for banking sector, the stock CUB.NS had the highest accuracy of 57% for the derived five trading interval days and four stocks BANKBARODA.NS, CANBK.NS, ICICIBANK.NS, and PNB.NS had the lowest accuracy of 52% for their derived trading interval days. In the sample taken for food sector, the stock ITC.NS had the highest accuracy of 57% for the derived five trading interval days and the stock GODFRYPHLP.NS had the lowest accuracy of 43% for the derived four trading interval days. In the sample taken for IT sector, the stock DSSL.NS had the highest accuracy of 58% for the derived eight trading interval days and the stock COFORGE.NS had the lowest accuracy of 50% for the derived four trading interval days. In the sample taken for automobile sector, the stock EICHERMOT.NS had the highest accuracy of 55% for the derived four trading interval days and the stock COFORGE.NS had the lowest accuracy of 44% for the derived 17 trading interval days. There is a 17% difference in accuracy between the stock's predicted trading interval days. This is mainly due to the difference in the trading volumes of the stocks. Some stocks also failed to touch the defined threshold to be considered as a profit in the wavelet features, and this also affected the accuracy of the stock.

4 Conclusions

The analysis and findings concluded that the Haar Wavelet method used for smoothening and predicting the interval days for the stock price worked very well for forecasting share prices. This work excluded any stock with one or more missing values in the close price for all the sectors as the objective of this research is to find the interval trading days. If a stock is not traded on a particular trading day, it does not align with this research work's objective of finding minimum trading interval days and may lead to erroneous results.

Further empirical result analysis of selected stocks of various sectors and companies confirmed the price movements with accuracy. This model, when used with care, may benefit investors. This work will reduce the time in terms of the holding period. An investor can get equity gain with minimum interval days for a given threshold. Further research can be done to include stocks with missing close prices on trading days to improve accuracy within each sector.

References

Angadi MC, Kulkarni AP (2015) Time series data analysis for stock market prediction using data mining techniques with R. Int J Adv Res Comput Sci 6(6):104–108

Boopalan C (2014) Technical analysis in select stocks of Indian companies. Int J Bus Admin Res Rev 2(4):26–36

Chkraborty M (2006) On validity of random walk hypothesis in colombo stock exchange, Sri Lanka. Decision 33(1):135–161

Crowley PM (2005) An intuitive guide to wavelets for economists. Bank of Finland Research discussion paper no, 1/2005. https://doi.org/10.2139/ssrn.787564

Chandar K, Mahadevan S, Sivanandam S (2016) Prediction of stock market price using hybrid of wavelet transform and artificial neural network. Indian J Sci Technol 9. https://doi.org/10.17485/ijst/2016/v9i8/87905

Daubechies I (1990) The wavelet transform: time–frequency localization and signal analysis IEEE Trans Inf Theor 36:961–1005

Fama EF (1991) Efficient capital markets II. J Fin 46(5):1575–1617.

Gencay R, Selcuk F, Whitcher B (2002) An introduction to wavelets and other filtering methods in finance and economics. Academic Press, New York

Graps A (1995) An introduction to wavelets. In: IEEE conference on computational science and engineering, summer, vol 2, no 2. IEEE Computer Society, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720, USA

Gupta N, Gedam A (2014) Testing of efficient market hypothesis: a study on Indian stock market. IOSR J Bus Manage 16(8):28–38

Hagenau M, Liebmann M, Neumann D (2013) Automated news reading: stock price prediction based on financial news using context capturing features. Decis Support Syst 55(3):685–697

Hsieh T-J, Hsiao H-F, Yeh W-C (2011) Forecasting stock markets using wavelet transforms and recurrent neural networks: an integrated system based on artificial bee colony algorithm. Appl Soft Comput 11:2510–2525

Hu Z, Zhao Y, Khushi M (2021) A survey of forex and stock price prediction using deep learning. Appl Syst Innov 4:9. https://doi.org/10.3390/asi4010009

Huang S-C (2011) Forecasting stock indices with wavelet domain kernel partial least square regressions. Appl Soft Comput 11:5433–5443

Huang SC, Wu TK (2008) Combining wavelet-based feature extractions with relevance vector machines for stock index forecasting. Expert Syst 25:133–149

Jothimani D, Shankar R, Yadav SS (2015) Discrete wavelet transform—based prediction of stock index, a study on national stock exchange fifty index. J Fin Manage Anal 28(2):35–49

Kao L-J, Chiu C-C, Lu C-J, Chang C-H (2013) Chang a hybrid approach by integrating wavelet-based feature extraction with MARS and SVR for stock index forecasting

Khanna M (2014) Value relevance of accounting information: an empirical study of selected Indian firms. Int J Sci Res Publ 4(10):1–6

Kishikawa Y, Tokinaga S (2000) Prediction of stock trends by using the wavelet transform and the multi-stage fuzzy inference system optimized by the GA. IEICE Trans Fundamentals Electron Commun Comput Sci 83(2):357–366

Kloptchenko TE, Back B, Karlsson J, Vanharanta H, Visa A (2004) Combining data and text mining techniques for analysing financial reports. Int J Intell Syst Account Fin Manage 12(1):29–41

Lahmiri S (2013) Forecasting direction of the S&P 500 movement using wavelet transform and support vector machines. Int J Strateg Dec Sci 4:78–88

Lahmiri S (2014) Wavelet low- and high-frequency components as features for predicting stock prices with backpropagation neural networks. J King Saud Univ Comput Inf Sci 26(2):218–227. ISSN 1319-1578

Lai KK, Huang J (2007) The application of wavelet transform in stock market

Lee DTL, Yamamoto A (1994) Wavelet analysis: theory and application. Hewlet Packard J: 44–52

Li J, Shi Z, Li X (2006) Genetic programming with wavelet-based indicators for financial forecasting. Trans Inst Meas Control 28:285–297

Mala S, Saravanan S (2018) Stock market prediction system: a wavelet based approach. Appl Math Inf Sci 12(3):579–585

Matei M (2009) Assessing volatility forecasting models: why GARCH models take the lead. J Econ Forecast 4:42–65

Miranda B, Vinicius H, Sobreiro A, Kimura H (2018) Stock price prediction using support vector regression on daily and up to the minute prices. J Fin Data Sci 4(3):183–201

Mousazadeh Abbasi N, Aghaei MA, Moradzadeh Fard M (2015) Forecasting stock market using wavelet transforms and neural networks and ARIMA (case study of price index of Tehran stock exchange). Int J Appl Oper Res 5(3):31–40

Ntim C, Opong K, Danbolt J, Dewotor F (2011) Testing the weak-form efficiency in African stock markets. Manag Financ 37:195–218. https://doi.org/10.1108/03074351111113289

Patel J, Shah S, Thakkar P, Kotecha K (2015) Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Expert Syst Appl 42(1):259–268

Ramsey JB (1999) The contribution of wavelets to the analysis of economic and financial data. Philos Trans R Soc Lond Ser A Math Phys Eng Sci: 2593–2606

Simons D, Laryea S (2006) The efficiency of selected African stock markets. Fin India 20(2):553–571

Shah FA, Debnath L (2017) Wavelet neural network model for yield spread forecasting. Mathematics (MDPI) 5(72):1–15

Stankovic J, Markovic I, Stojanovic M (2015) Investment Strategy Optimization using Technical Analysis and Predictive Modeling in Emerging Markets', Procedia Economics And Finances, vol. 19, pp. 51-62

Sudheer V (2015) Trading through technical analysis: an empirical study from Indian stock market. Int J Dev Res 5(8):5410–5541

Umaprabha S, Malavika M (2015) A study on technical analysis of stocks listed in NSE with reference to pharmaceutical industries. Int J Manage (IJM) 6:79–86

Wafia AS, Hassana H, Mabrouka A (2015) Fundamental analysis models in financial markets—review study. Procedia Econ Fin 30:939–947

Wang YJ, Liu H, Guo Q et al (2019) Stock volatility prediction by hybrid neural network. IEEE Access 7:154524–154534

Wang J-Z, Wang J-J, Zhang Z-G, Guo S-P (2011) Forecasting stock indices with back propagation neural network. Expert Syst Appl 38:14346–14355

Intelligent Camera-Based Eye-Controlled Wheelchair System: Raspberry Pi and Advanced Algorithms

V. Viswanatha, A. C. Ramachandra, Gowkanapalli Lokeshwar Reddy, A. V. Sai Tharun Reddy, Biyyam Pranay Kumar Reddy, and Gavvala Bhanu Kiran

1 Introduction

Older people and others who are physically disabled rely on wheelchairs. The design and execution models of an entirely eye-controlled wheelchair are as shown in Fig. 1. Different automatic systems, such as voice-controlled wheelchair or joystick-controlled wheelchair systems, are offered on the market depending on the needs of people with impairments (Tharwat et al. 2022; Patel and Prakash 2015). People who are completely paralyzed occasionally find it very complex to use that kind of equipment. Here, the eye-controlled wheelchair system gives them independence they need to live more independently and comfortably. Additionally, they conserve a significant amount of energy and outside labour. Real-time picture analysis from the camera is used to send the commands for communicating with the motor driver to operate the connection through GPIO pins. With the help of motor driver, various operations can carry out, including left, right, advance, and stop (Viswanatha et al. 2022; Culpepper and Keller 2003).

The face and eye detections are done by using OpenCV at the advanced level of image processing. For positioning the accurate pupil, tracking and detection, a variety of applications and algorithms are used. One of these, the Haar cascade detection technique, is utilised to detect both eyes as well as single and many faces (Jaffe 1982; Abe et al. 2002). The final objective of this technique is to identify the precise eye pupil and pinpoint its centre. OpenCV libraries like cv2, numpy are used for image processing like image colour conversion, pattern matching edge detection, object detection, and edge detection are used to automatically determine and track the eye pupil. There are numerous more methods for tracking eye pupils. However,

V. Viswanatha (✉) · A. C. Ramachandra · G. L. Reddy · A. V. S. T. Reddy · B. P. K. Reddy · G. B. Kiran
Nitte Meenakshi Institute of Technology, Bengaluru, India
e-mail: viswanatha.v@nmit.ac.in

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_3

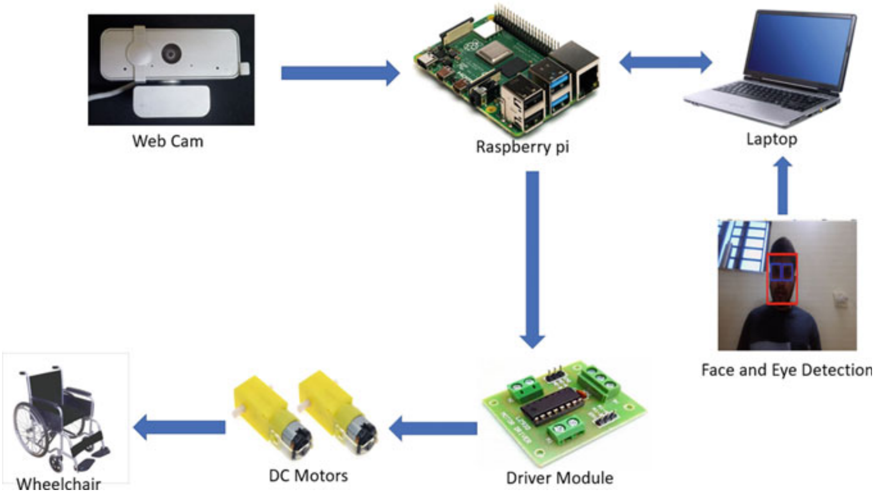


Fig. 1 Block diagram

they have their own restrictions. However, different outputs would be produced for various users, leading to incorrect pupil position in the eye (Viswanatha et al. 2020; Kuno et al. 1998; Robinson 1963).

The camera sends the images that it has captured to the base station, which could be a computer or laptop. After identifying the eyes using trained Haar cascade-like features for eye detection, the images are processed using image processing methods in Python utilizing the Open-source Computer Vision Library. With the use of the Hog method and 68 points face detection, image processing can pinpoint the pupil's location. Two blocks make up the full frame (Ito 2002; Kishimoto 2001; Reddy and Rajeswari 2019; Corno and Farinetti 2002).

The algorithm specifies each block's width and length. Through repeated analysis, we may determine the direction in which the eye is looking by calibrating the block in which the pupil is located. Processing essentially separates the image into three sections, left, right, and centre. The output is "Right" if the pupil's position is in the first block. It produces the result "Left" if it is located in the third block (Abe and Oi 2003, 2004; Ramachandra et al. 2022; Barea et al. 2002).

The output is "Straight" if the pupil is in the centre. The default response upon three eye blinks is "Stop." The Raspberry Pi is then informed of this output.

1.1 *Motivation*

A wheelchair is required by over 132 million disabled persons worldwide, or 1.86% of the total population. Millions of people worldwide have the inability to control any of their limbs or even their head due to diseases associated with upper and lower motor neurons, thereby weakening the muscles. Therefore there is a need to build the system with better technology to give disabled people, wonderful opportunities to feel autonomous and lead accessible lives. For the system based on eye control and cost-effective, a revolutionary technique is used. To reduce the need for the disabled person's assistance, we strive to suggest an electric wheelchair with eye movement bias control (Sarangi et al. 2007; Wang and Ishimatsu 2005; Jia and Hu 2005; Viswanatha and Reddy 2017; Purwanto et al. 2008).

1.2 Objectives

- Controls the wheelchair based on the eye movements and detection of obstacle in the path.
- Highlight the features of digital image processing and represents the alternative resources available.
- Learn use of algorithms like Haar cascade, Canny edge detection, and Hough transform to track the eye movements.
- Use Python 3.0 and OpenCV library installed into Raspberry Pi to do image processing.

2 Methodology

Eye pupil detection and eye tracking are the system's fundamental concepts. These are achieved through various image processing techniques. The actual working methodology of this system is as follows—detecting human face and eyes using Haar cascade algorithm, eyes as region of interest, webcam captures the images of eye and the pupil movement detection by using Canny edge detection and Hough Transform. After tracking the eye movement, the chair is instructed to move accordingly with the help of driver module. As a measure of safety, an ultrasonic sensor is used to avoid the obstacles on the way by detecting them in well advance. The actual working of algorithms mentioned is shown in Fig. 2.

2.1 Haar Cascade Algorithm

This algorithm contains "Haar Wavelet transfer functions" as shown in Fig. 3 serve as the foundation of Haar cascade and are described as a series of rescaled "square-shaped" functions that collectively make up a wavelet family or basis.

The working flow of Haar Wavelet is shown in Fig. 4. It uses the Haar Wavelet approach to divide the image's pixels into squares according to their functions. To obtain a high level of accuracy from what is referred to as "training data," this involves

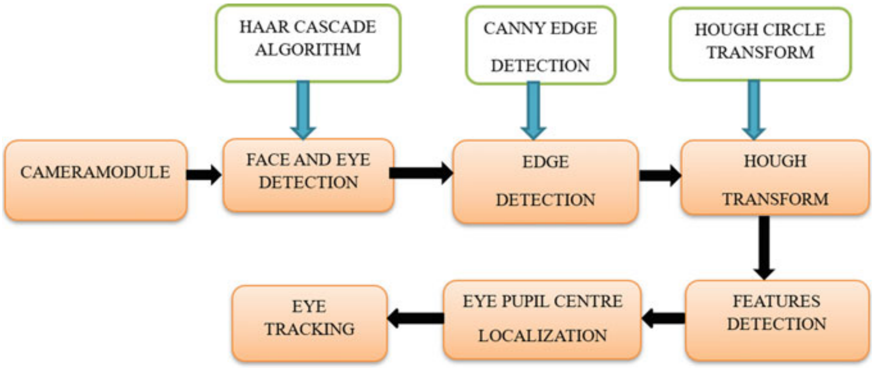


Fig. 2 Architecture of methodology

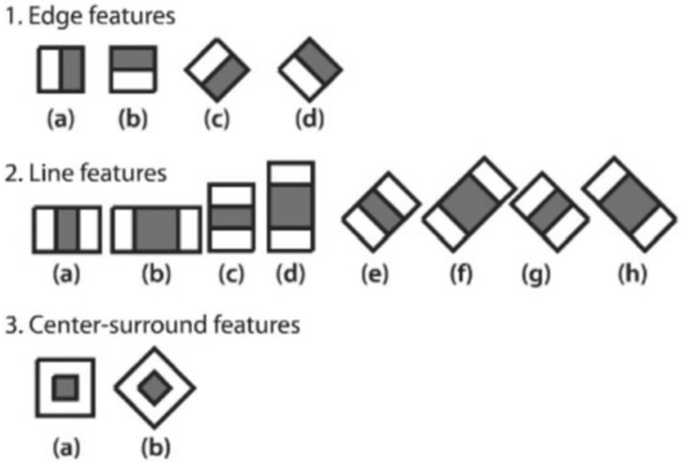


Fig. 3 Haar wavelets

machine learning techniques. The "features" that are detected are computed using "integral image" principles. The AdaBoost learning method, which is used by Haar cascades to provide effective classifier results, chooses a small number of crucial features from a huge set.

The workflow of Haar Cascade Algorithm is as shown in Fig. 4. In this context, it involves the key steps, starting from taking clean data followed by training the classifer to real time face detection. Once face detection is success, it will move on to find the region of interest followed by eye detection on the face of human.

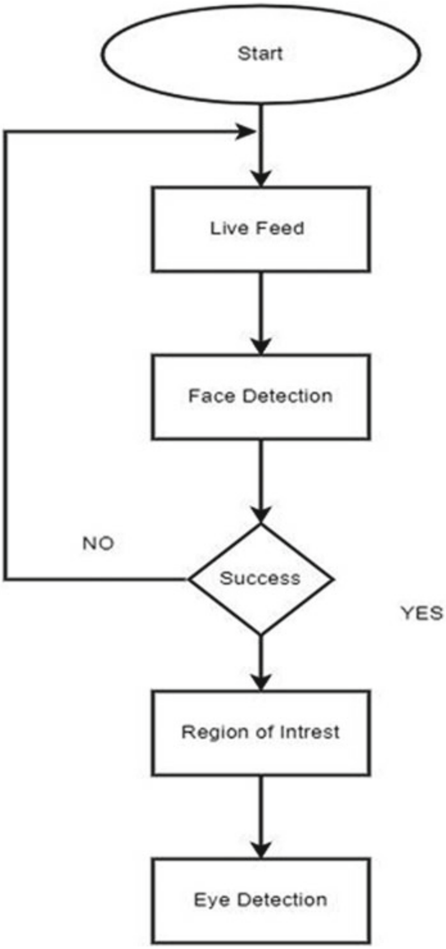


Fig. 4 Flowchart of Haar cascade algorithm

2.2 *Canny Edge Detection*

The Canny edge detection technology helps to reduce size of data that must be processed and still obtains the structured information of various objects. It is commonly employed in different systems that use OpenCV. According to Canny edge detection algorithm, if necessary conditions like gaussian smoothing, edge tracking by hysteresis are met for an object, most computer vision based systems can use edge detection for it. The algorithm's internal process steps are shown in Fig. 5.

The steps shown in the above flowchart of Canny edge detection are as follows:

1. Gaussian smoothing uses a Gaussian filter for the removal of noise.
2. In gradient filtering, x and y directions of image are calculated by convolution mask method and gradient strength and edge's direction are calculated.

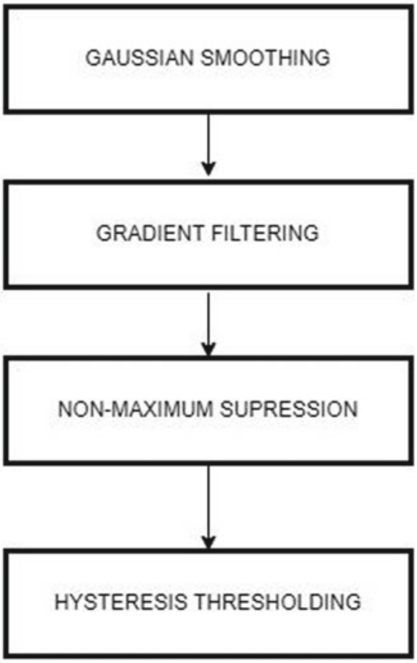


Fig. 5 Flowchart of Canny edge detection

3. Non-maximum Suppression step is used to remove the unwanted pixels which are not edge's part.
4. Hysteresis in this step upper threshold and lower threshold is used as follows.

   (i) Upper threshold where the gradient value is more for a pixel is considered as edge pixel.
   (ii) Lower threshold where the gradient value is less for a pixel is rejected.
   (iii) If the value is in between two thresholds then, the pixel is accepted if and only if the surrounding/neighbouring pixel is higher than upper threshold.

Some general requirements for edge detection.

1. The error rate should be very low such that the algorithm captures visible edges in image precisely.
2. The centre of the edge will be correctly localized by the operator's detection of edge point.
3. False edges are not to be produced by the image noise and each edge in the image should only be marked once in the image.

2.3 *Hough Transform*

This method is used to seclude the information of shape like circle or lines in a picture as shown in Fig. 6. This Hough transform method is mostly used for the detection of normal curves like circles, lines, and ellipses and so on.

3 Implementation

The implementation of the system is shown in Fig. 7. It is OS-based system which collects the live data. A low-power on-board computer-based Raspberry Pi is used. This provides some input and output pins and some useful ports like UART, HDMI, USB, and an Ethernet adapter port to connect over the internet via a wired and can also be connected wirelessly. Raspberry Pi also supports external memory up to 32 GB

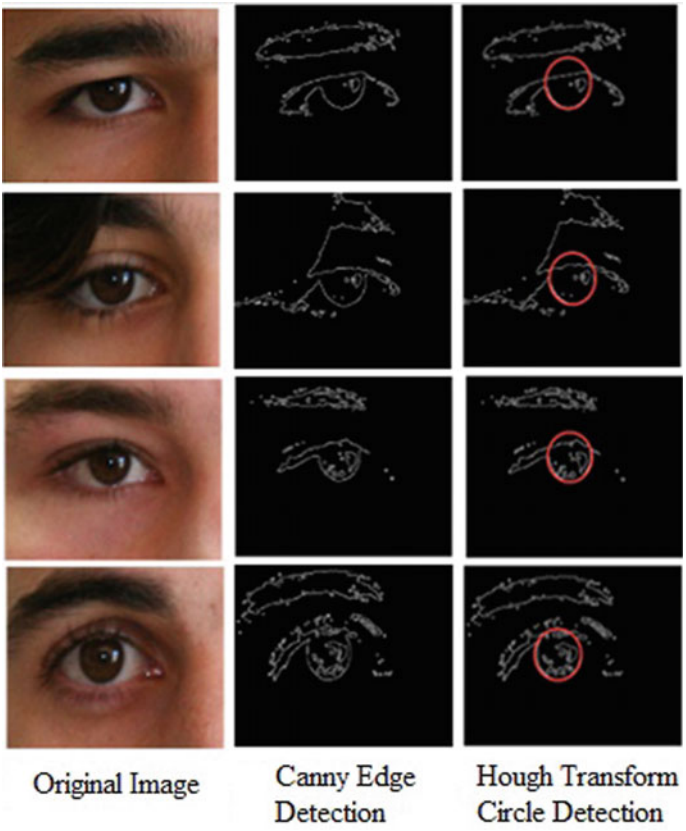


Fig. 6 Edge and pupil detection

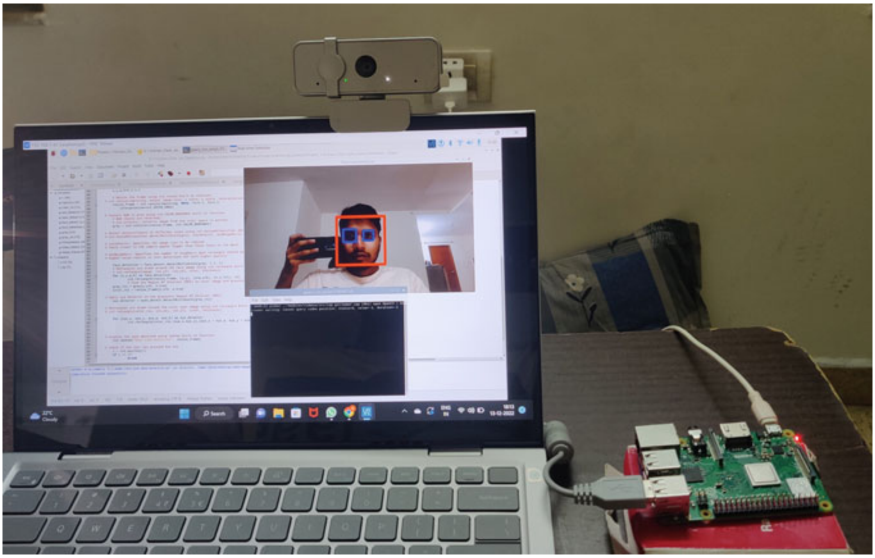


Fig. 7 Experimental setup

and has RAM of 4 GB. It is controlled with the help of ARM architecture. Figure 7 represents the experimental setup of system. A web camera is placed in before the user's eye. Here, the crucial part is the length between eyes and the camera is set that can be 9–12 cm. The camera captures the images of person face and determines the eyeball position accurately. After identification of pupil position, the algorithm measures the mean from outer edge/corner of eye. This provides the right information of eyeball movement and accordingly it controls the motor driver along with a relay board to control the motor connected to the wheelchair.

The motor driver circuit is connected to Raspberry Pi 4 and also a camera module which captures the images and then produces the command signal to the GPIO pins. Perform the left, right, forward, and stop operations.

4 Result and Discussion

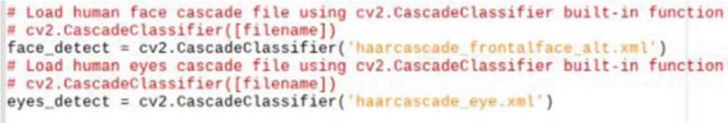
The results obtained at each stage of image processing are shown step by step. The webcam detects the face and eyes, creating Regions of Interest (ROIs) by capturing images using the Haar Cascade algorithm. By using the Canny edge detection, the next process which converts the image from BGR to grey removes the noise and detects edges. Then, the circles which are pupils in our case are detected by Hough transform. This pupil movement or direction information is sent to the driver motor to drive the chair in the respective direction. The outputs obtained in each stage are shown as.

4.1 Haar Cascade Algorithm for Face and Eye Detection

The face and eyes are detected using the Haar cascade algorithm, and its output as shown in Fig. 8, featuring rectangular boxes around the regions of interest. Subsequently, images of the eyes, identified as the Region of Interest (ROI) in this case, are captured. This dataset is used for training in edge detection and Hough transform for pupil location.

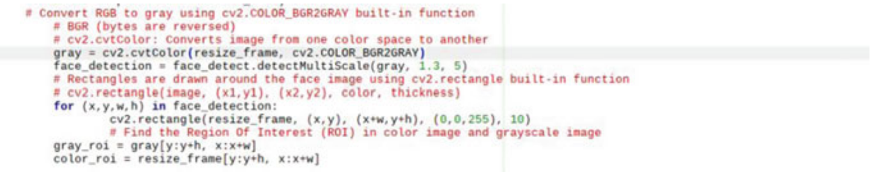
The important libraries used in the image processing are imported first, importing CV2, and numpy is must. For detection of face and eye, a html file containing the features of face and its regions should be installed before performing detection algorithm. The important parts of coding with final output are shown below:

The below function is used to load face and eye features.



```
# Load human face cascade file using cv2.CascadeClassifier built-in function
# cv2.CascadeClassifier([filename])
face_detect = cv2.CascadeClassifier('haarcascade_frontalface_alt.xml')
# Load human eyes cascade file using cv2.CascadeClassifier built-in function
# cv2.CascadeClassifier([filename])
eyes_detect = cv2.CascadeClassifier('haarcascade_eye.xml')
```

Face detection is done by using inbuilt functions which are as follows.



```
# Convert RGB to gray using cv2.COLOR_BGR2GRAY built-in function
    # BGR (bytes are reversed)
    # cv2.cvtColor: Converts image from one color space to another
    gray = cv2.cvtColor(resize_frame, cv2.COLOR_BGR2GRAY)
    face_detection = face_detect.detectMultiScale(gray, 1.3, 5)
    # Rectangles are drawn around the face image using cv2.rectangle built-in function
    # cv2.rectangle(image, (x1,y1), (x2,y2), color, thickness)
    for (x,y,w,h) in face_detection:
            cv2.rectangle(resize_frame, (x,y), (x+w,y+h), (0,0,255), 10)
            # Find the Region Of Interest (ROI) in color image and grayscale image
    gray_roi = gray[y:y+h, x:x+w]
    color_roi = resize_frame[y:y+h, x:x+w]
```

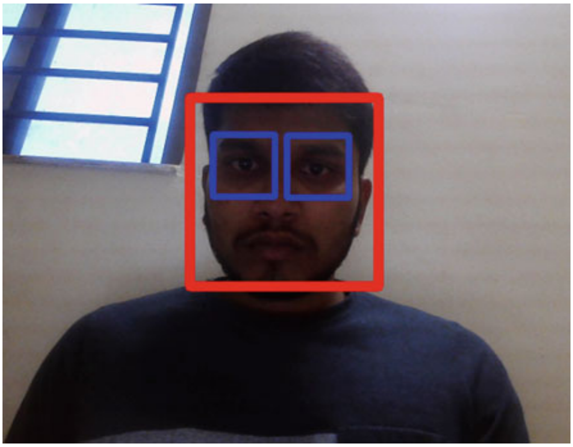


Fig. 8 Output of detection of face and eye

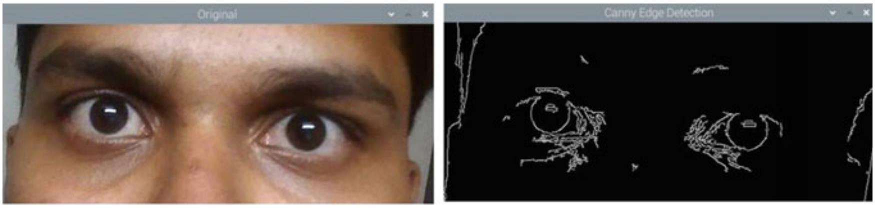
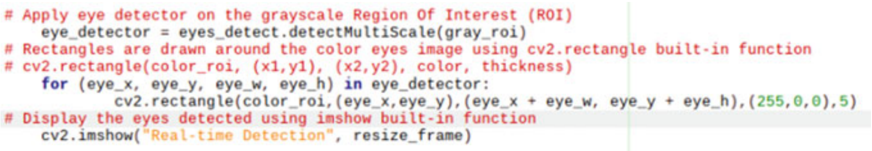


Fig. 9 Output of canny edge detection

Eye detection is done by using the inbuilt functions which are as follows.

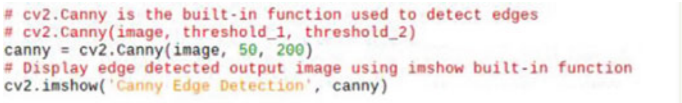


```
# Apply eye detector on the grayscale Region Of Interest (ROI)
    eye_detector = eyes_detect.detectMultiScale(gray_roi)
# Rectangles are drawn around the color eyes image using cv2.rectangle built-in function
# cv2.rectangle(color_roi, (x1,y1), (x2,y2), color, thickness)
    for (eye_x, eye_y, eye_w, eye_h) in eye_detector:
            cv2.rectangle(color_roi,(eye_x,eye_y),(eye_x + eye_w, eye_y + eye_h),(255,0,0),5)
# Display the eyes detected using imshow built-in function
    cv2.imshow("Real-time Detection", resize_frame)
```

4.2 Canny Edge Detection

The edges are detected from the images captured in Haar cascade and it is shown in Fig. 9. Here, detecting the edges avoids the noise in the picture which is not necessary in detecting the pupils which are circle objects which can be detected in next step of the process.

The edge detection can be done by using the inbuilt function available in CV2 library as mentioned in the below function.



```
# cv2.Canny is the built-in function used to detect edges
# cv2.Canny(image, threshold_1, threshold_2)
canny = cv2.Canny(image, 50, 200)
# Display edge detected output image using imshow built-in function
cv2.imshow('Canny Edge Detection', canny)
```

4.3 Hough Transform

Here, Hough transform is used to detect circles which can also be used to detect lines. Here, in this case, pupils which are circle in shape detected by this algorithm are shown in Fig. 10.

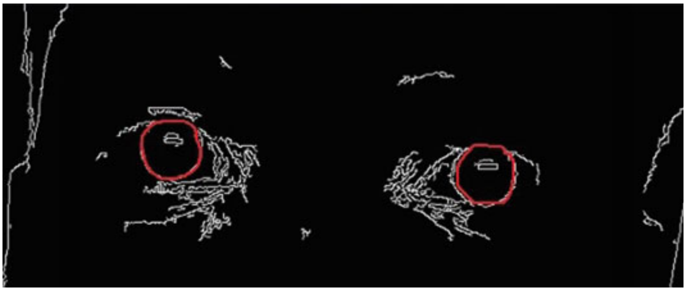
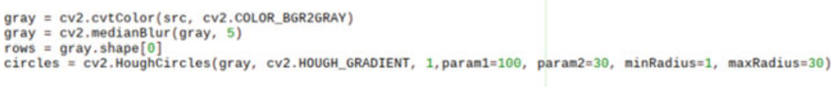


Fig. 10 Output of Hough transform



```
gray = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)
gray = cv2.medianBlur(gray, 5)
rows = gray.shape[0]
circles = cv2.HoughCircles(gray, cv2.HOUGH_GRADIENT, 1,param1=100, param2=30, minRadius=1, maxRadius=30)
```

Detection of circles can be done by using the Hough function that is available in CV2 library as mentioned in the instructions as follows.

4.4 Driving Motor

The system takes the resulting image processing data and sends it to a motor drive circuit for moving the wheelchair based on the average pupillary signal of the eye. There, the system as shown in Fig. 11 uses ultrasonic sensors to detect obstacles. Succeeded in measuring the length between the obstacle on the way and the wheelchair, where the obstacle is detected directly in front of the wheelchair and if it exceeds a distance threshold, emergency braking is used to halt the system.

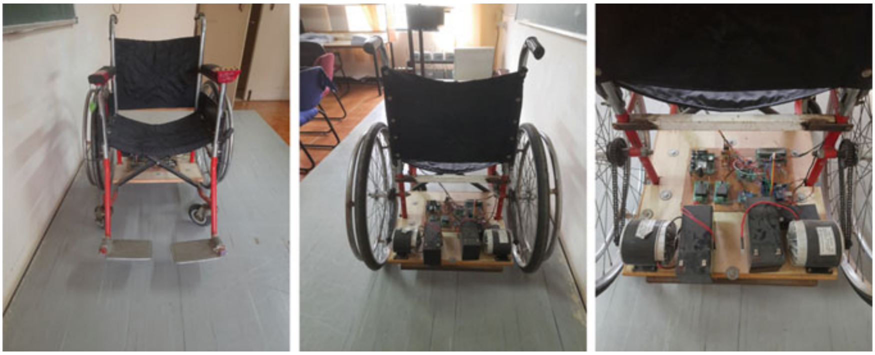


Fig. 11 Wheelchair setup with electric motors powered by battery

5 Conclusion

The implemented wheelchair system is not only important as an alternative resource but also assists individuals with physical disabilities in leading independent lives. The sole purpose for implementation of proposed wheelchair system is to emphasize the properties of image processing. Time needed for the system for video is to be processed in a real-time environment, so there are some real-time design constants that are calculated. Therefore, the system executes the wheelchair transfer process with a constant delay time. Since in low light it is difficult to track the pupil movement in eye, the system works perfectly in fluorescent mercury room lighting, which contains little ambient light or infrared light.

References

Abe O,Oi D (2003) The look input system using the sclera reflection method by image analysis. Lett Inst Image Inf Telev Eng 57(10):1354–1360

Abe D, Oi (2004) The look input platform for serious physically handicapped persons. In: Human interface society human interface symposium 2004 collected papers, pp 1145–1148

Abe K, Ohiamd S, Ohyama M (2002) An eye-gaze input system based on the limbus tracking method by image analysis for seriously physically handicapped people. In: Proceedings of the 7th ERCIM workshop user interface for all adjunct proceeding, pp 185–186

Barea R, Boquete L, Mazo M, Lopez E (2002) System for Assisted Mobility using Eye Movements based on Electrooculography. IEEE Trans Neural Syst Rehabil Eng 10(4):209–218

Corno L, Farinetti IS (2002) A cost-effective solution for eye- gaze assistive technology. In: Proceedings of the IEEE international conference on multimedia and expo, vol 2, pp 433–436

Culpepper BJ, Keller RM (2003) Enabling computer decisions based on EEG input. IEEE Trans Neural Syst Rehabil Eng 11:354–360

Ito N (2002) Eye movement measurement by picture taking in and processing via a video capture card, an Institute of electronics. Inf Commun Eng Tech R 102:128, 31–36

Jaffe DL (1982) An ultrasonic head position interface for wheelchair control. J Med Syst 6(4):337–342

Jia P, Hu H (2005) Head gesture based control of an intelligent wheelchair. In: Proceedings of the 11th annual conference of the chinese automation and computing society in the UK [CACSUK05], pp 85–90

Kishimoto Y, Hirose C (2001) Development of the look input system by a cursor move system. Lett Inst Image Inf Telev Eng 55(6):917–919

Kuno Y, Fujii K, Uchikawa (1998) Development of the look input interface using EOG. In: The information processing society of Japan paper magazine C39C5, pp 1455–1462

Patel SN, Prakash V (2015) Autonomous camera based eye controlled wheelchair system using raspberry-pi. In: 2015 International conference on innovations in information, embedded and communication systems (ICIIECS), pp 1–6. https://doi.org/10.1109/ICIIECS.2015.7192876

Purwanto D, Mardiyanto R, Arai K (2008) Electric wheelchair control with gaze direction and eye blinking. In: Proceedings of the fourteenth international symposium on artificial life and robotics, GS21-5, B- Con Plaza, Beppu

Ramachandra AC, Viswanatha V, Kishor K, Suhas H, Pannir Selvam E (2022) In-cabin radar monitoring system: detection and localization of people inside vehicle using vital sign sensing algorithm. Int J Recent Innov Trends Comput Commun 10(8):104–109. https://doi.org/10.17762/ijritcc.v10i8.5682

Reddy VV, Rajeswari VS (2019) Stability and dynamic response of analog and digital control loops of bidirectional buck-boost converter for renewable energy applications. Int J Recent Technol Eng (IJRTE). Available at SSRN: https://ssrn.com/abstract=3515804

Robinson DA (1963) A method of measuring eye movement using a sclera search coil in a magnetic field. IEEE Trans Biomed Electron 10:137–145

Sarangi P, Grassi V, Kumar V, Okamoto J (2007) Integrating human input with autonomous behaviors on an intelligent wheelchair platform. J IEEE Intell Syst 22(2):33–41

Tharwat M, Shalabi G, Saleh L, Badawoud N, Alfalati R (2022) Eye-controlled wheelchair. In: 2022 5th international conference on computing and informatics (ICCI), pp 097–101. https://doi.org/10.1109/ICCI54321.2022.9756116

Viswanatha V, Reddy RVS (2017) Digital control of buck converter using arduino microcontroller for low power applications. In: 2017 international conference on smart technologies for smart nation (SmartTechCon). IEEE

Viswanatha V, Venkata Siva Reddy R, Rajeswari (2020) Research on state space modeling, stability analysis and PID/PIDN control of DC–DC converter for digital implementation. In: Sengodan T, Murugappan M, Misra S (eds) Advances in electrical and computer technologies. Lecture Notes in Electrical Engineering, vol 672. Springer, Singapore

Viswanatha V, Ramachandra AC, Venkata Siva Reddy R (2022) Bidirectional DC-DC converter circuits and smart control algorithms: a review. J Electr Syst Inf Technol 9(1):1–29

Wang H, Ishimatsu T (2005) Vision-based navigation for an electric wheelchair using ceiling light landmark. J Intell Rob Syst 41(4):283–314

TinyML-Based Human and Animal Movement Detection in Agriculture Fields in India

V. Viswanatha, A. C. Ramachandra, Puneet T. Hegde, Vivek Hegde, and Vageesh Sabhahit

1 Introduction

TinyML attempts to remove the conventional power barrier that inhibits broadly dispersed machine intelligence by bringing ML model on ultra-low-power IoT devices, generally near to milli-watt. TinyML allows increased reactiveness while eliminating the energy cost involved with wireless connectivity, which is significantly greater than that of computers, by doing offline inference close to the data source. Given the importance of TinyML for future technologies, the area needs a widely regarded benchmark to reach its full potential. Recent developments in Internet of Things (IoT) technology have made a variety of time- and data-dependent applications possible, including virtual reality, embedded computer vision, and autonomous driving. Due to the frequent need for complex processing of huge amounts of sensor data under strict time limitations, the use of deep neural network (DNN) algorithms on MCU and edge devices is an emerging trend. TinyML's growing popularity has contributed to some industry maturity, resulting in the release of software stacks like EdgeML, Open Neural Network, Restricted Coulomb Energy, and CNN's TensorFlow Lite inference. The TFLite in particular draws attention since it enables the simple and portable execution of neural nets on memory constrained MCUs.

TFLite provides portability and minimal memory overhead on MCUs. Additionally, TFLite executes a NN graph via an interpreter, enabling the deployment of the same model graph across many IoT platforms, including GPUs and MCUs. NNs (CNNs, DNNs, and LSTMs) are the dominant force in machine learning; however, non-NN-based ML solutions also play a significant role in TinyML due to their low-power and memory-intensive processing (TinyML Benchmark 2021). And we must address a number of major issues, including as heavy consumption of power, network throughput, stability of network and performance, latency, resilience, and

V. Viswanatha (✉) · A. C. Ramachandra · P. T. Hegde · V. Hegde · V. Sabhahit
Nitte Meenakshi Institute of Technology, Bengaluru, India
e-mail: viswanatha.v@nmit.ac.in

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_4

Table 1 Comparison between ML on different platforms

Entity	Comparison		
	Cloud ML	Mobile ML	TinyML
Algorithm	DNN	Optimized algorithm like CNN, MobileNets, etc.	CNN-micro
RAM	Large models in terms of gigabytes (HBM)	Constrained resources 4 to 8 GB RAM (DRAM)	Severely constrained 100 KB RAM (SRAM)
Model	AlexNets, Inception, ResNets, VGG Nets	MobileNetsV1, MobileNetV2, ShuffleNets, ImageNets	MobileNets and TFLite micro
Accuracy	High accuracy	Focus on accuracy and efficiency	Low accuracy increase in privacy
ROM	SSD/disk in TB-PB	Flash 64-128 GB	eFlash 1 MB
Power	−250 W	−5–8 W	−0.05 W

security, i.e., data privacy that arise when ML inference is moved closer to the edge. The TinyML (as per Table 1) paradigm is still in its infancy and must be properly aligned to interact with current Edge-IoT frameworks.

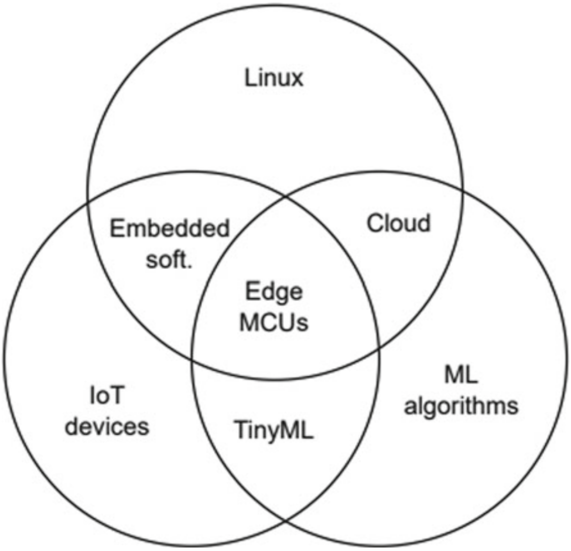
2 Literature Survey

The popularity of TinyML is growing among academic and business circles (Shafique et al. 2021). A recent survey outlined successes and barriers for advancing embedded ML. Improved algorithms are made to use resources more effectively, enhance model performance, and deployment optimization in a real-world setting. Additionally, ultra-low-power AI processors and accelerators have been suggested to allow always-on ML capabilities for a prolonged length of time powered by a battery. To get the best results from TinyML's machine learning solutions for devices with limited resources, it is important to work together to design both the hardware and the algorithm to maximize performance. Integration of different ML technologies is shown in Fig. 1.

Visual attention condensers are used into carry out deep picture recognition (Wang et al. 2023). AttendNets is a low-precision on-device image recognition system. It makes advantage of micro–macrostructures that were generated by a machine throughout the design process. To demonstrate the importance of the AttendNets, it is measured against the ImagNet50 benchmark. Results indicate that it reduces 3x multiply–add operations and increases accuracy by 7.2%. In comparison with MobileNetV1, it also consumes 16.7 times less RAM and 4.17 times fewer parameters. The framework offers the best option between the accuracy and network efficiency trade-offs. By utilizing automated micro–macro-architecture elements, it enhances spatial-channel selective attention.

The paper provides a higher level, as independent of hardware and software as feasible abstraction of TinyML software using TinyMLaaS. Additionally, they have

Fig. 1 Representation of different ML technologies



performed this as an “as-a-service”. Because ML compilers that are specially made to adapt a given ML model to the target hardware platform must be weighed against the advantages of using specialized hardware for ML (Doyu et al. 2021). It also offers insufficient flexibility against the possibility of fast-changing hardware context because it necessitates re-compiling the ML inference model for the targeted device.

2.1 Advantages of TinyML

1. Latency: For time-critical applications, latency is a vital requirement. Typically, data processing on an embedded device results in less latency than data transfer to the cloud. Examples of applications where delay is not acceptable and data processing on the device is required include user experience apps and driver-assisting systems (Banbury et al. 2021).
2. Reduction of data transmission: When network access is unreliable or not available, it is necessary to process data at the edge device. Streaming large amounts of raw data from sensors like cameras, microphones, and accelerometers is often not practical.
3. Power effectiveness: MCUs may run for a long period on a tiny amount of power since they are built to do exceptionally low-power computations. Microcontroller units (MCUs) can be powered by batteries or other energy sources, such as energy harvesting.

2.2 Current Challenges and Solutions

1. Finding practical models for the edge is a difficult challenge: Based on the factors that impact the size of the model, how quickly it can make predictions, and how much power it requires for the application, TinyML systems must choose the target MCU hardware for their projects (Banbury et al. 2021). To focus on model architectures that are pertinent to the system design decisions, constrained neural architectural search (NAS) can be performed (Fedorov et al. 2019; Paissan et al. 2021).
2. There may be very little data in the given datasets. In addition to identifying strategies to test corner situations based on system specifications, test-driven development may be utilized to verify that test data is not used during training (Lakshman and Eisty 2022; Giray 2021).
3. Maintaining model development tools for several target hardware platforms is challenging. So, iterate more quickly by appropriately establishing the TinyML TFLM framework architecture from the beginning (David et al. 2021). The strain brought on by duplicate computations can be lessened by using event-based techniques to visual information processing (Sironi 2021).
4. The repeatability of a hyperparameter search process might be problematic. It is challenging to train TinyML OpenCV models from a performance-driven standpoint. So, it will be beneficial to include verification-based counter-examples to the training process (Lomuscio 2021). And platforms like EdgeImpulse and Qeexo provide AutoML features which are already offered via the web and implemented directly into edge CV applications.
5. While coming to model evaluation and compilation to handle environmental abnormalities or noise, TinyML models must be resilient. Before an application is installed on the MCUs, performance cannot be evaluated. MLCommons has released the MLPerfTiny benchmark, the first industry-standard benchmark suite for extremely low-power machine learning systems (Ramachandra et al. 2022). Since MCUs are primarily single-threaded, compilers must be able to provide access to performance metrics for them.
6. Due to computing limitations, it is challenging to debug compiled model's performance on the edge device. TinyML development requires the usage of emulators and simulators for MCUs. Device simulators can be used to compare how well the built TinyML models perform vs. raw models (Doria and Kadlec 2018).
7. If we succeed in model validation in a training environment, TinyML deployment in the real world remains bit challenging part. Deploying an embedded program with flaws is difficult. So, one must continuously evaluate and validate the model (Viswanatha et al. 2019; Viswanatha and Reddy 2017; Viswanatha et al. 2020; Sudharsan et al. 2021; Wong et al. 2020).

2.3 *Some Algorithms and Framework at Edge*

There are three types of frameworks, and associated algorithms are depicted in the following subsections.

2.3.1 FOMO

The new deep learning architecture, Faster Objects, More Objects (FOMO), has the potential to open up new computer vision applications. Small CPUs cannot handle the memory and processing demands of the majority of object detection deep learning models. FOMO is a technique that only requires a small amount of memory, making it well-suited for use in TinyML. TinyML is a subfield of machine learning that focuses on running ML models on devices with limited memory, such as microcontrollers. In contrast to other techniques that may require more memory, FOMO can be used effectively on devices with only a few hundred kilobytes of memory.

2.3.2 TFLite Framework

When only a few tens of kilobytes of RAM is available, embedded devices can execute the TensorFlowLite framework.

- The design of on-device machine learning focused on addressing five main constraints: reducing model and binary size, minimizing battery consumption, protecting privacy by ensuring that no personal data leaves the device thereby increasing the security and improving latency by eliminating the need for round-trip communication with a server, resulting in efficient inference and a lack of network connections (Sanchez-Iborra and Skarmeta 2020).
- The technology supports a range of platforms, such as MCUs, embedded Linux, and Android and iOS smartphones.
- The technology offers support for several programming languages, such as C, C++, Python, and Java.
- High performance is provided by the technology through model optimization and code and hardware optimization.
- Examples of complete solutions for a variety of platforms for well-known machine learning issues are text classification, object recognition, posture prediction, knowledge discovery, and picture categorization (Sanchez-Iborra and Skarmeta 2020).

2.3.3 MobileNetV2

- MobileNetV2 is made up of a convolutional neural network with 53 layers.
- It can categorize photographs into over a thousand distinct categories after being trained on the ImageNet database, which has over a million images.
- The network in its pretrained state has mastered the recognition of a wide variety of items, including monitor, keyboards, mouse, pen, pencils, and animals.
- Images having a resolution of 224 by 224 pixels are supported by the network.
- The calculation is optimized by using Eq. (1) and choosing the computing order that reduces the total number of tensors that must be retained in memory, hence lowering the amount of memory needed.

$$M(G) = \min_{\pi \in \sum(G)} \max_{i \in 1..n} \left[\sum_{i,\pi,G}^{\cdot} |A| \right]^{\cdot} + \text{size}(\pi i) \ldots \tag{1}$$

- The collection of intermediate tensors that are linked to any of nodes I through n is represented by $R(i, \pi, \mathrm{G})$.
- Size (i) is the entire amount of memory required for internal storage during operation *i,* and |*A*| stands for the size of the tensor A (Sandler et al. 2018).

3 Methodology

Initially, collect the data and extract the features using algorithms like CNN, and we can use frameworks like MobileNetsV2 to reduce the computational cost. Then identify the model and train it. After getting the results, evaluate and validate the model using validation and test set, and then finally convert the model to TensorFlowLite and deploy it on edge devices.

Figure 2 shows the workflow of deployment of TinyML on MCUs

In this paper, the proposed example shows the complete procedure of how TinyML works edge devices and the implementation of generic applications wherein ML, IoT, and computer visions are incorporated for better efficiency like in object detection, gesture detection, voice recognition, etc.

3.1 Object Detection

In this paper, we made a comparison between YOLO families where YOLOv7 tiny found to be the best algorithm for object detection and as shown in Fig. 3.

YOLOv7-tiny and YOLOv7-W6 are among the several basic YOLOv7 models. The foundational model that is most suited for standard GPU computation is

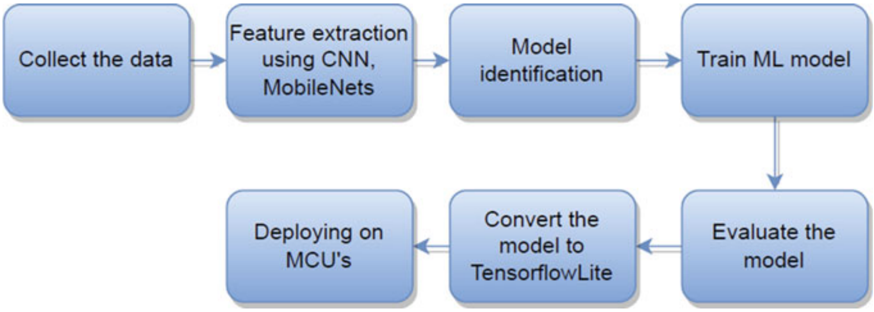


Fig. 2 TinyML workflow

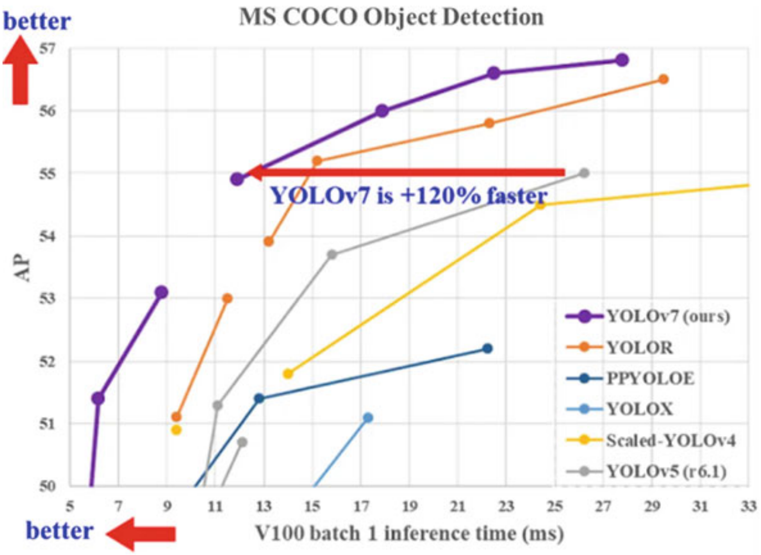


Fig. 3 Comparison of YOLO families (Wang et al. 2023)

YOLOv7. A simple model optimized for edge GPU is called YOLOv7-tiny. Computer vision models with TFLite are lighter to execute ML on mobile computing devices or distributed edge servers and devices. They are also designed for Edge AI and deep learning workloads. For distributed real-world computer vision applications, this approach is crucial. The edge-optimized YOLOv7-tiny differs from the previous versions in that it employs Leaky ReLU as the activation function as opposed to SiLU in the other models.

3.2 Person and Animal Detection

To implement the machine learning model on MCU, the model which is developed to be deployed and can detect person or animal like cow, monkey, elephant, etc. as these animals can harm the field of farmer, the proposed system can notify if animal or person is detected near farmland. To achieve this by using many machine learning models, here YOLO and CNN algorithms are used for the purpose mentioned above**.** The complete implementation process of YOLO and CNN is depicted in Figs. 4 and 5, respectively.

3.2.1 Using YOLO Algorithm

Steps:

1. Create a custom dataset.

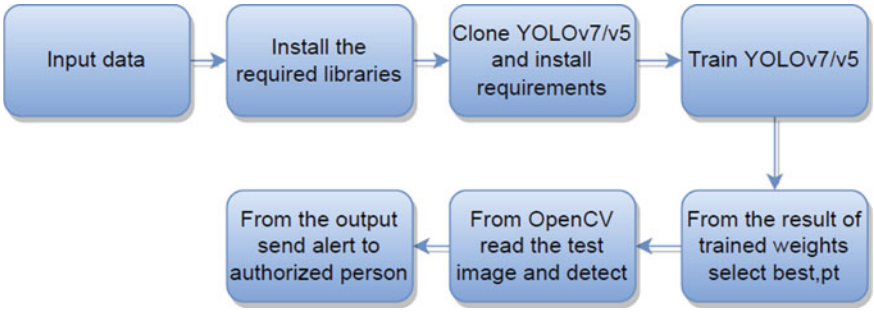

Input data
Install the required libraries
Clone YOLOv7/v5 and install requirements
Train YOLOv7/v5
From the result of trained weights select best,pt
From OpenCV read the test image and detect
From the output send alert to authorized person

Fig. 4 Steps involved in model for person/animal detection

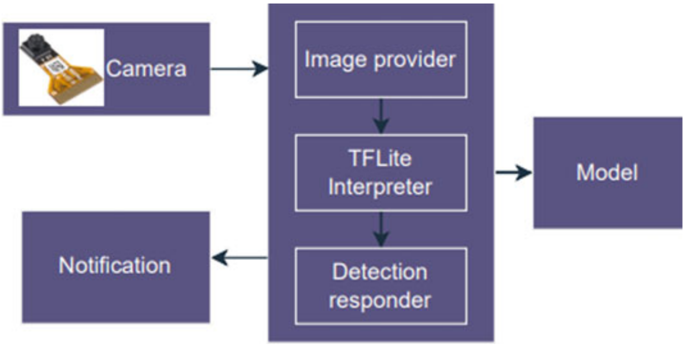


Fig. 5 Architecture of person detection (Viswanatha and Reddy 2017)

2. YOLOv7-tiny may be trained to recognize just people/animal using a custom dataset and the DarkNet.
3. YOLOv4-tiny may be trained to recognize just people using a custom dataset and the DarkNet.
4. Convert the trained DarkNet model to TFLite model.
5. Deploy the model on MCUs like SparkFun Edge, Arduino Nano BLE 33 Sense, etc.
6. From BLE connected to the cloud, get notifications if animal/person is detected.

3.2.2 Using CNN

The camera module will detect the object, and the output will be sent to main loop.

Image provider captures the image data from the Himax camera.

Then TFLite interpreter runs the model.

Then the machine learning model will run trained to detect person/ animal.

Detection responder takes the further action.

Notifications will be sent to authorized person if person/animal is detected.

Convolutional neural networks are used to train person detection model using the Visual Wake Words dataset. This collection of 115,000 photographs has been categorized with the presence or absence of people.

The model is considerably bigger around 250 KB. This increased size means that running a single inference will take much longer in addition to using significantly more memory.

Grayscale photographs of 96 by 96 pixels can be entered into the model. Each picture is offered as a 3D tensor of shape (96, 96, 1), with the last dimension containing an 8-bit value that denotes a single pixel. The value, which ranges from 0 (completely black) to 255, describes the color of the pixel (fully white).

3.3 Hardware Specifications

- The SparkFun Edge Development Board is shown in Fig. 6 along with a Himax CMOS camera that was created in collaboration with Google and Ambiq.
- This cutting-edge board is perfect for learning about speech and gesture detection without having to depend on other services.
- The Apollo3 Blue microcontroller from Ambiq Micro, which is well-known for its energy-efficient ARM Cortex-M4F 48 MHz CPU, powers the board.
- This CPU is specifically made to run TensorFlowLite and requires only 6uA/ MHz, making it an excellent choice for the job.
- Due to its low power consumption, the SparkFun Edge Board can function solely on a CR2032 coin cell battery for up to 10 days.

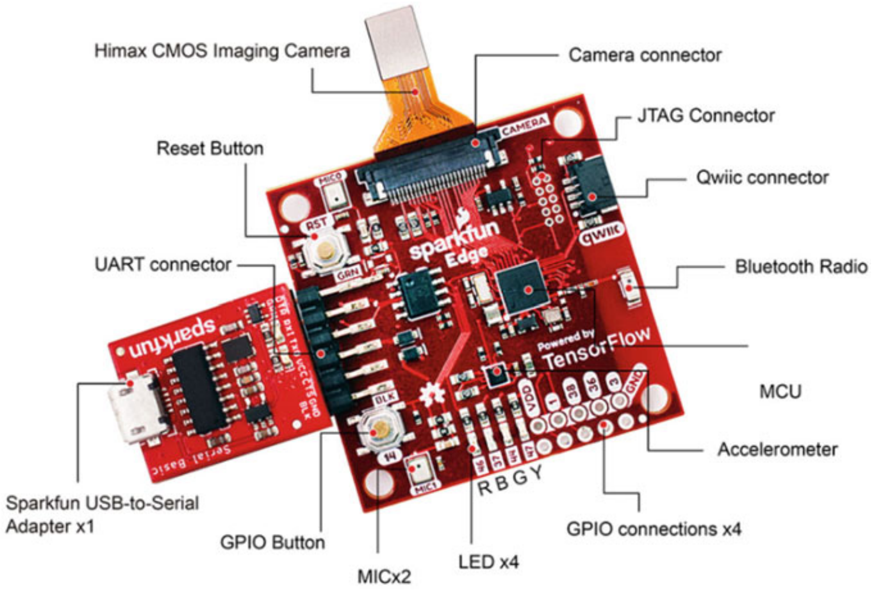


Fig. 6 SparkFun Edge and its parts

4 Results and Discussion

The results obtained from the model which is deployed on SparkFun Edge gave the good results and are shown below. Firstly, the code to detect the person and animals was ran on Google Colab using YOLOv5, then the same saved model is converted into TensorFlowLite model. Then the TFLite model is deployed on SparkFun Edge Board. Using BLE, we can connect the board to server and send the email notifications to authorized person if an animal or a person is detected. So, we can provide security to the farmland.

The model ran on PC, and the model deployed on SparkFun Edge **is** shown below in Fig. 7.

Figure 8 represents the person detection and its score.

If we face the SparkFun's Himax camera to persons face or body, the person score increases and no person score decreases.

The animal detection which uses YOLOv5 is as shown in Figs. 9 and 10.

The response thrown by the tool while running important TinyML libraries and functions on Colab platform for animal detection is as follows.

- Cloning, installing, and checking YOLOv5

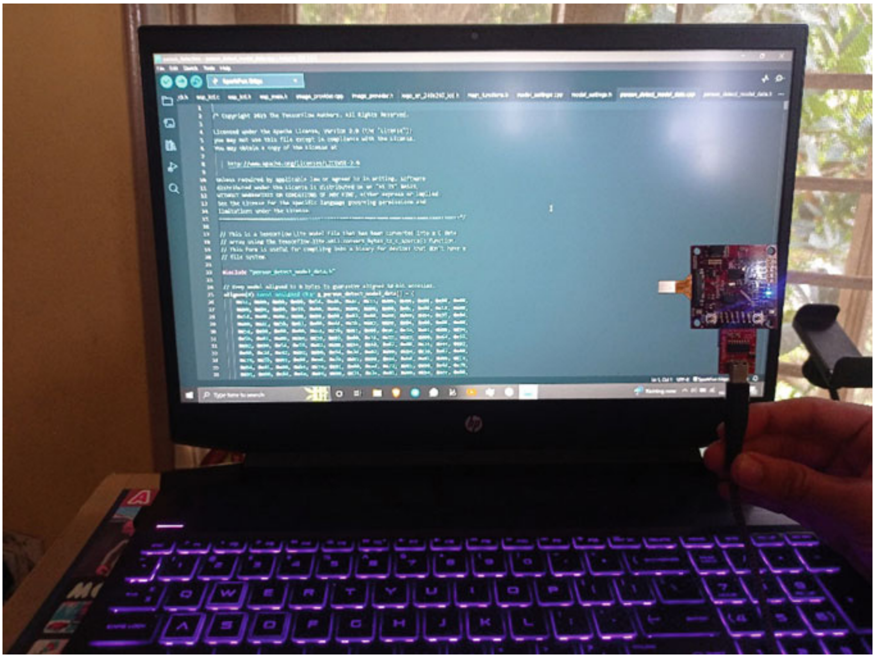
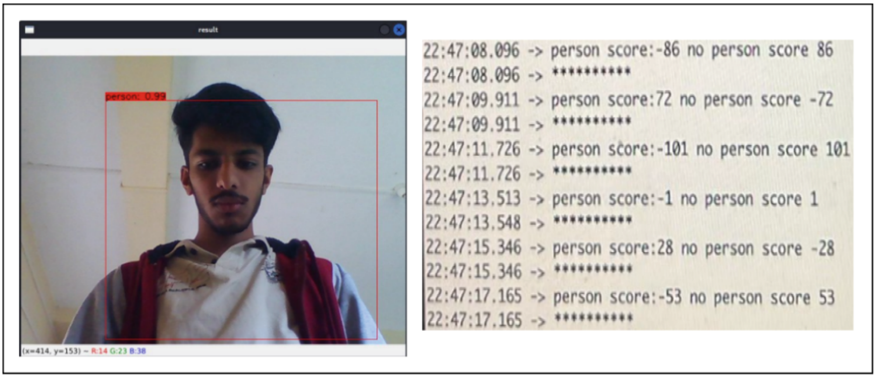


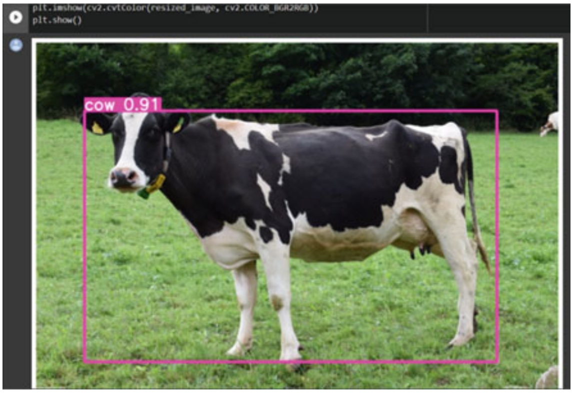
Fig. 7 SparkFun Edge Board with Himax CMOS imaging camera



```
22:47:08.096 -> person score:-86 no person score 86
22:47:08.096 -> **********
22:47:09.911 -> person score:72 no person score -72
22:47:09.911 -> **********
22:47:11.726 -> person score:-101 no person score 101
22:47:11.726 -> **********
22:47:13.513 -> person score:-1 no person score 1
22:47:13.548 -> **********
22:47:15.346 -> person score:28 no person score -28
22:47:15.346 -> **********
22:47:17.165 -> person score:-53 no person score 53
22:47:17.165 -> **********
```

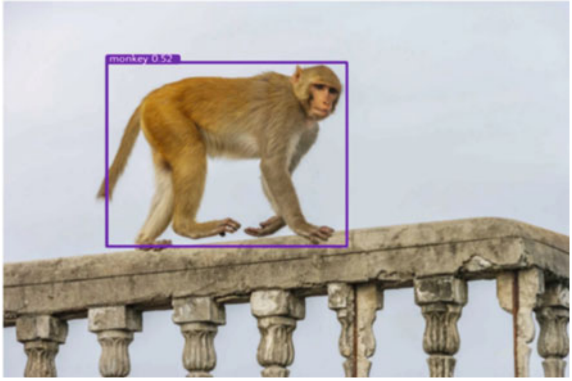
Fig. 8 Person detection results

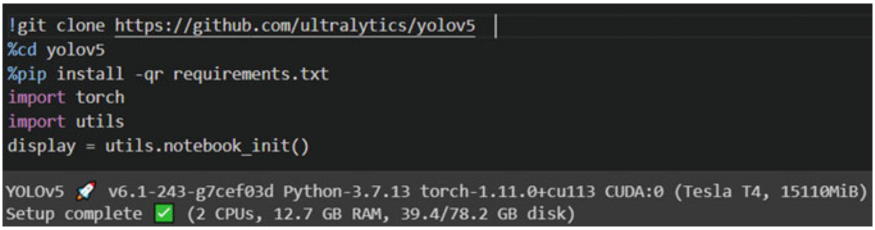
Fig. 9 Cow detection



```
plt.imshow(cv2.cvtColor(resized_image, cv2.COLOR_BGR2RGB))
plt.show()
```

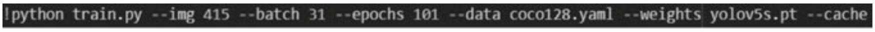
Fig. 10 Monkey detection





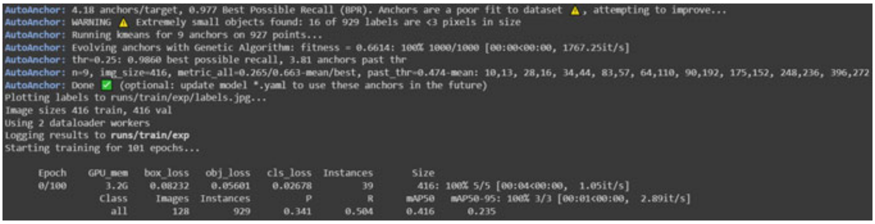
```
!git clone https://github.com/ultralytics/yolov5
%cd yolov5
%pip install -qr requirements.txt
import torch
import utils
display = utils.notebook_init()

YOLOv5 🚀 v6.1-243-g7cef03d Python-3.7.13 torch-1.11.0+cu113 CUDA:0 (Tesla T4, 15110MiB)
Setup complete ✅ (2 CPUs, 12.7 GB RAM, 39.4/78.2 GB disk)
```

- Training YOLOv5



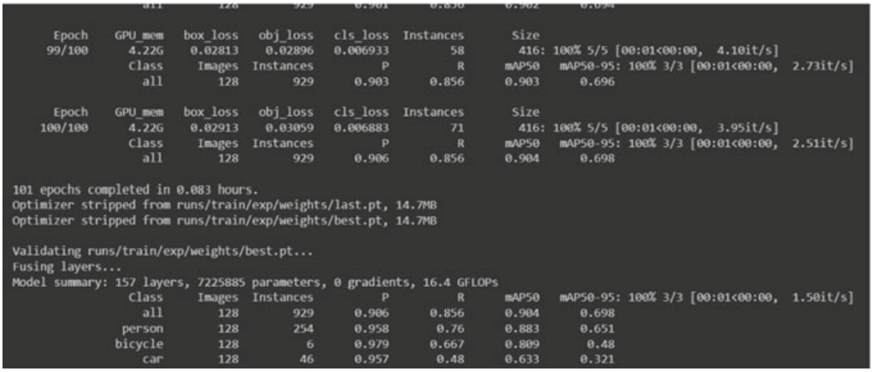
```
!python train.py --img 415 --batch 31 --epochs 101 --data coco128.yaml --weights yolov5s.pt --cache
```

- After running for 101 epochs, we got the result stored in runs/train/exp directory



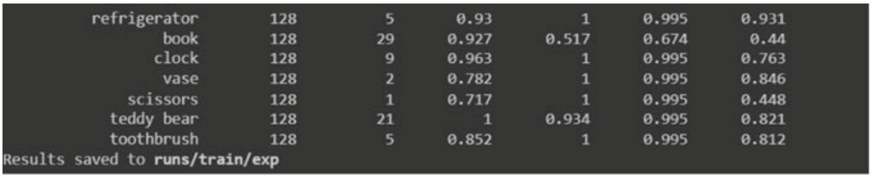
```
AutoAnchor: 4.18 anchors/target, 0.977 Best Possible Recall (BPR). Anchors are a poor fit to dataset ⚠, attempting to improve...
AutoAnchor: WARNING ⚠ Extremely small objects found: 16 of 929 labels are <3 pixels in size
AutoAnchor: Running kmeans for 9 anchors on 927 points...
AutoAnchor: Evolving anchors with Genetic Algorithm: fitness = 0.6614: 100% 1000/1000 [00:00<00:00, 1767.25it/s]
AutoAnchor: thr=0.25: 0.9860 best possible recall, 3.81 anchors past thr
AutoAnchor: n=9, img_size=416, metric_all=0.265/0.663-mean/best, past_thr=0.474-mean: 10,13, 28,16, 34,44, 83,57, 64,110, 90,192, 175,152, 248,236, 396,272
AutoAnchor: Done ✅ (optional: update model *.yaml to use these anchors in the future)
Plotting labels to runs/train/exp/labels.jpg...
Image sizes 416 train, 416 val
Using 2 dataloader workers
Logging results to runs/train/exp
Starting training for 101 epochs...

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      0/100      3.2G    0.08232    0.05601    0.02678         39        416: 100% 5/5 [00:04<00:00,  1.05it/s]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 3/3 [00:01<00:00,  2.89it/s]
                   all        128        929      0.341      0.504      0.416      0.235
```

……



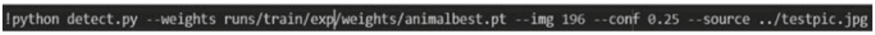
```
      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
     99/100     4.22G    0.02813    0.02896   0.006933         58        416: 100% 5/5 [00:01<00:00,  4.10it/s]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 3/3 [00:01<00:00,  2.73it/s]
                   all        128        929      0.903      0.856      0.903      0.696

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
    100/100     4.22G    0.02913    0.03059   0.006883         71        416: 100% 5/5 [00:01<00:00,  3.95it/s]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 3/3 [00:01<00:00,  2.51it/s]
                   all        128        929      0.906      0.856      0.904      0.698

101 epochs completed in 0.083 hours.
Optimizer stripped from runs/train/exp/weights/last.pt, 14.7MB
Optimizer stripped from runs/train/exp/weights/best.pt, 14.7MB

Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7225885 parameters, 0 gradients, 16.4 GFLOPs
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 3/3 [00:01<00:00,  1.50it/s]
                   all        128        929      0.906      0.856      0.904      0.698
                person        128        254      0.958       0.76      0.883      0.651
               bicycle        128          6      0.979      0.667      0.809       0.48
                   car        128         46      0.957       0.48      0.633      0.321
```

….

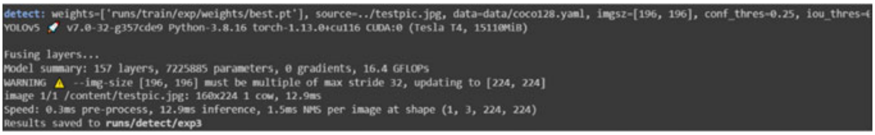


```
          refrigerator        128          5       0.93          1      0.995      0.931
                  book        128         29      0.927      0.517      0.674       0.44
                 clock        128          9      0.963          1      0.995      0.763
                  vase        128          2      0.782          1      0.995      0.846
              scissors        128          1      0.717          1      0.995      0.448
            teddy bear        128         21          1      0.934      0.995      0.821
            toothbrush        128          5      0.852          1      0.995      0.812
Results saved to runs/train/exp
```

From results obtained above, choose animalbest.pt from weights and check for testpic.jpg

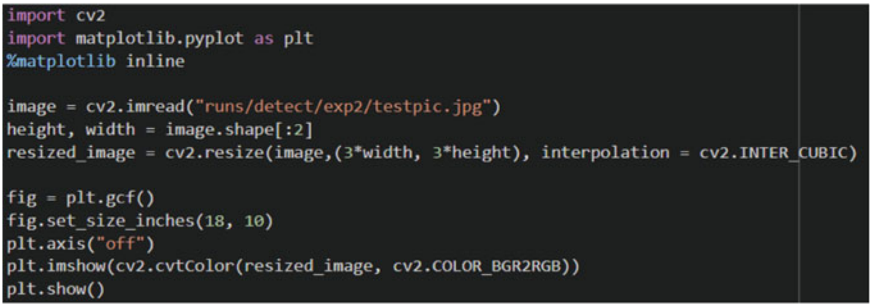


```
!python detect.py --weights runs/train/exp/weights/animalbest.pt --img 196 --conf 0.25 --source ../testpic.jpg
```



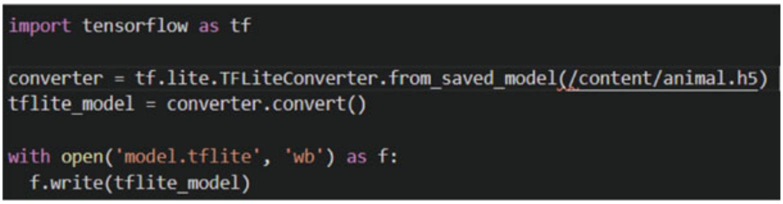
```
detect: weights=['runs/train/exp/weights/best.pt'], source=../testpic.jpg, data=data/coco128.yaml, imgsz=[196, 196], conf_thres=0.25, iou_thres=
YOLOv5 🚀 v7.0-32-g357cde9 Python-3.8.16 torch-1.13.0+cu116 CUDA:0 (Tesla T4, 15110MiB)

Fusing layers...
Model summary: 157 layers, 7225885 parameters, 0 gradients, 16.4 GFLOPs
WARNING ⚠ --img-size [196, 196] must be multiple of max stride 32, updating to [224, 224]
image 1/1 /content/testpic.jpg: 160x224 1 cow, 12.9ms
Speed: 0.3ms pre-process, 12.9ms inference, 1.5ms NMS per image at shape (1, 3, 224, 224)
Results saved to runs/detect/exp3
```

- From the result, it has got using OpenCV; read the test image, and predict it as cow with probability 0.76



```
import cv2
import matplotlib.pyplot as plt
%matplotlib inline

image = cv2.imread("runs/detect/exp2/testpic.jpg")
height, width = image.shape[:2]
resized_image = cv2.resize(image,(3*width, 3*height), interpolation = cv2.INTER_CUBIC)

fig = plt.gcf()
fig.set_size_inches(18, 10)
plt.axis("off")
plt.imshow(cv2.cvtColor(resized_image, cv2.COLOR_BGR2RGB))
plt.show()
```

- Then save the model and convert it to TensorFlowLite as shown



```
import tensorflow as tf

converter = tf.lite.TFLiteConverter.from_saved_model(/content/animal.h5)
tflite_model = converter.convert()

with open('model.tflite', 'wb') as f:
  f.write(tflite_model)
```

- Then finally deploy it on MCUs.

Memory consumption on MCU is also taken into consideration as its edge computing.

The SRAM contains the runtime variables created during NN execution. The selected boards only feature 384 KB to 1 MB of SRAM. This limits the execution and deployment of big models.

SRAM in MCUs is limited in size, which can affect the execution and deployment of large neural networks. The amount of SRAM available in a given MCU is determined by the compiler used to build the neural networks and IoT applications and can be limited by factors such as power leakage and production costs. Large neural networks that were trained on datasets with many features may require more SRAM and take longer to compile and flash onto target boards.

5 Conclusion

Machine learning analysis on low-power devices, TinyML, is receiving more and more interest in both academia and industry. The potential to use great decision at the network's edge is enormous for the MCUs ecosystem. Using machine learning in embedded systems with little resources in light of prospective uses, gadgets are becoming necessary. TinyML is a rapidly growing field that involves making decisions about how to balance various integral components, such as hardware, software, and machine learning and deep learning algorithms. It is an important area of development in the field of artificial intelligence. On well-known MCU board, demonstrate and examine the onboard functionality of neural network models. TinyML was developed as a result of the need for a scalable solution caused by the ingestion of traditional ML models and the requirements of edge devices.

The variety of classification methods, including SVM, MLP, decision trees, and RF, may be combined into an MCU board. The conclusion was RF and decision trees offer the highest accuracy, memory footprint, and classification speed.

In this study, we tested the responsiveness of TinyML model that was trained using information gathered by the SparkFun Edge and Himax CMOS sensors. We conducted tests on the ability to recognize person, animal, speech patterns, and noises. Results demonstrate that TinyML may be used for object detection, smart farming, defense, etc. opening the door for the creation of fresh, innovative, and long-lasting applications.

Future works: There are many potential developments in the field of TinyML in the future. One possibility is to conduct a multi-dimensional analysis of trade-offs between different factors, such as power consumption, energy management, inference rates, and memory management rates. This analysis could help to better understand the differences between tiny and non-tiny implementations of various machine learning algorithms. Also, in our work, we can detect wild animals by the sound they make as we can run speech recognition models on MCUs.

References

Banbury CR, Reddi VJ, Lam M, Fu W, Fazel A, Holleman J, Huang X, Hurtado R Kanter D, Lokhmotov A, Patterson D, Pau D, Sun Seo J, Sieracki J, Thakker U, Verhelst M, Yadav P (2021) Benchmarking TinyML systems: challenges and direction

David R, Duke J, Jain A, Reddi VJ, Jeffries N, Li J, Kreeger N, Nappier I, Natraj M, Regev S, Rhodes R, Wang T, Warden P (2021) Tensorflow lite micro: embedded machine learning on TinyML systems

Doria IE, Kadlec B (2018). CVPR18: tutorial: software engineering in computer vision systems. Comput Vis Found Jun 2018

Doyu H, Morabito R, BrachmannyM (2021) A TinyMLaaS ecosystem for machine learning inIoT: overview and research. Ericsson Research, Finland Ericsson Research, Sweden

Fedorov I, Adams RP, Mattina M, Whatmough PN (2019) Sparse: sparsearchitecture search for cnns on resource-constrained microcontrollers

Giray G (2021) A software engineering perspective on engineering machine learning systems: state of the art and challenges. J Syst Softw 180:111031

Lakshman SB, Eisty NU (2022) Boise software engineering approaches for TinyML based IoT embedded vision: a systematic literature review. Shashank Bangalore. State University Boise, USA

Lomuscio A (2021) TinyML talks: verification of ML-based AI systems and its applicability in edge ML. tinyML, Oct 2021

Paissan F, Ancilotto A, Farella E (2021) PhiNets: a scalable backbone for lowpowerai at the edge

Ramachandra AC, Viswanatha V, Kishor K, Suhas H, Pannir Selvam E (2022) In-cabin radar monitoring system: detection and localization of people inside vehicle using vital sign sensing algorithm. Int J Recent Innov Trends Comput Commun 10(8):104–109. https://doi.org/10.17762/ijritcc.v10i8.5682D

Sanchez-Iborra R, Skarmeta AF (2020) TinyML-enabled frugal smart objects: challenges and opportunities

Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C (2018) MobileNetV2: inverted residuals and linear bottlenecks. Google Inc.

Shafique M, Theocharides T, Reddy VJ, Murmann B (2021) TinyML: current progress, research challenges, and future roadmap. In: 2021 58th ACM/IEEE design automation conference (DAC). ACM/IEEE, San Francisco, US, pp 1303–1306

Sironi A (2021) TinyML talks: machine learning for event cameras. tinyML, Oct 2021

Sudharsan B, Salernoy S, Nguyen D-D, Yahya M, Wahid A, Yadav P, Breslin JG, Aliz MI (2021) TinyML benchmark: executing fully connected neural networks on commodity microcontrollers. Data Science Institute, NUI Galway, Ireland

TinyML Benchmark (2021) Executing fully connected neural networks on commodity microcontrollers. Data Science Institute, NUI Galway, Ireland Conference Paper May 2021

Viswanatha V, Reddy RVS (2017) Digital control of buck converter using arduino microcontroller for low power applications. In: 2017 international conference on smart technologies for smart nation (SmartTechCon). IEEE

Viswanatha V, Reddy VS, Rajeswari D (2019) Stability and dynamic response of analog and digital control loops of bidirectional buck-boost converter for renewable energy applications. Int J Recent Technol Eng (IJRTE). Available at SSRN: https://ssrn.com/abstract=3515804

Viswanatha V, Venkata Siva Reddy R, Rajeswari (2020) Research on state space modeling, stability analysis and PID/PIDN control of DC–DC converter for digital implementation. In: Sengodan T, Murugappan M, Misra S (eds) Advances in electrical and computer technologies. Lecture Notes in Electrical Engineering, vol 672. Springer, Singapore

Wang C-Y, Bochkovskiy A, Mark H-Y (2023) Liao YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. Institute of Information Science, Academia Sinica, Taiwan kinyiu@iis.sinica.edu.tw, alexeyab84@gmail.com, and liao@iis.sinica.edu.tw

Wong A, Famouri M, Shafiee MJ (2020) AttendNets: tiny deep image recognition neural networks for the edge via visual attention condensers. arXiv preprint arXiv:2009.14385

Social Distancing Monitoring for Real-Time Deep Learning Framework

Sunil S. Harakannanavar, R. Pramodhini, A. S. Sudarshan, Atish, Rohit Jadhav, and S. Kruthik

1 Introduction

An effective measure against the novel COVID-19 pandemic is social distancing. However, in a real sense for the public, it is not possible for them to estimate the distance mentally. An automated surveillance system can aid the augmentation of the individual's perspective capability. Employing a dynamic monitoring system brings in a lot of ethical concerns. The privacy of the individuals is at stake if data is recorded and stored. The monitoring system is expected to be instantaneous without any capacity for data storage. Additionally, the detector must not non-favor any object class during identification. This can be attained by developing a system based on deep learning. A crucial aspect over here is that the system is supposed to be non-intrusive meaning that the warning system cannot target the individuals directly. This can be done by displaying visual cue if the subjects in the frame violate COVID-19 norms. This is a vital for establishment between surveillance and society. After finding the gaps in the literature survey, the existing systems are found to be inefficient when it comes to real-time implementation. The deep learning model for detecting humans in the frame and creating bounding boxes around them to determine the distance is proposed. Further, detections are represented using spatial coordinates. If the measured distance between two or three people is less than the assigned value, the system will display a visual cue. In the previous outbreak, a lot of families lost their loved ones and until vaccines prove to be completely efficient, maintaining social distance is an unswerving solution to the pandemic. Any form of social gathering like weddings, parties, or family events was restricted for the public and following the COVID-19 protocol everywhere was not feasible for people, especially for senior citizens who were more susceptible to the infection. People went outside carrying

S. S. Harakannanavar (✉) · R. Pramodhini · A. S. Sudarshan · Atish · R. Jadhav · S. Kruthik
Department of Electronics and Communication Engineering, Nitte Meenakshi Institute of Technology, Yelahanka, Bangalore, Karnataka 560064, India
e-mail: sunilsh143@gmail.com

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_5

a fear inside them of the invisible virus which took a toll on their mental health. According to the WHO, maintaining social distance is proven to be 98% efficient and they insist on following the norms even after pandemic. Hence, what people lacked was a surveillance system that would detect whether people are following the measures or not and thereby giving them a sense of security.

The paper is organized as follows. Section 2 describes the descriptions of exist sting model. The proposed model is discussed in Sect. 3. The result analysis of the proposed model is done in Sect. 4. The conclusion and future research are addressed in Sect. 5.

2 Related Work

In this section, the work related to existing methodologies used for Social Distancing Monitoring Framework Machine learning algorithms are discussed. Yadav et al. (2022) developed a model to monitor and manage the crowd for social distance. The YOLO and deep sort techniques are used to evaluate the model. Ahmed et al. (2021) discussed a platform related to deep learning, the YOLO was used for object recognition. Mangshor et al. (2021) introduced the YOLO algorithm which was implemented with the SDLC to recognize an object. The deep learning model was developed by Gopal and Ganesan (2022) to measure the distance between the crowds. Here, ED is used for classification process. The model for COVID-19 patients using cough information is discussed by Pahar et al. (2022). The PCA and ML algorithms are used for object recognition. The Fast-RCNN was implemented by Ren et al. (2015) to make the process fast and to run the model. The small objects were recognized by Liu et al. (2016) with the contribution of YOLO algorithm. The SSD algorithm was implemented by Ganiger et al. (2017) to have a control on the crowd management system. Kahale (2020) described the measures in social distancing and provided the information which can reduce economically. The tracking of person whether an individual is affected by COVID-19 is determined by the model proposed by Georgievski (2020). The tracking and object detection algorithms are proposed by Georgievski (Harakannanavar et al. 2019a). The various deep learning models and their impacts were addressed in Wang et al. (Harakannanavar et al. 2018a) model. The model for detection of an individual is carried with video frames and Gaussian mixture for deep learning approach. The CNN approach in order to detect the human is carried out by Brunetti et al. (Harakannanavar et al. 2019b). The control of static approach was discussed by the Manfredi et al. (Harakannanavar et al. 2020a). Here, the text features are extracted and the classification was carried out using SVM. Huang et al. (Harakannanavar et al. 2018b) cameras were installed to monitor and find out the individual in the crowd management system. The significant information with respect to D detectors was described in the model suggested by Zou (Harakannanavar et al. 2018c). The social distance monitoring and controlling were made by using the networks suggested by Landing et al. (Harakannanavar et al. 2019c). The NN which uses synthetic data was suggested by Deng et al. (Harakannanavar et al.

2019d) to propose stereo images. The fused 3D point clouds are described in the suggested model by Zhang et al. (Harakannanavar et al. 2019e) to obtain the outdoor scenes.

2.1 Open Issues and Challenges (Wang et al. 2017)

- One of the primary drawbacks of the existing system is that it is not user-friendly because the data retrieval is very slow, and data is not maintained efficiently (Brunetti et al. 2018).
- The existing systems are not completely automated meaning the system needs to be monitored (Manfredi and Calderara 2014).

The systems raise a lot of privacy concerns that have been neglected until now. The system needs to be real-time. It fails to give high-end detail in surveillance that can be neglected through the naked eye (Huang et al. 2010).

2.2 Problem Definition

A surveillance system for monitoring the social distancing norms is needed to fight against COVID-19. To propose the proper solution for the above problem, a system that implements object detection on real-time video and monitors social distancing between human beings using a deep learning framework is required to develop. The model is explained in Sect. 3.

3 Proposed Methodology

The proposed system is validated using PyCharm (IDE) which integrates the laptop camera collecting real-time input. Initially, the video is captured frame-by-frame and resized to 300 × 300 pixels to meet the object detection model requirements. This is done with the help of the OpenCV module. SSD Mobile Net CAFFE model is a pre-trained deep learning model, used for object detection. Here, the resized frames are fed to the system. The model has been pre-trained for 20 labels such as bottle, bicycle, chair, monitor. For the model to detect specifically humans, conditional statements are used that filter only the human class. The class object humans are assigned a class id of 15. The humans are detected by drawing a bounding box around them. Then, calculate the mid-point by determining x and y coordinates of the bounding boxes. The mid-point helps us to find the distance between two bounding boxes or people. The model then enters a conditional loop, which determines whether the subjects (humans) are maintaining 2 m or not. If they are maintaining 2 m or more, the color

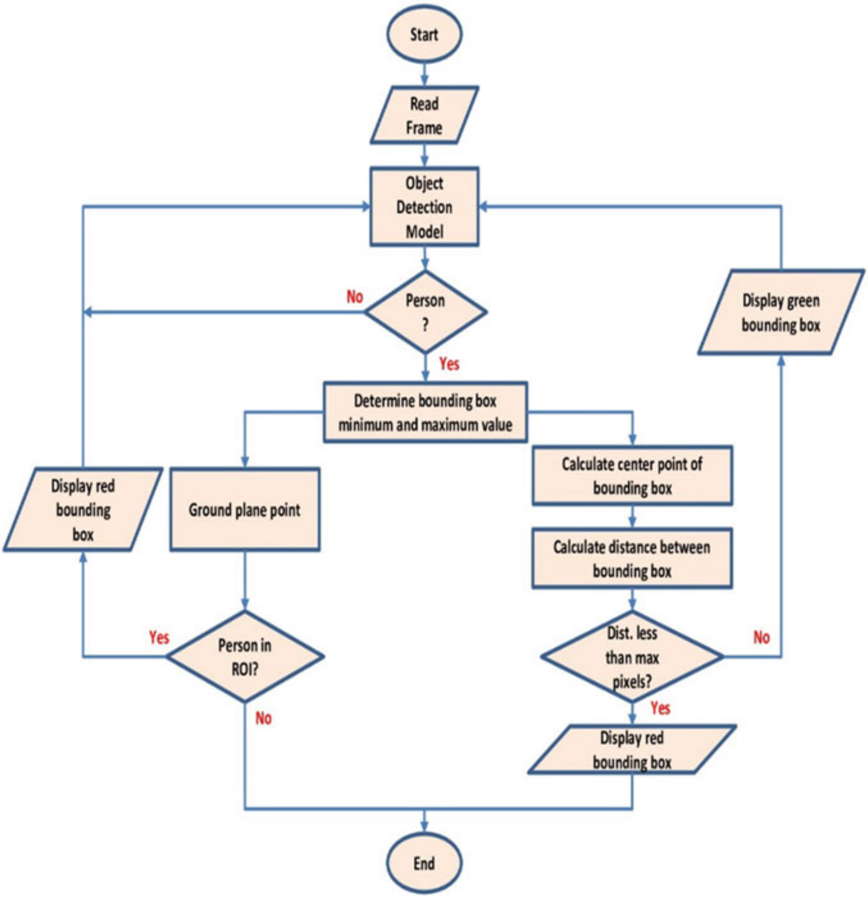

Start
Read Frame
Object Detection Model
Person ?
No
Yes
Determine bounding box minimum and maximum value
Display green bounding box
Display red bounding box
Ground plane point
Calculate center point of bounding box
Calculate distance between bounding box
Yes
Person in ROI?
No
Dist. less than max pixels?
No
Yes
Display red bounding box
End

Fig. 1 Proposed model

of the bounding box appears green, and the non-alarming visual cue signifies safety and social distancing norms are met. If the subjects are maintaining less than 2 m, the color of the bounding box appears red, and the non-alarming visual cue indicates threat of exposure toward COVID-19 and is expected to maintain social distance. The proposed model is shown in Fig. 1.

3.1 Dataset Description

The proposed deep learning model has been pre-trained and tested on MS COCO dataset (Zou et al. 2019). Microsoft Common Objects in Context is widely known for its 328,000 organized and labeled images. It contains images of other things or stuff such as chair, grass, table, bicycle, airplane, street, background. More the number of images in the dataset, more accurately the deep learning model will learn to identify and distinguish between objects (human being). It is open source and suitable for implementation in machine learning projects (Landing 2020).

3.2 Finding the Mid-point of the Co-ordinates

Once the bounding box is created around a person in the frame, the system computes the four coordinates. Using the mid-point formula, then find the center point (Zhang and Vetter 2015). This center point will now act as a reference for other bounding boxes in the frame. The mid-point formula is $C\ (x, y) = (x_{\text{min}} + x_{\text{max}}/2, y_{\text{min}} + y_{\text{max}}/2)$.

3.3 Finding the Distance Between Two Bounding Boxes

$C_1(x_{\text{min}}, y_{\text{min}})$ and $C_2(x_{\text{max}}, y_{\text{max}})$ are the four coordinates that were determined in the previous process. The distance between each of the detected people in the frame is measured by evaluating the distance between their respective mid-points (Harakannanavar and Puranikmath 2017). The distance is calculated using Eq. 1.

$$d(C_1, C_2) = \sqrt{(x_{\text{max}} - x_{\text{min}})^2 + (y_{\text{max}} - y_{\text{min}})^2}. \tag{1}$$

3.4 Modules

Modules can be defined as units of the project that work in tandem to ensure the smooth running of the system. The three key modules are as follows (Harakannanavar et al. 2019f):

- Input module.
- Human detection module.
- Recognition module.

3.5 Input Module

Here, two options are provided in view of input for the proposed system. Own laptop recorded with real time video. In case of option 2, using a pre-recorded video file for

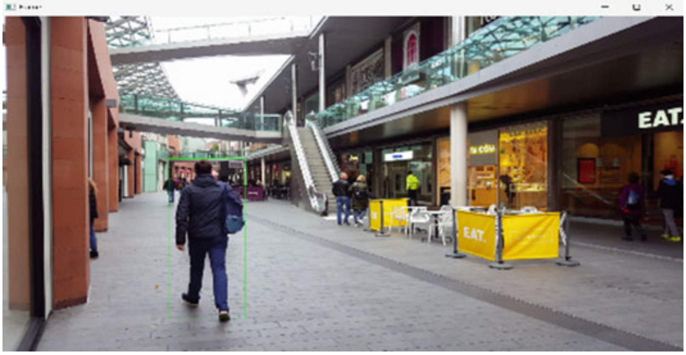


Fig. 2 Input module example 1

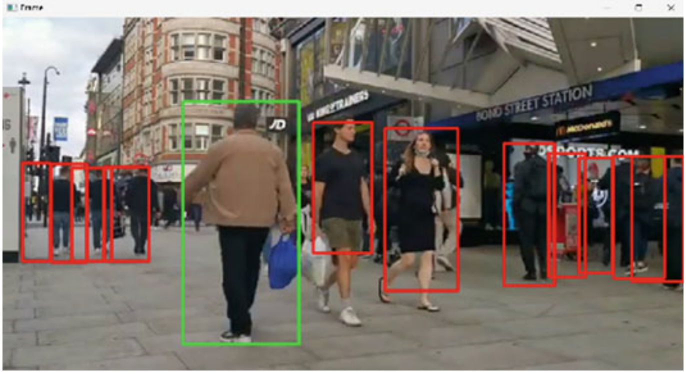


Fig. 3 Human detection module

which the file path will be given as input in the code and process further to produce the desired output. The input is collected and calibrated frame-by-frame using feature extraction. Now, import the OpenCV module from the Python library into virtual environment. It is responsible for image processing and development of real-time computer vision applications (Harakannanavar et al. 2019g). The input module and human detection module are shown in Figs. 2, 3, and 4, respectively.

3.6 Human Detection Module

The SSD Mobile Net V2 is an object detection module that was considered in the proposed model. It is pre-trained on the Common Objects in Context dataset which consists of 164K images (Harakannanavar et al. 2020b). Once the real-time video starts running, the model captures multiple frames successively and detects human

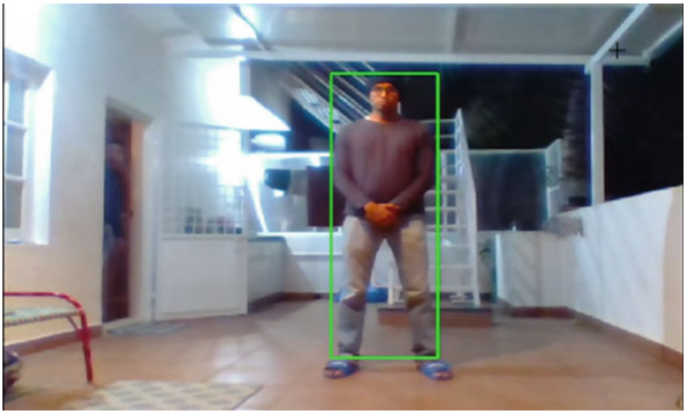


Fig. 4 Result Example 1

according to the assigned class object. Here, open-source software library is used to improve on the object class detection.

For training the deep learning module, the Common Objects in Context dataset also known as MS COCO (Harakannanavar et al. 2022a) is used for obtaining the results. Next, assign a class id to objects in the frame that categorize into humans. Although the model can identify a vast number of objects, that were focusing only identify the person in the frame with accuracy (Jayalaxmi et al. 2022). However, due to camera limitations, the accuracy reduces as the subjects move far from the camera (Kajabad 2019). The proposed model is tested for other object classes such as cat, dog, motorcycle, pots, TV, sofa, and monitor (Kreiss 2020).

4 Result Analysis

Following are the software requirements used PyCharm and Python. The hardware requirements are Laptop Processor (minimum) (Harakannanavar et al. 2022b): Pentium 4 (2.4 GHz), Hard Disk space (minimum): 250 GB, Random Access Memory (minimum): 1 GB, and camera and resolution: HD webcam (1280 × 720p) which are used to conduct the experiments. The video snippets below show the results of the human detection module. In this work, the video is recorded in an open space with dimensions 5.5 m depth × 3 m width at a fixed angle to replicate a surveillance system. Own laptop camera (Harakannanavar et al. 2022b) is used to capture the images for the implementation of the module, and the perspective view of the video frame is transformed to a bench view approximately 0.5 m above the ground. Figure 4 shows successful human detection along with a bounding box. The detection is a complete top-down detection, and since there is only a single person in the frame, the box will appear in green.

Next, another person in the frame to test the modules accuracy is considered. The model primarily focuses on measuring the distance between two or more people and emitting a non-alarming visual cue. Here, red indicates that the subjects in the frame are not maintaining social distancing norms. As mentioned, the permissible distance between two people is set to be 2 m, and the bounding boxes will appear green if the distance is greater than the threshold. In Fig. 5, the above-discussed framework is successfully employed. The neighboring bounding boxes appear in red color.

Next, the subjects in the frame maintain social distancing measures and the system being real-time tracks their movement and calculates the distance. The BB immediately changes to green as soon as the subjects in the frame move beyond threshold. A model is developed that would be applicable in social situations and tested the module for three people (Fig. 6).

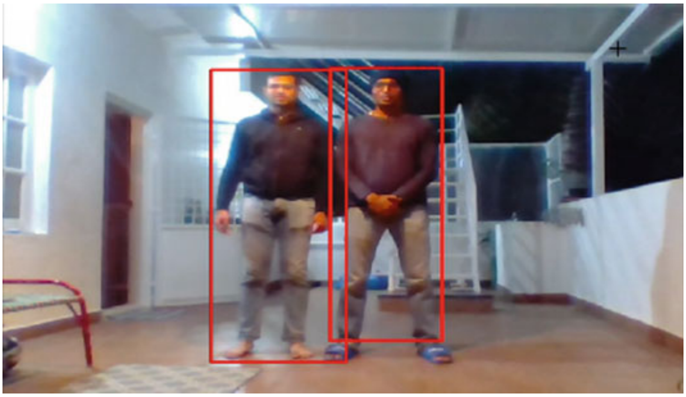


Fig. 5 Result Example 2

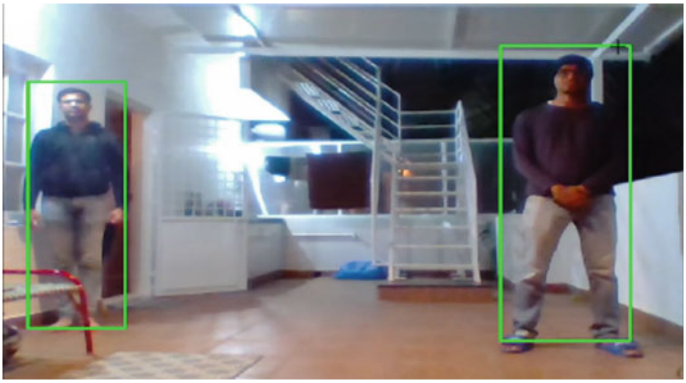


Fig. 6 Result Example 3

In Fig. 7, third team member is positioned between the two. Since the three members are not following the protocol in respect to each other, all boxes over here appear red.

Next step is to make everyone in the frame to be positioned at 2 m distance from each other. The three bounding boxes appear green and is shown in Fig. 8.

Now, another permutation of positioning the three subjects in the frame to imitate a real-world scenario where there are constant movements is considered for the next event. The model works for all sorts of positioning. Figure 9 shows two members who are not following the social distancing measures in respect to a third person who is standing at the far end to the right. The visual cue alters according to the scenario indicating a real-time efficiency of the model.

Another variation in the positioning is carried out further and is shown in Fig. 10. It is observed that fourth member entered the frame who was maintaining distance in respect to all. The person on the right is following social distancing too.

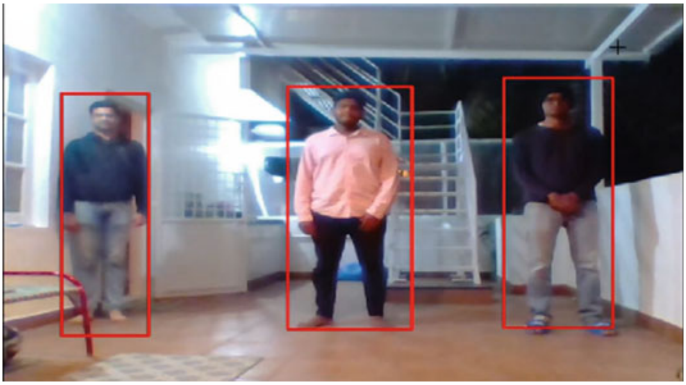


Fig. 7 Result Example 4

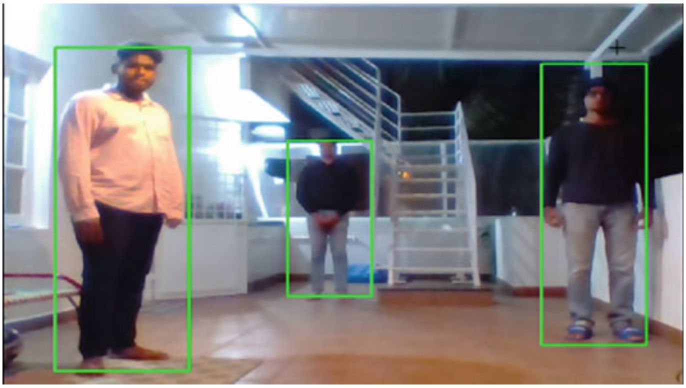


Fig. 8 Result Example 5

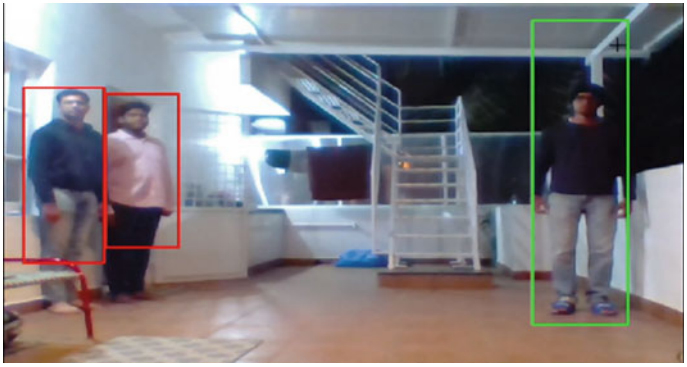


Fig. 9 Result Example 6

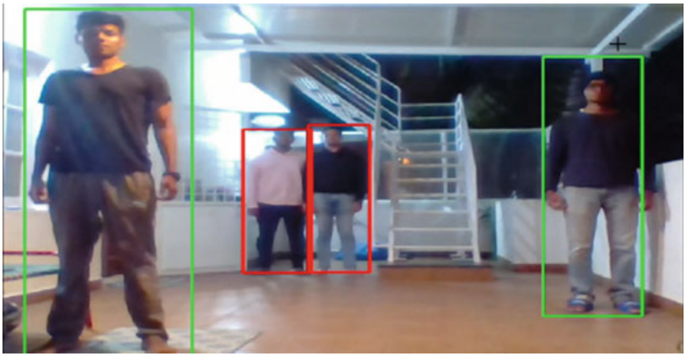


Fig. 10 Result Example 7

For verifying the efficiency of the code, rest of the two members were closely positioned behind. The model accurately calculates the depth and lateral distance between the all the members.

5 Conclusion and Future Scope

Social detection and distancing monitoring using a DL model are implemented by taking the frames of an input video or from a real-time video recording. The DL model is deployed to detect humans and create a bounding box around them. The spatial coordinates and the mid-point of the bounding box are calculated. The proposed model using deep learning is tested on a video portraying pedestrian on a street to identify, and the model can be used as a solid foundation for the development of projects in the field of study such as, study of behavioral patterns of people, predicting crowd or traffic movement. The model proved with an accuracy of 98.6%

on real-time dataset. In future, the model can be modified and customized for use in other environments such as shopping malls, cinema theaters, offices, restaurants, and schools.

References

Ahmed I, Ahmad M, Rodrigues JJ, Jeon G, Din S (2021) A deep learning-based social distance monitoring framework for COVID 19. Int J Sustain Cities Soc 65:1–12

Brunetti A, Buongiorno D, Trotta GF, Bevilacqua V (2018) Neuro computing computer vision and deep learning techniques for pedestrian detection and tracking: a survey. J Neurocomputing 300:17–33

Ganiger D, Patil KA, Patil P, Anandhalli M (2017) Automatic control of power supply in classroom using communication technology. In: IEEE ınternaional conderence of applications of electronics and communication technologies, pp 230–234

Georgievski (2020) Tracking COVID-19. Biomedical Eng Community 1–4

Gopal B, Ganesan A (2022) Real time deep learning framework to monitor social distancing using improved single shot detector based on overhead position. Earth Science Inform 585–602

Harakannanavar SS, Jayalaxmi H, Asha CN, Prashanth K, Hudedavar P (2022a) Biometric trait: offline signature ıdentification and verification based on multimodal fusion techniques. J Posit Sch Psychol 6(4):2180–2191

Harakannanavar SS, Ramachandra AC, Pramodhini R, Surekha M, Puranikmath VI, Prashanth CR (2022b) Performance evaluation of feature level fusion for multimodal biometric systems. Math Stat Eng Appl 71(4):2775–2792

Harakannanavar SS, Asharani TS (2018) Design of an efficient algorithm for Iris recognition. In: IEEE international conference on inventive communication and computational technologies, pp 876–880. https://doi.org/10.1109/ICICCT.2018.8473179

Harakannanavar SS, Puranikmath VI (2017) Comparative survey of ıris recognition. In: IEEE ınternational conference on electrical, electronics, communication, computer, and optimization techniques, pp 280–283

Harakannanavar SS, Prashanth CR, Raja KB, Chetan M (2018a) Face recognition based on the fusion of bit-plane and binary ımage compression techniques. In: IEEE ınternational conference on recent trends in electronics, ınformation and communication technology, pp 1889–1994

Harakannanavar SS, Vidyashree, Prashanth CR, Raja KB (2018b) Performance evaluation of face recognition based on multiple feature descriptors using Euclidean distance classifier. Int J Adv Netw Appl 10(3):3864–3879

Harakannanavar SS, Prashanth CR, Raja KB (2018c) Performance evaluation of face recognition based on the fusion of bit-plane and binary ımage compression techniques using euclidean distance classifier. Int J Intell Eng Syst 11(6):52–64

Harakannanavar SS, Prashanth CR, Raja KB, Patil S (2019a) Face recognition based on SWT, DCT and LTP. In: International conference on ıntegrated ıntelligent computing, communication and security, pp 565–573

Harakannanavar SS, Prashanth CR, Raja KB (2019b) MSB based ıris recognition using multiple feature descriptors. In: International conference on ınternational conference on ıntelligent computing, ınformation and control systems, pp 615–623

Harakannanavar SS, Vidyashree, Puranikmath VI, Prashanth CR, Raja KB (2019c) Comprehensive study of biometric authentication systems, challenges and future trends. Int J Adv Netw Appl 10(4):3958–3968

Harakannanavar SS, Vidyashree, Puranikmath VI, Prashanth CR, Raja KB (2019d) An extensive study of ıssues, challenges and achievements in ıris recognition. Asian J Electr Sci 8(1):25–35

Harakannanavar SS, Vidyashree, Puranikmath VI, Prashanth CR, Raja KB (2019e) Technical challenges, performance metrics and advancements in face recognition system. Int J Comput Sci Technol 7(3):836–847

Harakannanavar SS, Prashanth CR, Raja KB (2019f) Performance analysis of MSB based ıris recognition using hybrid features extraction technique. Int J Recent Technol Eng 8(6):1–10

Harakannanavar SS, Prashanth CR, Raja KB (2019g) Performance analysis of ıris recognition using multi-stage wavelet transform decomposition and bicubic ınterpolation technique. Int J Recent Technol Eng 8(6):230–239

Harakannanavar SS, Prashanth CR, Raja KB (2020a) Iris recognition using bicubic ınterpolation and multi level DWT decomposition. In: International conference on computational vision and bio ınspired computing, pp 1146–1153

Harakannanavar SS, Rudagi J, Puranikmath V (2020b) Development of algorithm for offline signature verification using fusion extraction techniques. Int J Seybold Rep 15(9):1421–1431

Huang P, Hilton A, Starck J (2010) Shape similarity for 3d video sequences of people. Int J Comput Vision 89(2):362–381

Jayalaxmi H, Anitha TG, Harakannanavar SS, Prashanthi HJ (2022) Robust ıris recognition algorithm using EMD and support vector machine. J Posit Sch Psychol 6(4):4279–4288

Kahale (2020) On the economic impact of social distancing measures. SSRN Electron J 89–98

Kajabad I (2019) People detection and finding attractive areas by the use of movement detection analysis and deep learning approach. Procedia Comput Sci 156:327–337

Kreiss A (2020) Joint human pose estimation and stereo 3D localization. In: IEEE ınternational conference on robotics and automation, pp 2324–2330

Landing (2020) A.I. Landing AI Creates an AI Tool to Help Customers Monitor Social Distancing in the Workplace. IEEE Magazines

Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) SSD: single shot multibox detector. In: European conference on computer vision. Springer, pp 21–37

Manfredi V, Calderara C (2014) Detection of static groups and crowds gathered in open spaces by texture classification. Pattern Recogn Lett 44:39–48

Mangshor NN, Rajudin MN, Aminuddin R (2021) Framework for JARAK: a monitoring and notifier system for social distancing. In: IEEE student conference on research and development, pp 422–426

Pahar M, Niesler T (2022) Machine learning based COVID-19 detection from smartphone recordings: cough, breath and speech. arXiv e-prints, arXiv-2104

Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: towards real-time object detectionwith region proposal networks. arXiv preprint arXiv:1506.01497

Wang X, Ng W, Liang (2017) Lapped convolutional neural nettworks for embedded systems. In: IEEE global conference on signal and ınformation processing, pp 1135–1139

Yadav S, Gulia P, Gill NS, Chatterjee JM (2022) A real-time crowd monitoring and management system for social distance classification and healthcare using deep learning. Hindawi J Healthc Eng 1–11

Zhang C, Vetter Z (2015) Sensor fusion for semantic segmentation of urban scenes. In: IEEE ınternational conference on robotics and automation, pp 1850–1857

Zou S, Guo Y, Ye J (2019) Object detection in 20 years: a survey. arXiv

Designing Energy Routing Protocol with Energy Consumption Optimization in Cognitive Radio Networks

Dileep Reddy Bolla, P. Ramesh Naidu, J. J. Jijesh, T. R. Vinay, Satya Srikanth Palle, and Keshavamurthy

1 Introductıon

Regarding energy economy and the smallest hop count in the routing protocol, the routing protocol in wireless sensor networks is crucial. The terminal node in a cognitive radio network, which may have any number of mobile nodes, often uses a battery to supply power and ensures it can communicate over extended distances. In cognitive radio networks, energy resources play a key part in this. One node in a network loses energy, which means that node cannot continue to participate in the data transmission process and is referred to as a dead node. Numerous issues, including data interruption faults in the link and excessive energy consumption, could be brought on by these dead nodes. We can create a forwarding link node that is positioned in the middle of the nodes to address this issue, but this would result in congestion at the data points and a bigger degree of transmission loss as a result of the link cluster head's energy exhaustion. Additionally, it has been found that routing protocols that disregard the energy component use a lot of energy and are expensive.

These days, there is a lot of room for energy-saving methods, such as sensor networks with clusters and a hierarchical routing protocol to lengthen the lifetime of

D. Reddy Bolla (✉) · P. Ramesh Naidu
Department of CSE, Nitte Meenakshi Institute of Technology, Yehahanka, Bangalore, India
e-mail: dileep.bolla@gmail.com

P. Ramesh Naidu
e-mail: ramesh.naidu@nmit.ac.in

J. J. Jijesh
Department of E&CE, Sri Venkateshwara College of Engineering, Bangalore, India

T. R. Vinay
Department of AIDS, M S Ramaiah Institute of Technology, Bangalore, India

S. Srikanth Palle · Keshavamurthy
Department of Electronics and Communications, Atria Institute of Technology, Bangalore, India

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_6

the network. It is noted that the energy state of the cluster heads and sensor nodes has not been taken into account in the LEACH protocol. To address this problem, various researchers have proposed various techniques, such as orthogonal frequency-division multiplexing (OFDM) type of sensors and optimal sleep–wake scheduling, but these have not had an impact and also have an issue with increased packet delay, because each of the sensor nodes has its own energy state.

In this work, we suggest an energy-efficient routing protocol for cognitive radio networks based on the comparative analysis conducted in this article. This protocol primarily focuses on enhancing network performance at various levels of energy stages, such as normal, warning, and dangerous stage. In the suggested study, we have concentrated on the node's energy consumption and the network lifespan residual energy model. Additionally, we conducted a comparison of the proposed EERP with the LEACH protocol and found certain advantages including lower energy consumption, a longer network survival time, a decrease in packet loss rate and packet delay of data delivery, and protection of low-energy nodes.

2 Related Work

The network throughput would drastically decrease as a result of such uncooperative behavior. We suggest a credit-based Secure Incentive Protocol (SIP) to encourage collaboration across mobile nodes with different objectives in order to solve this issue (Zhang et al. 2007). In addition to introducing energy-efficient uneven clustering protocol to APTEEN, the strategy also integrates cross-layer design methodology with routing and spectrum allocation. Ant colony algorithm is used to complete inter-cluster path search, which reduces the workload of the cluster head (Wang and Wang 2019). NP-complete optimization issue with an EQS route is serving as a workable solution. We develop a new deep reinforcement learning model that supports the DRQR protocol to create EQS routes in real time through offline training as opposed to online training as most literature research does in order to address this issue (Tran et al. 2022).

In order to accomplish efficient routing protocol operation in terms of maximum energy conservation, maximum-possible routing pathway setups, and minimal delays, a signaling mechanism and an energy-efficient system were developed based on a simulation scenario (Mastorakis et al. 2014). In order to address bandwidth constraints and battery life issues in wireless sensor networks, cognitive radio technology is being studied (Srividhya and Shankar 2022). The energy utility function is optimized by customizing the sensing period, sensing threshold, and number of collaborating SUs at the same time with the restriction of offering adequate protection for the primary user (PU) (Wu et al. 2014). But without considering energy efficiency, low-energy adaptive clustering hierarchy (LEACH) has been included into CWSN. The fact that node energy is finite and cannot be increased is one of the shortcomings of CWSN. To increase energy efficiency and increase the lifespan of CWSN, an efficient routing (Ge et al. 2018) protocol is required. In the CWSN, it is crucial to minimize energy use during data transmission. Along with the research work, an enormous work as in Bolla and Shivashankar (2017), Jijesh et al. (2021), Shankar et al. (2020), Palle et al. (2020), Bolla and Shankar (2020), Ding et al. (2017), Wang et al. (2013), Shome et al. (2021), and Ibrahim et al. (2022) is helpful in carrying out the work.

3 System Model for Energy Consumption of CRNs

As per the LEACH protocol, a system model for consumption of the energy for sending the nodes is discussed in Fig. 2 as follows.

In Fig. 2, we assume that the distance from transmitting node to receiving node is given by a variable d and energy consumption for the amplifier is E_{amp}. Energy to receive the signal is $E_{\min}$ and the relation between E_{amp} and $E_{\min}$ is given in Eq. 1.

$$E_{\text{amp}}(d) = kd^n E_{\min}, \tag{1}$$

where the value of n can be 2 or 4 and k is a constant. For instance, $n = 2$ and $k = 1$, then lifetime analyzed to lifetimes how outperforms lifetime down on 1 can be rewritten as in Eq. 2.

$$E_{\text{amp}}(d) = d^2 E_{\min}. \tag{2}$$

To calculate the energy transfer parameters and receiver parameters, we use E_{TX} and E_{RX}, respectively, where $\mathrm{E_{TR}}$ is the computational energy. And it is given by Eq. 3.

$$E_{TR} = E_{TX} + E_{RX}. \tag{3}$$

For sending or receiving of L bits of data, then we use Eqs. 4–6.

$$E_{TX}(l, d) = l(E_{\text{elec}}) + d^2 E_{\min}, \tag{4}$$

$$E_{RX}\ (l = lE_{\text{elec}}), \tag{5}$$

$$E_{TR} = E_{TX} + E_{RX} = 2lE_{\text{elec}} + ld^2 E_{\min}. \tag{6}$$

Further, residual energy (R) can be calculated based on the percentage of ratio given below as follows as in Eq. 7.

$$R = \frac{\text{current residual energy value}}{\text{initial energy value}}. \tag{7}$$

4 Routing Protocol

In the proposed work, we have focused on the residual energy, and based on this, the routing strategy has been proposed. To understand clearly, for instance, a link is said to be free if its R (overall) residual energy is more than zero. If not, the link is not free. The routing path has been established on the residual energy (with greater ER). It is essential for distributing the network's load. It is not assumed that the nodes will be sent if their leftover energy is in the danger stage. A prolonged survival time can be considered only for the source and destination nodes.

In this article, we have assumed two thresholds T_1 and T_2 are the node's relative residual energy values are thought to be, and the energy within the node has been separated into three stages. Stage 1: Normal stage (T_{S1}), Stage 2: warning stage (T_{S2}), Stage 3: danger stage (T_{S3}). T_1 is smaller than T_2. As per Eqs. 8–10, the proportion of remaining energy T is compared with T_1 and T_2.

$$T_{s1} = \text{normal} \quad T_2 < T < 1.0 \quad E_r \text{ is high,} \tag{8}$$

$$T_{s2} = \text{warning} \quad T_1 < T < T_2 \quad E_r \text{ is middle,} \tag{9}$$

$$T_{s3} = \text{danger} \quad 0.0 < T < T_1 \quad E_r \text{ is low.} \tag{10}$$

According to the equations above, the node is safeguarded and is not transmitted when it is in a dangerous stage.

4.1 Route Request

Any node that needs to transfer data must first check the availability of the routing database to see if a valid route is present. It must first send a route request (RREQ) message to connect with the closest node. According to Fig. 1, the RREQ is made up of the source ID, sequence number, destination ID, residual energy message, and relative residual energy value.

When the RREQ message is received by the intermediate nodes which are the neighboring nodes from the source node, then the protocol has to first verify the corresponding route concerning the routing table, and as per the requirements, the effective routing table is updated to overcome the loop generation sequence (14, 15, and 16). The process of continuous forwarding of the RREQ is continued until the

S-ID	D-ID	Seq-No	R_ msg	R_rel

Fig. 1 Route request RREQ message format

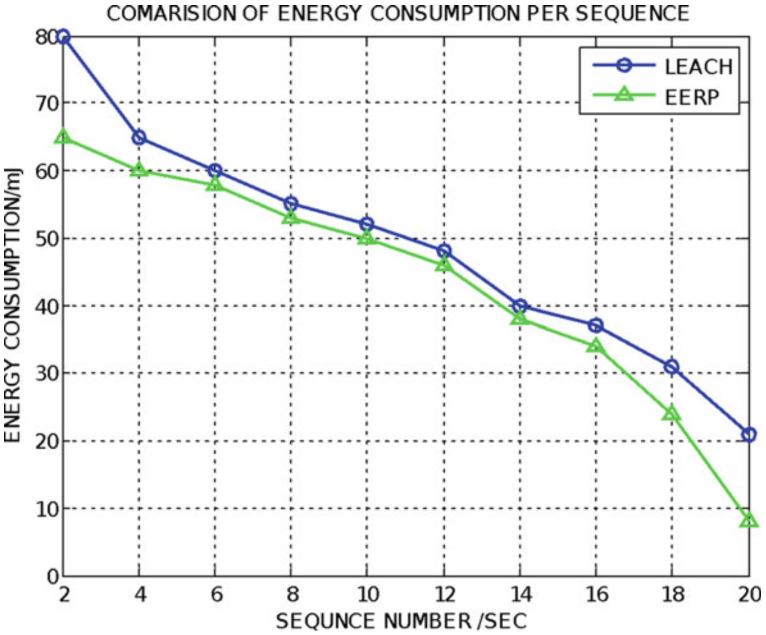


Fig. 2 Energy consumption versus sequence number

route to the destination node appears. The node that is being forwarded will obtain the information of the energy from the RREQ message format. With the acquired information, it compares it with its energy information, and then it decides and checks whether the information in the routing table needs to be updated or not. The relationship between the old and new energy levels is as follows, and the E (new) is as in Eq. 11.

$$\begin{aligned} &\textbf{If } E(i) < E(\text{old}) \\ &\textbf{Then} \\ &E(\text{new}) = E(i) \qquad . \\ &\textbf{Else} \\ &E(\text{new}) = E(\text{old}) \end{aligned}$$

The new energy E(new) is

$$E(\text{new}) = E(\text{old}) \sum_{i=0}^{N} E(i). \tag{11}$$

4.2 Route Discovery

The route discovery process is complete once the RREQ can be delivered to the source node. If the intermediate node has the destination node, the RREQ message is forwarded to the source, for example. The destination node will check the energy level information (E) with the threshold residual energy, say T_1 and T_2, after receiving the RREQ. The method of establishing the current link and completing the routing is based on these thresholds, and after a little wait, the RREQ source node receives a message indicating that the route discovery procedure is finished.

4.3 Route Establishment and Route Maintenance

In the proposed work, when the residual energy of a node is exhausted, then the node needs to find out the next-hop possible route sensing node. This involves the following two phases.

Phase 1: To select the neighboring sensing node, the node with the comparatively largest residual energy is used.
Phase 2: To pick up the neighboring sensing node for the next hop to transmit the data, we need to compare the neighbor sensing node residual energy with the predefined threshold value (E).
Phase 3: If the energy of the chosen neighboring sensing node is below the predefined threshold, we will reduce the predetermined threshold (E) and reselect the next-hop node.

Based on the three steps, the routing between the source node and destination node can be determined, previously described by taking into account the residual energy levels. But in order to maintain the route, the following route maintenance technique is used.

Once the route from source to destination has been established, the path is referred to as an active path. In the event that the source node has changed locations during data transmission, the route discovery process needs to be restarted. A route error (RERR) message is provided to the source node if the destination node or any intermediary nodes move.

There are two categories of route maintenance in this situation. During the source route rediscovery procedure, the source node must first broadcast the RREQ message to its neighbors along with the destination node sequence number, residual energy, and relative residual energy. In order to repair the damaged link by using another intermediate node, attention must be given with the upkeep of the intermediate nodes. In order to reestablish the path to the destination node, the corresponding intermediate nodes broadcast the RREQ message to their nearby nodes. When the path between the source node and the destination node is re-established, this operation can be finished.

5 Performance Analysis

Researchers have been able to assess the suggested routing protocol's performance in terms of typical energy usage, packet loss rate, and network longevity by contrasting it with the LEACH protocol. The comparative analysis of the above-mentioned metrics for the existing and the proposed works is presented as follows in Fig. 2.

5.1 *Packet Loss Rate*

This metric is helpful for tracking the effectiveness of cognitive radio networks. As per the results obtained, we observed that the packet loss rate of sensing the node is rising. This is due to the reason that the death date in the network nodes has been increased. The proposed work has a smaller packet loss rate because of the energy consumption optimization mechanism using residual energy concept, and further due to the maintenance of the lifetime analyzed to lifetime show out performs lifetime drop-down quality lifetime analyzed to lifetimes how outperforms lifetime of communication in the network as in Fig. 3.

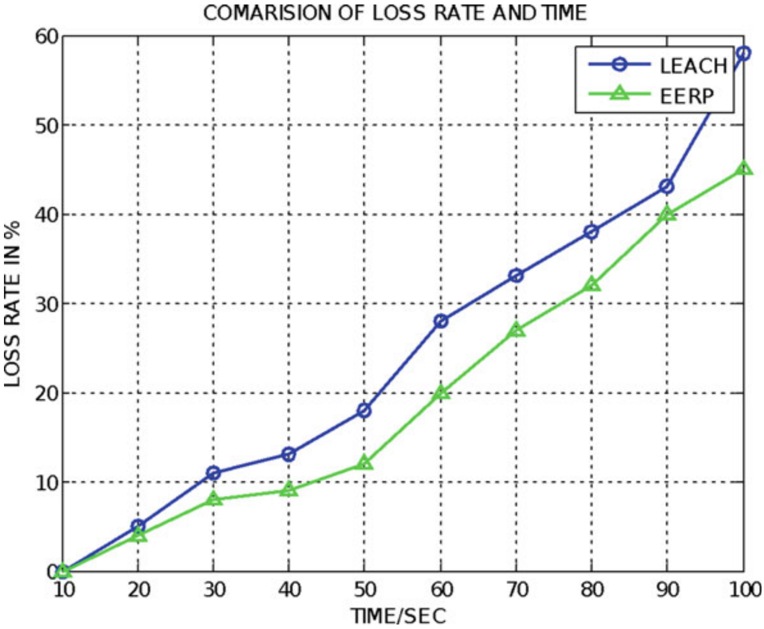


Fig. 3 Loss rate versus time

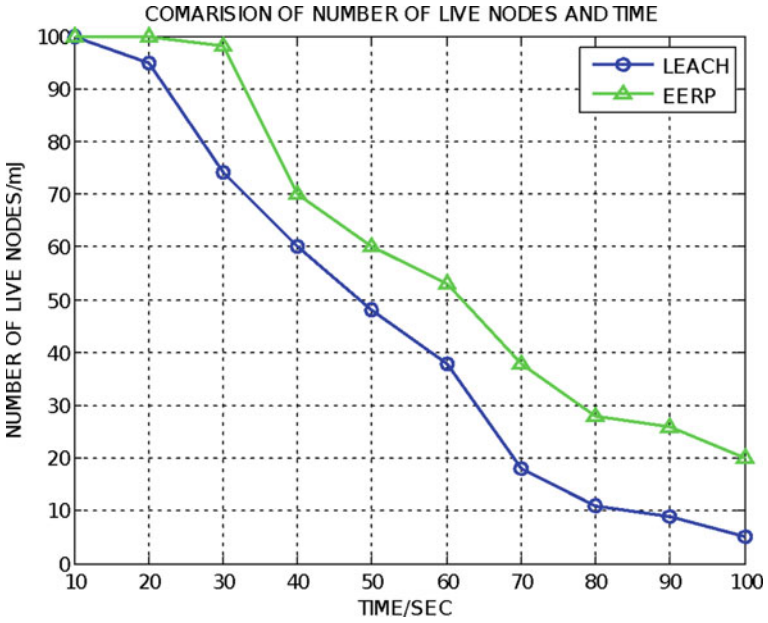


Fig. 4 Number of live nodes versus time

5.2 Network Lifetime

This is often determined by how long the network's sensing nodes survive. From the results obtained in Fig. 4 it is observed that as the network working hours increases with time.

The survival rate of network nodes has decreased, but the proposed protocol has a better number of live nodes than the existing protocol.

6 Conclusion

The research work proposed in the article is mainly focused on improvising the network lifetime in CRNs. As per the study carried out, in order to assess the proposed work, we assessed the two energy consumption optimization routing protocols, taking into account the network lifetime, node energy consumption, and packet loss rate. We found that the LEACH protocol performs better for CRNs. Overall, the outcomes show that the suggested approach performs better than the existing methodology. The energy consumption appears to have a rather positive impact on the network lifetime and consumption of energy is drop-down by about 5%, the live nodes concerned has been improved by about 13%, and the loss rate has been reduced by about 6%.

References

Bolla DR, Shankar S (2020) A robust QSCTA-EDRA routing protocol for cognitive radio sensor networks. Int J Commun Netw Distrib Syst 25(4):385–413. https://doi.org/10.1504/IJCNDS.2020.110523

Bolla DR, Shivashankar (2017) An efficient protocol for reducing channel interference and access delay in CRNs. In: 2017 2nd IEEE international conference on recent trends in electronics, information & communication technology (RTEICT), pp 2247–2251. https://doi.org/10.1109/RTEICT.2017.8257000

Ding H, Fang Y, Huang X, Pan M, Li P, Glisic S (2017) Cognitive capacity harvesting networks: architectural evolution toward future cognitive radio networks. IEEE Commun Surv Tutor 19(3):1902–1923. https://doi.org/10.1109/COMST.2017.2677082

Ge Y, Wang S, Ma J (2018) Optimization on TEEN routing protocol in cognitive wireless sensor network. J Wirel Commun Netw 2018:27. https://doi.org/10.1186/s13638-018-1039-z

Ibrahim NK, Sali A, Karim HA, Ramli AF, Ibrahim NS, Grace D (2022) Multiple description coding for enhancing outage and video performance over relay-assisted cognitive radio networks. IEEE Access 10:11750–11762. https://doi.org/10.1109/ACCESS.2022.3146396

Jijesh JJ, Jinesh JJ, Bolla DR, Sruthi PV, Dileep MR, Keshavamurthy D (2021) Development of food tracking system using machine learning. In: 2021 5th international conference on electrical, electronics, communication, computer technologies and optimization techniques (ICEECCOT), pp 802–806. https://doi.org/10.1109/ICEECCOT52851.2021.9708031

Mastorakis G et al (2014) Energy-efficient routing in cognitive radio networks. In: Resource management in mobile computing environments. modeling and optimization in science and technologies, vol 3. Springer, Cham. https://doi.org/10.1007/978-3-319-06704-9_14

Palle SS, Jijesh JJ, Bolla DR, Penna M, Sruthi PV, Alla G (2020) Modernized compartment with safety measures in railways. In: 2020 international conference on recent trends on electronics, information, communication & technology (RTEICT), 2020, pp 363–367. https://doi.org/10.1109/RTEICT49044.2020.9315565

Shankar S, Penna M, Bolla DR, Murthy K, Jijesh JJ, Navyashree HA (2020) Real time image classification based smart railway platform using convolutional neural network. In: 2020 international conference on recent trends on electronics, information, communication & technology (RTEICT), pp 398–402. https://doi.org/10.1109/RTEICT49044.2020.9315555

Shome A, Dutta AK, Chakrabarti S (2021) BER performance analysis of energy harvesting underlay cooperative cognitive radio network with randomly located primary users and secondary relays. IEEE Trans Veh Technol 70(5):4740–4752. https://doi.org/10.1109/TVT.2021.3073025

Srividhya V, Shankar T (2022) An energy efficient distance-based spectrum aware hybrid optimization technique for cognitive radio wireless sensor network. J Inst Eng India Ser B. https://doi.org/10.1007/s40031-022-00837-0

Tran T-N, Nguyen T-V, Shim K, da Costa DB, An B (2022) A deep reinforcement learning-based QoS routing protocol exploiting cross-layer design in cognitive radio mobile ad hoc networks. IEEE Trans Veh Technol. https://doi.org/10.1109/TVT.2022.3196046

Wang C, Wang S (2019) Research on uneven clustering APTEEN in CWSN based on ant colony algorithm. IEEE Access 7:163654–163664. https://doi.org/10.1109/ACCESS.2019.2950855

Wang J, Huang A, Wang W, Quek TQS (2013) Admission control in cognitive radio networks with finite queue and user impatience. IEEE Wirel Commun Lett 2(2):175–178. https://doi.org/10.1109/WCL.2012.122612.120918

Wu X, Xu JL, Chen M et al (2014) Optimal energy-efficient sensing in cooperative cognitive radio networks. J Wirel Commun Netw 2014:173. https://doi.org/10.1186/1687-1499-2014-173

Zhang Y, Lou W, Liu W et al (2007) A secure incentive protocol for mobile ad hoc networks. Wirel Netw 13:569–582. https://doi.org/10.1007/s11276-006-6220-3

Implementation of the AES Algorithm on FPGA

Bhukya Balaji Naik, Y. M. Sandesh, Naveen Kumar, and K. S. Vasundhara Patel

1 Introduction

Data privacy across unsecure networks is just one of numerous security concerns that have emerged as a result of the constant growth in Internet and wireless communications users (Rais and Qasim 2009). In order to protect user data from attacks and defeat them, cryptography emerged as a key strategy in order to maintain confidentiality and to keep away the unauthorized parties (Rais and Qasim 2009; Christy and Karthigaikumar 2012).

AES technique is one of the most widely used encryption protocols in the industry, protecting sensitive data from outside threats. The most widely used versions are 256-bit AES, which provides stronger security, and 128-bit AES, which guarantees superior performance throughout the encryption and decryption processes (Gupta et al. 2011). From one algorithm to the next, the process is different. Text can be encrypted or decrypted with the use of a key cypher. The plaintext data contained in cypher text is not readable by humans (Luo et al. 2011). AES is currently utilized for a variety of purposes. It is used to protect the personal data of the secret agencies which are kept on the cloud. It is also used to maintain sensitive information (Gupta et al. 2011; Luo et al. 2011).

The area and power comparison between Cadence Virtuoso and the Xilinx Zynq®-7000 SoC of the AES 128-bit algorithm designs are introduced in this paper. Ten

B. Balaji Naik (✉) · Y. M. Sandesh · N. Kumar · K. S. Vasundhara Patel
Department of Electronics and Communication, BMS College of Engineering, Bengaluru, India
e-mail: balajinaik@bmsce.ac.in

Y. M. Sandesh
e-mail: sandeshym.lvs21@bmsce.ac.in

N. Kumar
e-mail: naveenkumar.lv21@bmsce.ac.in

K. S. Vasundhara Patel
e-mail: vasu.ece@bmsce.ac.in

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_7

rounds make up the AES-128 bit, and a key extension module creates ten keys for each round.

This paper is structured as follows: The AES algorithm is briefly described while we compare the area and power parameters after that. The implementation, simulation, and synthesis results are presented, and the future scope is explained at the end of the paper.

2 Problem Statement

We have observed the implementation of AES on many devices, and it is implemented in various programming languages. We have implemented AES algorithm in Verilog HDL and compared the results in different tools in expectation of better results compared to one another. It is possible to implement on more advanced devices such as PYNQ board which is used for both embedded and hardware application.

3 AES Algorithm

When the data encryption standard algorithm, also known as the DES algorithm, was created and put into use, it made sense for that generation of computers. By today's standards of computation, the DES algorithm was easier and faster to crack each year. Longer key sizes and harder-to-crack ciphers called for a more reliable algorithm (Christy and Karthigaikumar 2012; Sumanth Kumar Reddy et al. 2011). To address this issue, they developed the triple DES, but due to its comparatively slower rate, it was never widely adopted (Advanced Encryption Standard 2001).

The AES was created as a result to address this flaw. AES-128 bits, AES-192 bits, and AES-256 bits are the three block ciphers that make up the symmetric advanced encryption standard (AES). Each cipher uses cryptographic keys that are 128 bits, 192 bits, or 256 bits in length to encrypt and decrypt the data in blocks of 128 bits, respectively (Zhang and Wang 2010). Depending on the key length that was used, the procedure runs through 10–12 or 14 execution rounds. The suggested architecture's base is AES-128-bit encryption. Each block of 128 data bits and the 128-bit cypher key are handled by a 4 * 4 state matrix and a key matrix, respectively (Advanced Encryption Standard 2001; Zhang and Wang 2010).

The state matrix and key matrix are initialized at the beginning of the algorithm using the original plaintext and the user input key, respectively. The ten rounds of AES-128 encryption are shown in Fig. 1. Each of the two matrices follows a different path through each round while performing various operations, and their outputs are merged in the add round key phase at the conclusion of each round. Sub-bytes, shift rows, mix columns, and add round keys are the four main transformations that make up the round block that handles the state matrix. The add round key modification is carried out initially by the algorithm. The add round key transformation is first applied

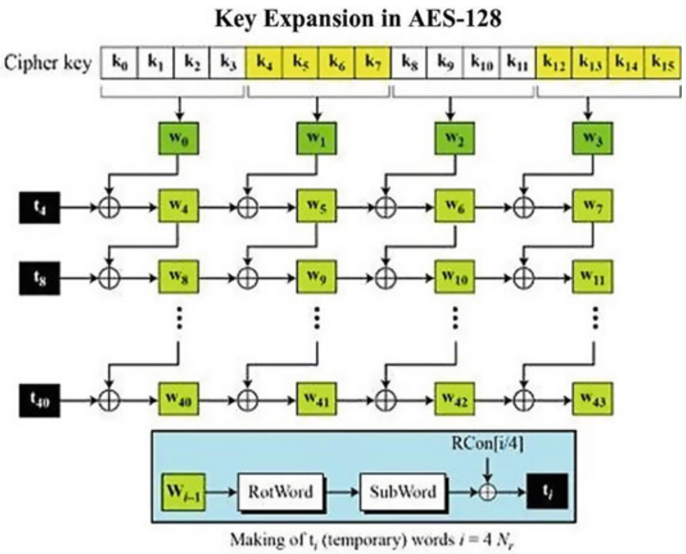


Fig. 1 Key expansion mechanism

to the original plain text and cypher key by the algorithm. Additionally, the final round of the algorithm is different from the first nine rounds in that it doesn't change the columns in a mix (Granado-Criado et al. 2010; Rahimunnisa et al. 2012; Maraghi et al. 2013). The cypher key, on the other hand, goes through many processing steps since it is put through a key schedule operation, which causes it to expand into ten keys throughout the course of the algorithm's ten rounds.

The following subsections provide explanations for each stage.

3.1 Sub-bytes Transformation

Byte is changed to a value in the S-box through a nonlinear conversion. Its use in the algorithm is predetermined for the S-box. Data substitution is done with the S-box. S-box can be thought of simply as a lookup table. Each block has 8 bits of data: The first 4 bits can be seen as row index, and the last 4 bits as column index. By using these row and column indexes, we can retrieve the value from the S-box (Rahimunnisa et al. 2012; Maraghi et al. 2013).

3.2 Shift Rows Transformation

In this procedure, each row of the state is cycled to the left based on the row index. There have been 0 spaces added to the left of the first row. One place is added to the left of the second row. There is a two-space leftward shift to the third row. Three spaces to the left, the fourth row is shifted (Mathew et al. 2011).

3.3 Mix Columns Transformation

The four-byte columns are now altered using a unique mathematical function. This function takes a single column's four bytes as input and replaces it with four brand-new bytes. The outcome is a second, brand-new matrix with 16 additional bytes. It should be made clear that this phase does not take place in the championship round (Mathew et al. 2011; Mozaffari-Kermani and Reyhani-Masoleh 2012). A transformation where a fixed polynomial is multiplied by each 4-byte column of the state matrix $a(x)$ modulo $x4 + 1$ and interpreted as a 4-term polynomial over GF (28), where

$$A(x) \text{ equal to } 03x3 + 01x2 + 0lx + 02x. \tag{1}$$

3.4 Add Round Key Transformation

The 16 bytes of the matrix, which are now regarded as 128 bits, are XORed with the 128 bits of the round key. If this is the last round, the output is the ciphertext. If not, the operation is repeated and the 128 bits that

3.5 Key Expansion

1. Employ the four-byte wi words. Four words make up subkey. In relation to AES and AES-128:
2. Cipher key = first subkey (w3, w2, w1, w0).
3. The formula for other words is as follows:

For each and every value of I that is not a multiple of 4, wi = wi-1 wi-4:

- RotWord: Left shift rotation of the w4k-1 bytes (nonlinearity).
- SubWord: All four bytes of Sub-Bytes fn are used (Diffusion).

- To break the symmetry, the result, rsk, is XORed with w4k-4 and a round constant, yielding w4k = rsk w4k-4 rconk.
- AES-192 bits and AES-256 bits key expansion is more challenging.

4 Encryption

The process of converting information into a secret code that conceals its true purpose is known as encryption. The cypher and variable that will act as the message's special key must both be selected by the sender at the beginning of the encryption procedure. Symmetric as well as asymmetric ciphers are the two most popular types of ciphers.

One key is all that is needed for symmetric cyphers, also referred to as secret key encryption. It is frequently referred to as a shared secret since the sender or computing system encrypting the communication must share the secret key with everyone allowed to decrypt the message. Asymmetric encryption frequently moves much more slowly than symmetric key encryption. Four subprocesses are included in each round: sub-bytes, shift rows, mix columns, and add round key (Zhang and Wang 2010; Granado-Criado et al. 2010; Karimian et al. 2012; Fig. 2).

5 Decryption

Data that has been encrypted can be decrypted and returned to their original form. Usually, encryption is carried out backward. It decodes the data to make decrypting the data possible only for a trustworthy user with access to the secret key or password.

Privacy is one of the reasons for using an encryption-decryption system. Examining unauthorized access from groups or people is necessary as information travels over the Internet. The AES cipher text can be returned to its original state using inverse encryption (Mozaffari-Kermani and Reyhani-Masoleh 2012). As was already noted, symmetric cryptography is used in the advanced encryption standard. In other words, both data encryption and decryption use the same key (Mozaffari-Kermani and Reyhani-Masoleh 2012; Karimian et al. 2012).

This distinguishes it from asymmetric encryption techniques, which demand both public and private keys. The inverted round key is used to start the AES decryption process in our scenario. The algorithm then reverses each and every step (shift rows, byte replacement, and later column mixing), until it is able to decrypt the original message.

6 Simulation Results

See Figs. 3 and 4.

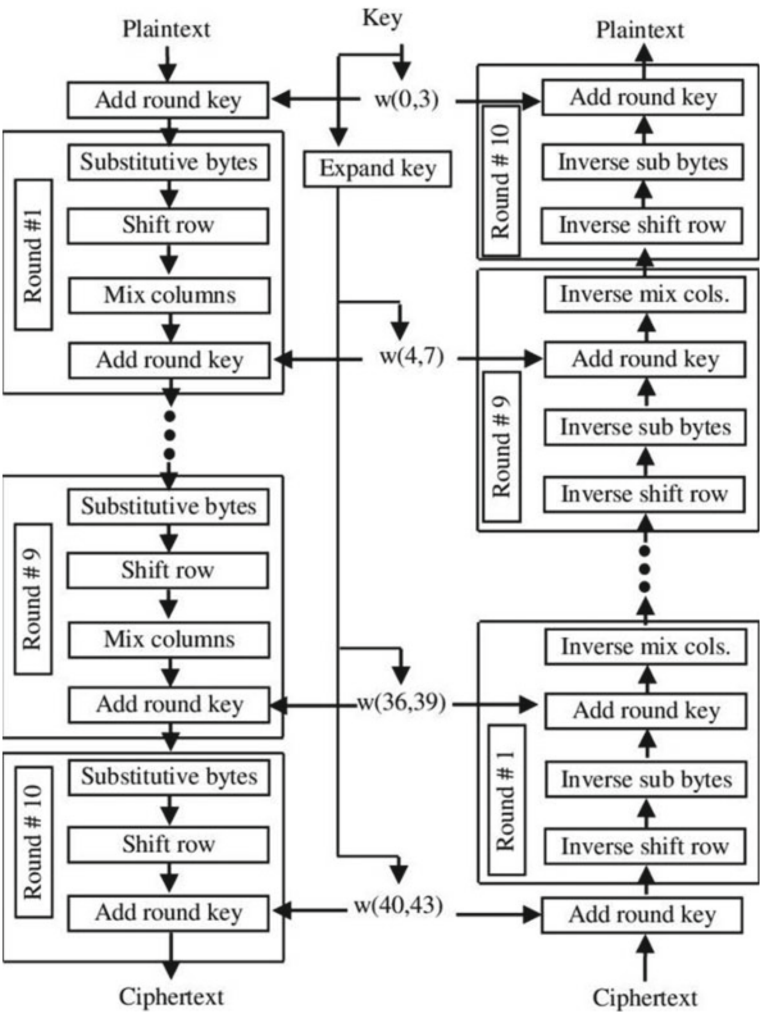
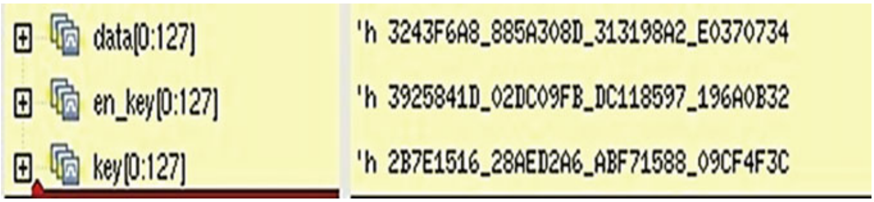
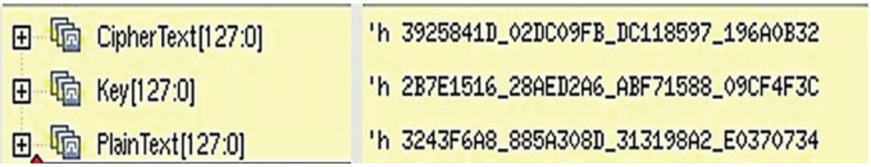


Fig. 2 Block diagram of AES encryption and decryption



```
data[0:127]     'h 3243F6A8_885A308D_313198A2_E0370734
en_key[0:127]   'h 3925841D_02DC09FB_DC118597_196A0B32
key[0:127]      'h 2B7E1516_28AED2A6_ABF71588_09CF4F3C
```

Fig. 3 Encryption result

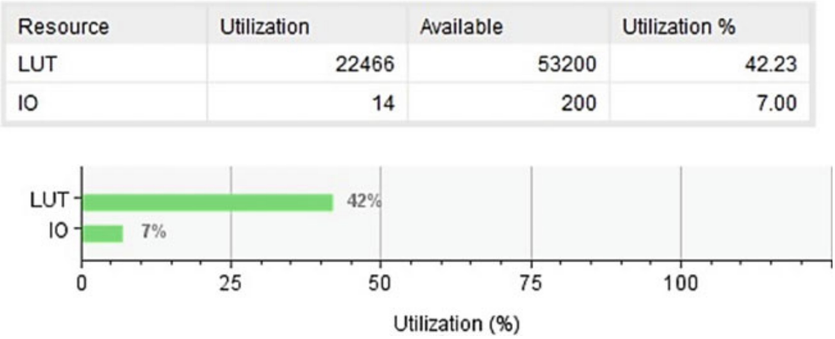


```
CipherText[127:0]   'h 3925841D_02DC09FB_DC118597_196A0B32
Key[127:0]          'h 2B7E1516_28AED2A6_ABF71588_09CF4F3C
PlainText[127:0]    'h 3243F6A8_885A308D_313198A2_E0370734
```

Fig. 4 Decryption result

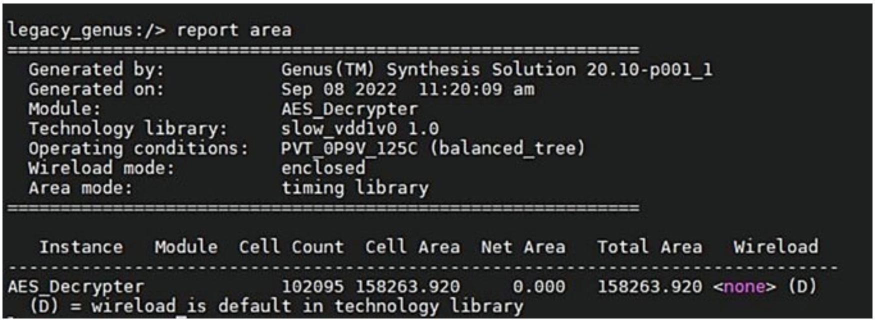
7 Area Analysis

It is observed that area utilization is better in the z-board compared to the cadence tool (Figs. 5 and 6).



Resource	Utilization	Available	Utilization %
LUT	22466	53200	42.23
IO	14	200	7.00

Fig. 5 In zed board



```
legacy_genus:/> report area
============================================================
  Generated by:           Genus(TM) Synthesis Solution 20.10-p001_1
  Generated on:           Sep 08 2022  11:20:09 am
  Module:                 AES_Decrypter
  Technology library:     slow_vdd1v0 1.0
  Operating conditions:   PVT_0P9V_125C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

   Instance   Module  Cell Count  Cell Area  Net Area   Total Area   Wireload
--------------------------------------------------------------------------------
AES_Decrypter              102095 158263.920     0.000   158263.920 <none> (D)
  (D) = wireload is default in technology library
```

Fig. 6 In cadence

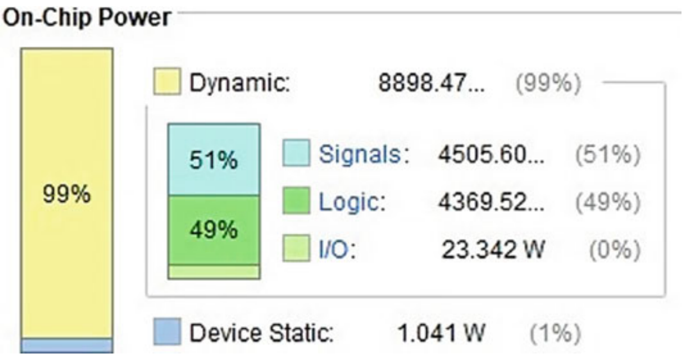
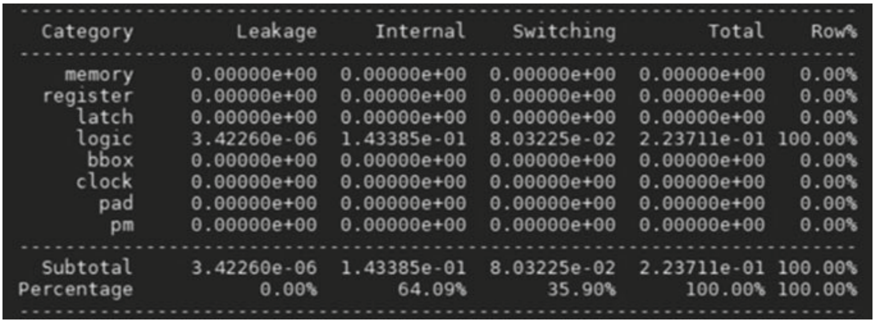


Fig. 7 Zed board



```
-------------------------------------------------------------------------------------
   Category         Leakage     Internal    Switching        Total    Row%
-------------------------------------------------------------------------------------
     memory     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00    0.00%
   register     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00    0.00%
      latch     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00    0.00%
      logic     3.42260e-06  1.43385e-01  8.03225e-02  2.23711e-01  100.00%
       bbox     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00    0.00%
      clock     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00    0.00%
        pad     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00    0.00%
         pm     0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00    0.00%
-------------------------------------------------------------------------------------
   Subtotal     3.42260e-06  1.43385e-01  8.03225e-02  2.23711e-01  100.00%
 Percentage           0.00%       64.09%       35.90%      100.00%  100.00%
-------------------------------------------------------------------------------------
```

Fig. 8 In Cadence

8 Power Analysis

We have seen better power reduction in z-board in comparison with cadence tool (Figs. 7 and 8).

9 Synthesis Result

The AES blocks are synthesized in Zynq-7000, XC7Z007S FPGA device. Synthesized block diagrams are shown in Fig. 9.

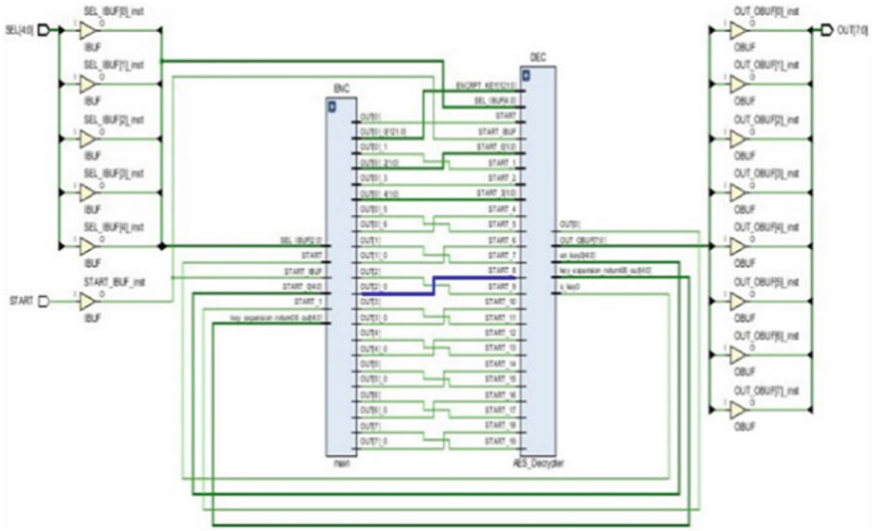


Fig. 9 Schematic of AES

10 Conclusion

This approach supports both the encryption and decryption standards, which is how the aforementioned proposed work, or AES core, functions. The design was tested using the Xilinx Vivado XC7Z007S for zed board FPGA and cadence simulation tool, and it was determined that area and power consumption are comparatively low in Xilinx Vivado XC7Z007S (Table 1).

This Rijndael algorithm-based implementation of AES core for 128-bit data blocks encourages the cryptographic standard to concentrate on even larger data blocks with lengths like 192 and 256 bits of plain text data and to go further with appropriately chosen key lengths in order to achieve the best cryptographic algorithm and deter adversaries and intruders. One of the most useful tools available today is characterized by its speed, versatility, and resistance to all types of cyberattacks, and its existence is a great blessing to us (Table 2).

Table 1 Synthesis report

Parameter	Design 1	Design 2
Vendor	Xilinx	Cadence
Device	XC7Z007S	Cadence
Slices	521	674
Slice LUTs	2490	1431
Frequency (MHz)	255.29	266.33
Throughput (Gbps)	32.08	34.09

Table 2 Dynamic power comparison

Parameter	Design 1	Design 2
Vender	Xilinx	Cadence
Device	XC7Z007S	Digiflow
Power consumption (mW)	170	223

11 Future Scope

When utilizing cypher keys with lengths of 128, 192, and 256 bits, an iterative private key symmetric block cypher, a popular strategy for advanced encryption, can accommodate data blocks of 128 bits. The 128-bit and 128-bit key AES cryptosystem is presented in this project as an efficient Verilog implementation. Xilinx is used to verify the description of the optimized and synthesizable Verilog code that is used to implement the 128-bit data encryption and decryption process. AES can be implemented on more advanced FPGA boards such as PYNQ to archive better results, and implementation of 192 bits and 256 bits is possible with better speed, power, area, etc. AES is a very secure form of encryption; however, in a few years it could not be completely impregnable. There haven't been any successful real-world attacks reported to date, but given how quickly technology is developing, there may be risks in the future. Mistakes also occur. If AES encryption is used improperly, hackers may gain access through implementation problems. Fortunately, effective attacks are prevented by using AES properly. As a result of their constant search for ways to defeat AES encryption, cryptographers have created a wide variety of theoretical attacks.

Up to now, only a few side-channel attacks have been successful, and nobody has been able to pull it off. Fortunately, no hacker will be able to break an AES system that has been properly set up. Therefore, assuming there isn't a mistake, your sensitive information is fully secure. There can be some possible hazards if the encryption is applied improperly.

References

Christy NA, Karthigaikumar P (2012) FPGA implementation of AES algorithm using Composite Field Arithmetic. In: 2012 international conference on devices, circuits and systems (iCDCS), pp 713–717

El Maraghi M, Hesham S, Abd El Ghany MA (2013) Real-time efficient FPGA implementation of AES algorithm. In: 26th international system on chip conference, Sept 2013, pp 203–208

FIPS 197. Advanced Encryption Standard (AES), 26 Nov 2001

Granado-Criado JM, Vega-Rodriguez MA, Sanchez-Perez JM, Gomez-Pulido JA (2010) A new methodology to implement the AES algorithm using partial and dynamic reconfiguration. Integration VLSI J 43:72–80

Gupta A, Ahmad A, Sharif MS, Amira A (2011) Rapid prototyping of AES encryption for wireless communication system on FPGA. In: 2011 IEEE 15th international symposium on consumer electronics (iSCE), pp 571–575

Karimian GH, Rashidi B, Farmani A (2012) A high speed and low power image encryption with 128-bit AES algorithm. Int J Comput Electr Eng 4(3)

Luo A-W, Vi Q-M, Shi M (2011) Design and implementation of area-optimized AES based on FPGA. In: 2011 international conference on in business management and electronic information (BMEI), pp 743–746

Mathew SK et al (2011) 53 Gbps native GF(24)2 composite field AES encrypt/decrypt accelerator for content-protection in 45nm high performance microprocessors. IEEE J Solid-State Circuits 46(4):767–776

Mozaffari-Kermani M, Reyhani-Masoleh A (2012) Efficient high performance parallel hardware architectures for the AES-GCM. IEEE Trans Comput 61(8):1165–1178

Rahimunnisa K, Karthigaikumar P, Rasheed S, Jayakumar J, Suresh Kumar S (2012) FPGA implementation of AES algorithm for high throughput using folded parallel architecture. J Secur Commun Netw 5(10)

Rais MH, Qasim SM (2009) Efficient hardware realization of advanced encryption standard algorithm using Virtex-5FPGA. Int J Comput Sci Netw Secur 9(9):59–63

Rashidi B, Rashidi B (2013) FPGA based a new low power and self-timed AES 128-bit encryption algorithm for encryption audio signal. Int J Comput Netw Inf Secur 10–20

Sumanth Kumar Reddy, S., R. Sakthivel, P. Praneeth, VLSI implementation of AES crypto processor for high throughput, International journal of advanced engineering sciences and technologies, vol. 6, no. 1, pp. 022–026, 2011.

Zhang V, Wang X (2010) Pipelined implementation of AES encryption based on FPGA. In: 2010 IEEE international conference on information theory and information security (icmS), Dec 2010

Security System Using PIR Sensor

K. Ashwitha, B. R. Puneeth, B. Sumithra, and Mohammed Taiyab

1 Introduction

Any person must have the need for security. For a calm life, it is essential that we feel secure and everything is okay. But how can one achieve that sense of security in this dangerous world where crime, terror, and terrorism are at their highest? We have a solution in the form of security systems using PIR sensor, and more and more people are installing them as a result to feel protected and secure (Salunke et al. 2021). For security and safety reasons, a variety of electronic security systems can be utilized at home and other crucial workplaces.

Our paper mainly depends on a scenario, and the model is constructed based on that. This security system model is designed in scenarios where expensive diamonds are kept for the showcase in a palace. Whenever a person or any object is around the expensive product within the measured circles, the security team will be notified, and the action would be taken on priority.

K. Ashwitha (✉)
Nitte Meenakshi Institute of Technology, Bengalore, Karnataka, India
e-mail: ashwitha.gk@nmit.ac.in

B. R. Puneeth
Nitte (Deemed to be University), NMAM Institute of Technology, Nitte, Karnataka, India

B. Sumithra
Business Applications Developer, Synopsys Inc., RMZ Infinity, Bengalore, Beneganahalli, India

M. Taiyab
Integra Micro System Pvt Ltd., Bellary Road, 12Th KM, Jakkur, Bangalore 560064, India

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_8

A security alarm is a tool used for safety (Takkar et al. 2021). It has numerous applications in the security and defense sectors, ranging from the protection of low-value organizational assets to the security of basic household goods. They were formerly pricey answers to security issues. This type of security system is becoming more accessible due to cost-cutting measures and rapid technical improvements.

A transmitter and a receiver are typically used in an electronic security system. In this case, the receiver receives an IR that was sent or transmitted by the transmitter. The device's IR beam is turned off when a person walks or passes across it, setting off the alarm. However, there are drawbacks to this kind of device, including poor line of sight and a small signal range. Using a PIR sensor, we can get rid of these drawbacks. A wireless security system called a PIR sensor-based security system places pyroelectric infrared motion sensors on each of the four sides of the area to be monitored, i.e., the front, back, left, and right.

2 Literature Review

A simple alarm unit and manually switch dependent sensors were suggested by Harshal Hemane and Debarati Sen. When a human moves in front of the motion sensor on a laser security system, the system's alert is set off by the person's body heat. In their research paper, "LASER BASED SECURITY SYSTEM FOR HOME," the alarm also alerts the security monitoring business and the local law enforcement (Hemane et al. 2022).

According to M. N. Lafta et al., design and fabrication of alarming system based on laser and LDR, the laser and LDR systems have a broad operating range and are highly sensitive. The laser's light is detected by an LDR that is connected to the circuit. Any time the Light's beam is altered, a warning is raised. This extremely reactive, low-computing-requirement approach is ideal for industrial applications, intelligent surroundings, and monitoring (Lafta et al. 2022).

The cloud-based intruder security system that Francis et al., proposed when compared to other security systems, the cloud-based intruder security system is more effective and safer. It can also be accessed and remotely viewed from any location, making it easy to guard. Despite the rapid advancement of intruder detection and security technologies, there are still certain problems with installation and maintenance expenses that need to be fixed (Lafta et al. 2022).

The purpose of the review is to raise awareness of the security alarm's use of Internet of Things (IoT) technology and the security measures that one should take on a regular basis due to the rise in burglaries and kidnappings in society today. It also aims to highlight the difficulties that arise when a security system is connected to the internet and how to prevent all cyber-criminal attacks.

3 Working Process

Security is the primary ideology. This is based on a PIR sensor that has an IC that emits buzzer or siren noises. The PIR sensor detects the infrared radiation that people release, and it subsequently generates a digital output. It is primarily utilized in applications for automatic lighting, security alarms, and motion detectors (Francis et al. 2022). They typically detect movement by detecting changes in infrared radiation. The Arduino Uno receives this digital output in return.

When a human or any kind of object is found, it thus makes the sound. A buzzer is activated.

It is used to produce sounds like police, fire engine, ambulance, and machine gun sirens, among others.

After the detection, the expensive ornament has been pulled down to a security place through a 12 V motor.

The System Works in the Following Steps

- The system developed is kept running always.
- The detector is connected to the sensing circuit.
- When the beam is interrupted and cannot reach the detector, its voltage output changes, and the circuit sense the change and put out a warning signal.
- Once the warning signal is on. The admin is notified with the message.

4 Working Principle

PIR sensors can often detect animal or human movement within a specified range. PIR is constructed of a pyroelectric sensor, which can recognize various infrared radiation intensities. The detector passively absorbs energy rather than emitting any of its own.

It picks up environmental infrared radiation. The pyroelectric gadget (Budijono et al. 2014) produces an abrupt electrical signal when focused on the optical system once infrared radiation from the human body particles with temperature is present.

Simply said, a human body or any other animal that passes by blocks the PIR sensor's first slot. The two bisects have a positive differential change as a result. The sensor creates a negative differential shift between the two bisects when a human being departs the sensing area (Fig. 1).

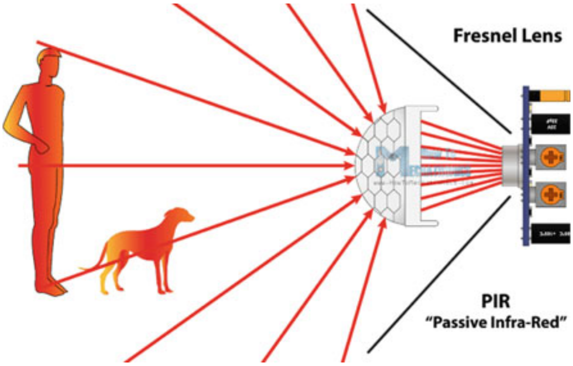


Fig. 1 Representation of human being interference in sensing area

5 Circuit Components

A. NodeMCU.
B. PIR sensor.
C. 12 V motor.
D. Buzzer.
E. Connecting wires.
F. Breadboard.
G. USB cables A–B.
H. Relay module.
I. 12 V power supply.

A. *Node MCU*: A low-cost System-on-a-Chip (SoC) called the ESP8266 serves as the foundation of the open-source Node Microcontroller Unit (NodeMCU). The Espressif Systems-designed and -produced ESP8266 has all of the essential components of a computer, including CPU, RAM, networking (WiFi), and even a contemporary operating system and SDK. This makes it a fantastic option for all types of Internet of Things (IoT) projects (Fig. 2).
B. *PIR Sensor*: An electronic sensor that measures the infrared (IR) light emitted by objects in its field of vision is known as a passive infrared sensor (PIR sensor). Most frequently, they are utilized in PIR-based motion detectors (Soltani et al. 2022). PIR sensors are frequently utilized in autonomous lighting and security alarm systems. PIR sensors can detect movement in general but cannot identify who or what moved (Fig. 3).
C. *12 V Motor*: A straightforward rotation motor spins 360° while running at 6000 RPM (Fig. 4).
D. *Buzzer*: A beeper or buzzer, for example, could be electromechanical, piezo-electric, or mechanical in design. The major function of this is to convert the signal from audio to sound (Andriansyah et al. 2017). It is often powered by DC

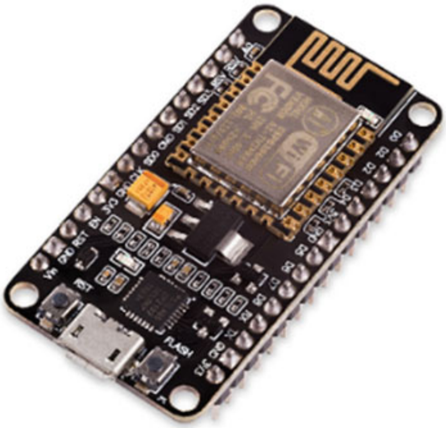


Fig. 2 NodeMCU

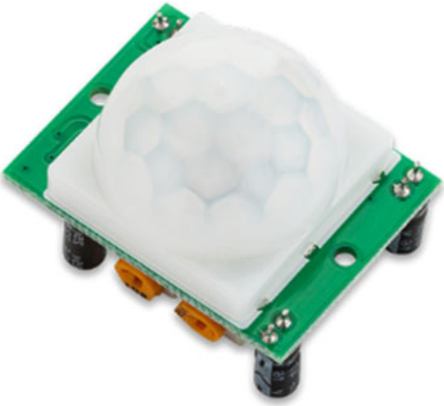


Fig. 3 PIR sensor

voltage and used in timers, alarm clocks, printers, computers, and other electronic devices. It can produce a variety of sounds, including alarm, music, bell, and siren, according on the varied designs (Fig. 5).

E. *Relay Module*: An electrical switch that is activated by an electromagnet is a power relay module. A distinct low-power pulse from a microcontroller is used to trigger the electromagnet (Satheeshkumar et al. 2017). The electromagnet pulls when it is turned on, allowing a circuit to be opened or closed (Fig. 6).

F. *12VDC 2A*: A 12 V power supply that you can connect into a standard outlet and works with inputs of 100–240 VAC. The most dependable and likely to function for all of the components you select (Saranraj et al. 2020). Use it when your circuit doesn't need to be mobile or situated somewhere without wall power (Fig. 7).

Fig. 4 12 V motor

Fig. 5 Buzzer

Fig. 6 Relay module

Fig. 7 12VDC 2A

6 Circuit Diagram

How each component is connected to each other to achieve the project "Security System."

A. For 12 V power supply, negative terminal is connected to DC 12 V motor negative terminal.
B. For 12 V power supply, positive terminal is connected to relay model (COM) terminal.
C. Relay model (NC) terminal is connected to DC 12 V motor positive terminal.
D. Positive terminal of PIR sensor is connected to 3.3 V of NodeMCU.
E. Negative terminal of PIR sensor is connected to GND of NodeMCU.
F. Output terminal of PIR sensor is connected to D1 pin of NodeMCU.
G. Negative terminal of buzzer is connected to GND pin of NodeMCU.
H. Positive terminal of buzzer is connected to D2 pin of NodeMCU.

We have programmed NodeMCU using Arduino IDE. Figure 9 shows the code that was used to upload into NodeMCU. Figure 10 shows how Admin/Security being notified when motion is detected (Fig. 8).

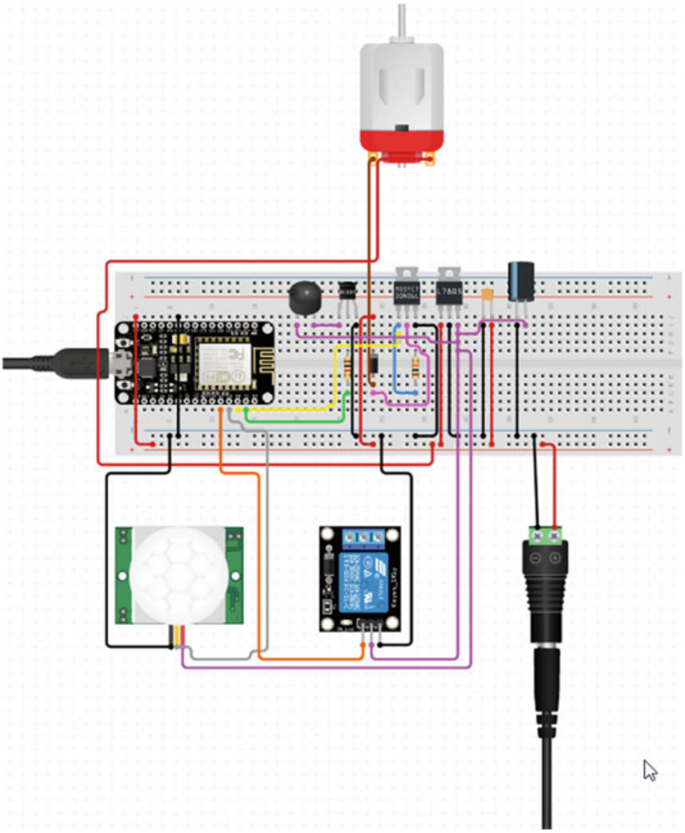
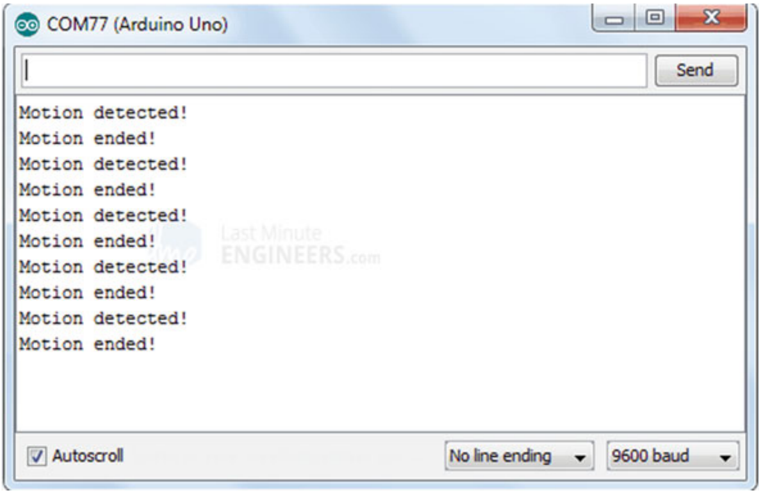


Fig. 8 Circuit diagram of the proposed model of security system



COM77 (Arduino Uno)

```
Motion detected!
Motion ended!
Motion detected!
Motion ended!
Motion detected!
Motion ended!
Motion detected!
Motion ended!
Motion detected!
Motion ended!
```

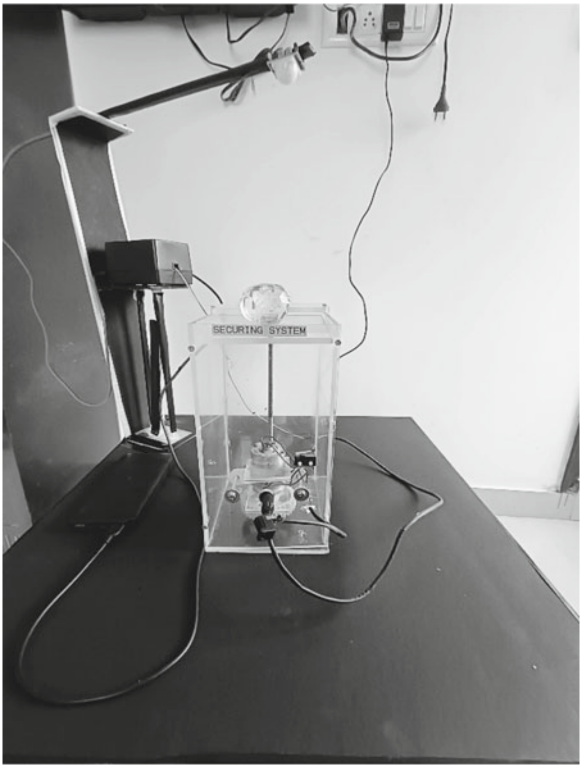
☑ Autoscroll

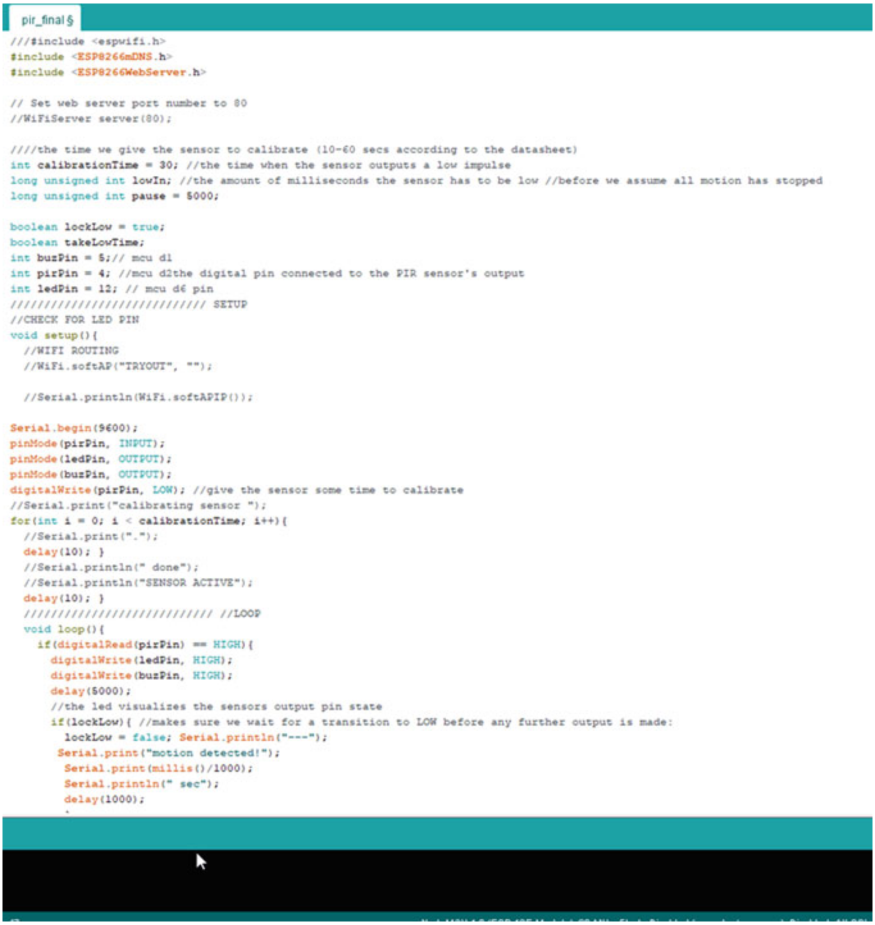
No line ending

9600 baud

Fig. 9 Code to upload NodeMCU

7 Project Image





```
///#include <espwifi.h>
#include <ESP8266mDNS.h>
#include <ESP8266WebServer.h>

// Set web server port number to 80
//WiFiServer server(80);

////the time we give the sensor to calibrate (10-60 secs according to the datasheet)
int calibrationTime = 30; //the time when the sensor outputs a low impulse
long unsigned int lowIn; //the amount of milliseconds the sensor has to be low //before we assume all motion has stopped
long unsigned int pause = 5000;

boolean lockLow = true;
boolean takeLowTime;
int buzPin = 5;// mcu d1
int pirPin = 4; //mcu d2the digital pin connected to the PIR sensor's output
int ledPin = 12; // mcu d6 pin
////////////////////////////// SETUP
//CHECK FOR LED PIN
void setup(){
  //WIFI ROUTING
  //WiFi.softAP("TRYOUT", "");

  //Serial.println(WiFi.softAPIP());

Serial.begin(9600);
pinMode(pirPin, INPUT);
pinMode(ledPin, OUTPUT);
pinMode(buzPin, OUTPUT);
digitalWrite(pirPin, LOW); //give the sensor some time to calibrate
//Serial.print("calibrating sensor ");
for(int i = 0; i < calibrationTime; i++){
  //Serial.print(".");
  delay(10); }
  //Serial.println(" done");
  //Serial.println("SENSOR ACTIVE");
  delay(10); }
  //////////////////////////// //LOOP
  void loop(){
    if(digitalRead(pirPin) == HIGH){
      digitalWrite(ledPin, HIGH);
      digitalWrite(buzPin, HIGH);
      delay(5000);
      //the led visualizes the sensors output pin state
      if(lockLow){ //makes sure we wait for a transition to LOW before any further output is made:
        lockLow = false; Serial.println("---");
       Serial.print("motion detected!");
        Serial.print(millis()/1000);
        Serial.println(" sec");
        delay(1000);
```

Fig. 10 Notification detected during motion

8 Conclusion

Author created the security on such a low budget. It had received complete security protection. High-tech home security options like laser security systems were previously exclusively accessible to the wealthy. It is a simple alarm system with manually switch dependent sensors. When a human moves in front of the sensor on a security system, that person cuts the laser, setting off the alarm.

References

Andriansyah M, Subali M, Purwanto I, Irianto SA, Pramono RA (2017) e-KTP as the basis of home security system using Arduino Uno. In: 2017 4th international conference on computer applications and information processing technology (CAIPT), pp 1–5. https://doi.org/10.1109/CAIPT.2017.8320693

Budijono S, Andrianto J, Noor MAN (2014) Design and implementation of modular home security system with short messaging system. EPJ Web Conf 68:00025. https://doi.org/10.1051/epjconf/20146800025

Dehghani Soltani M et al (2022) Safety analysis for laser-based optical wireless communications: a tutorial. Proc IEEE 110(8):1045–1072. https://doi.org/10.1109/JPROC.2022.3181968

Francis A, Thanga S, Thiruneelakandan T (2022) Cloud based intruder security system. https://doi.org/10.5281/zenodo.5839759

Hemane H, Sen D (2022) Laser based security system for home. 2395-0056

Lafta MN et al (2022) Design and fabrication of alarming system based on laser and LDR. In: 2022 5th international conference on engineering technology and its applications (IICETA), pp 550–553. https://doi.org/10.1109/IICETA54559.2022.9888487

Salunke A et al (2021) Laser tripwire security system using Arduino Uno. Int J Eng Res Appl (IJERA) 11(6):56–59

Saranraj B, Dharshini NSP, Suvetha R, Bharathi KU (2020) ATM security system using Arduino. In: 2020 6th international conference on advanced computing and communication systems (ICACCS), pp 940–944. https://doi.org/10.1109/ICACCS48705.2020.9074429

Satheeshkumar K, Ajithkumar N, Gopinath PA, Ranjithkumar S, Chandramohan J (2017) Implementation of smart home automation and security system using Arduino and Wi-Fi through android application. Int J Eng Res Technol (IJERT) ICONNECT 5(13)

Takkar S et al (2021) Advanced ATM security system using Arduino Uno. In: 2021 9th international conference on reliability, Infocom technologies and optimization (trends and future directions) (ICRITO), pp 1–5. https://doi.org/10.1109/ICRITO51393.2021.9596341

A Novel Design and Implementation of Full Adder Circuit Using QCA and Qiskit

Suman Mondal, Shruti Gatade, N. Samanvita, and Karthiganesh

1 Introduction

Complementary metal oxide semiconductor (CMOS), which is used to construct integrated circuits in various devices, is one of the most prominent and well-known technologies in today's world as far as the computer chip design business is concerned. In comparison with other MOS technologies available today, CMOS technology offers a lower power dissipation. The number of components on a chip would double every 18 months, according to Gordon Moore's law, which was introduced in 1965 (Kalogeiton et al. 2017). High consumption, low speed, and density beyond 10 nm limit the usage of CMOS technology. As a result, numerous studies have been carried out in order to identify various technologies that could be used to replace traditional CMOS technology (Bahar et al. 2013).

As an alternative to CMOS technology, QCA transistor-less technology, single electron transistor (SET), and carbon nanotube (CNT) are now being utilized. When compared to equivalent implementations using conventional VLSI technology, QCA on the nanoscale has a bright future because of its potential to achieve high performance in terms of clock frequency, device density, and power consumption (Gassoumi et al. 2021).

Basic logic gates, which can be realized using quantum gates, are the fundamental building blocks of logic circuits (Laajimi et al. 2017). Logic circuit implementations have been a focus of quantum system designers (Safoev et al. 2017; Singh et al. 2016). These were potentially constructed using quantum gates with a qubit characterization.

In the proposed work, the quantum gates and instantiation mechanism to build and implement full adder circuit. The fundamental logic circuits can be instantiated to perform high-end logic operations. As a result, the design elements become simpler

S. Mondal · S. Gatade (✉) · N. Samanvita · Karthiganesh
Department of Electrical and Electronics Engineering, ICAP Centre, Nitte Meenakshi Institute of Technology, Bengaluru, India
e-mail: Shruti.gatade@nmit.ac.in

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_9

because complicated code is no longer necessary, and these circuits are validated by simulation using the IBM quantum (Thalaimalaivanaraj et al. 2020) simulator.

The QCA design of basic quantum gates such as CNOT and Toffoli, as well as the construction of a full adder employing these gates in the QCA designer tool and IBM Qiskit simulator, is described in this work. The rest of the paper is organized as following, a summary of the QCA technology, quantum computing and working with qubits, the CNOT, Toffoli and full adder design, the simulation results in QCA designer tool, and the simulation results using Qiskit simulator.

Quantum Cellular Automata (QCA) Background

QCA represents data as a charge configuration rather than charge flow. Basic QCA cell has four quantum dots arranged in a square and has two stable states ‘1’ and ‘0’ (Vahabi et al. 2021). When two cells are put next to each other, they tend to assume the same configuration. Basic native gate in QCA is the majority gate with three inputs A, B, and C and one output. Computation happens in the central part. Truth table looks like AND and OR gates in the same gate. Majority gate works as the output is the majority of the input (Singh et al. 2016). Figure 1 shows a basic QCA cell and a QCA wire.

A clock is an important part in QCA that ensures which device holds onto the value and which device assumes its values (Daddala et al. 2015). The clock signals in QCA are generated by an electric field given to the cells, which either raises all of the tunneling barriers between dots within a QCA cell or lowers them all. QCA designer tool provides 4 clocks: Clock 0, Clock 1, Clock 2, and Clock 3. CMOS wires can also supply this electric field to the clock when it is burned under the QCA circuitry.

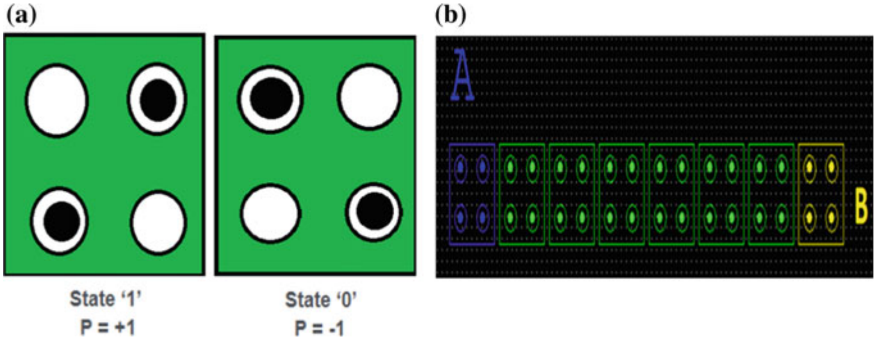


Fig. 1 **a** Basic quantum cell **b** QCA wire

2 Quantum Computing and Qubit

Quantum computing operates by controlling the behavior of particles like electrons and photons in a different way from regular computers. It simulates two bits at one time and is very fluid with non-binary identities or combinations. It can exist in a superposition of zeros and ones with some probability of being zero and some probability of being one. It has an identity on a spectrum.

Qubits use superposition to reduce the number of steps required to complete a computation. When a single phosphorus atom is placed in a silicon crystal immediately close to a transistor, the outermost electron of that atom is used as a qubit. The electron now has a magnetic dipole known as spin, which has two orientations: up and down, which can be represented as a traditional 1 and 0. A high magnetic field has to be provided to discern the energy state of the electron when it is in spin up or spin down. For this, a superconducting magnet is employed. The electron will be in its lowest energy state, with its pin pointing down, and it will need some energy to spin it up. We can place the qubit in a span of state by striking it with a microwave pulse, but it must be of a precise frequency to write information in it (Raman et al. 2022). The frequency is determined by the magnetic field in which the electron is located. As a result, if the pulse is generated at a certain location, a special quantum superposition of the spin up and spin down states is achieved, with a specified phase between the superpositions.

In this work, we use the IBM quantum circuit simulator, which is a collection of high-performance simulators and algorithms. It is a general-purpose simulator, and the optimum method required for the simulation is selected on the basis of the input parameters and circuit provided. Generic code can also be written using the Qiskit library, which could also be used for generating circuits and its relevant components. Qiskit is an open-source software library that covers the entire stack, from the actual interface with IBM Q hardware up to application-level algorithms. As a result, the tool is divided into four libraries that are each called after one of the four ancient elements terra, aqua, aer, and ignis. When integrated, the four Qiskit libraries provide the most comprehensive back-to-back quantum computing software solutions. They all interact with one another through shared data structure elements. The Qiskit software library is completed by an extensive educational collection (Sharma et al. 2022) that spans the entire spectrum of courses for beginners to professionals and from quantum theory to low-level notebook-assisted implementations of specialized quantum circuits.

3 Basic Quantum Gates and Full Adder

(a) CNOT Gate

The CNOT gate operates on two qubits. The control qubit is usually the first qubit to be realized. The target qubit is the second of the two qubits. The CNOT gate operates in base states which is expressed as (Fig. 2):

$$f(x, y) = (x, x \oplus y)$$

First input x is control bit; if it is 0, no effect on second bit; and if it is 1, it acts like a NOT gate on second bit.

(b) Toffoli Gate

Toffoli gate, commonly known as controlled-controlled NOT gate (CCNOT), is a universal reversible logic gate comprising 3-bit inputs and outputs (Bahar et al. 2013). When the first two bits are both 1, the third bit is inverted; otherwise, all bits remain unchanged (Tables 1 and 2).

(c) Full Adder

A full adder is required by most digital circuits in order to execute addition and subtraction. It adds two inputs and a carry in bit, which results in a sum and a carry out bit. As a result (Fig. 3), it has three inputs and two outputs, respectively. The circuit diagram of full adder is shown in Fig. 4.

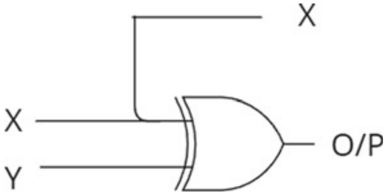


Fig. 2 Logic diagram for CNOT (XOR)

Table 1 Truth table for CNOT (XOR)

Input		Output	
X	*Y*	*X*	O/P
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Table 2 Truth table for Toffoli gate

Input			Output		
X	*Y*	*Z*	*X′*	*Y′*	*Z′*
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

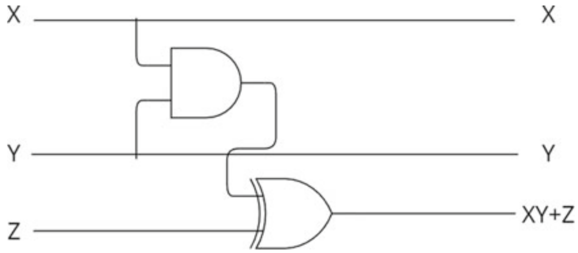


Fig. 3 Logic diagram for Toffoli gate

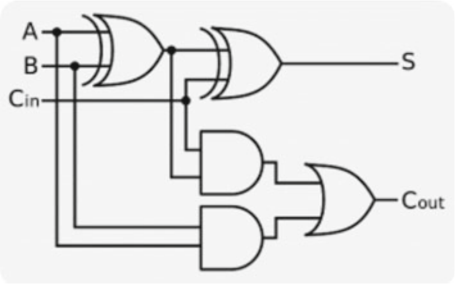


Fig. 4 Logic diagram for full adder circuit

4 Circuit Simulation and Results

(a) CNOT Gate (XOR Gate)

The CNOT gate performs a two-qubit operation. As per the simulation graph shown in Fig. 5b, we can observe that whenever both input values are the same that is ‘0’ or ‘1’, then corresponding output obtained is low, and whenever the input values are different (Fig. 5), the output obtained is high (Table 3).

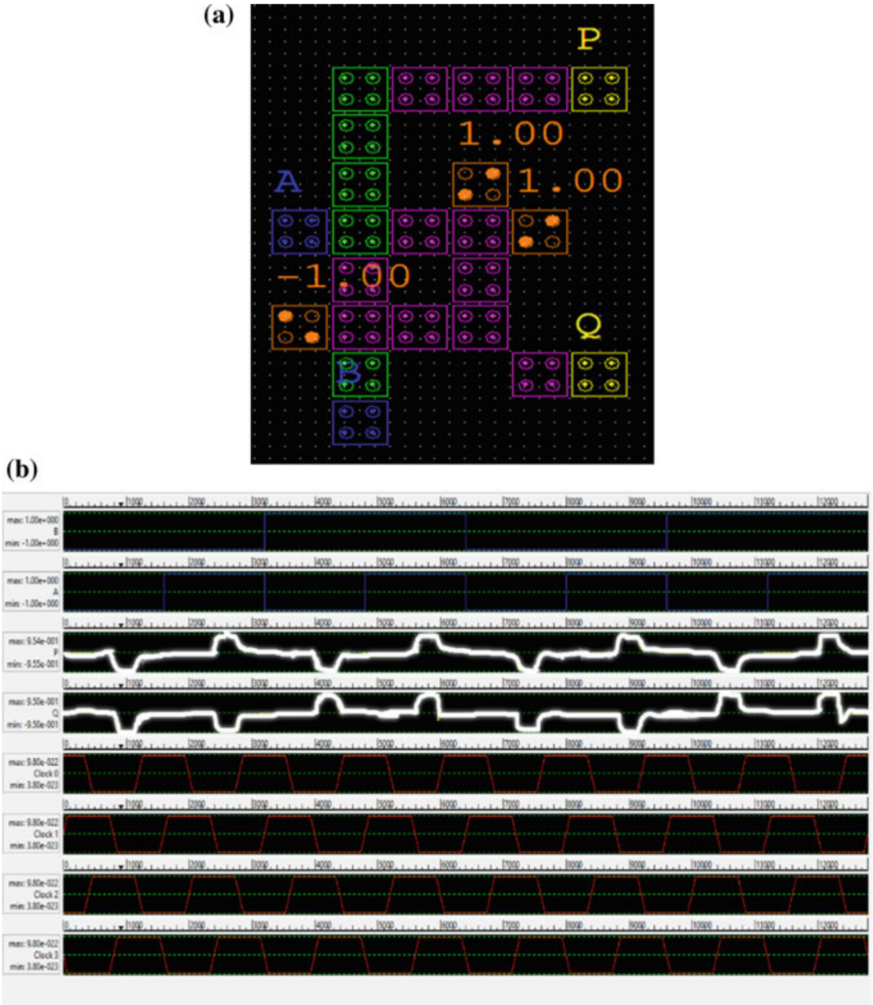


Fig. 5 **a** CNOT gate circuit gate **b** simulation output waveform of CNOT

(b) Toffoli Gate

The Toffoli gate performs a three-qubit operation. The circuit works as an AND gate circuit. One qubit is the control qubit, and when the remaining two qubits are the same, the output obtained is high; else, low output is obtained. Implementation of the circuit diagram and corresponding simulation output waveform are shown in Fig. 6a, b.

(c) Full Adder

A full adder circuit performs addition of the inputs. It adds three inputs—two operands and a carry bit (Gassoumi et al. 2021; Subashini and Koteeshwari 2018).

Table 3 Full adder truth table

Input			Output	
X	*Y*	C_{in}	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

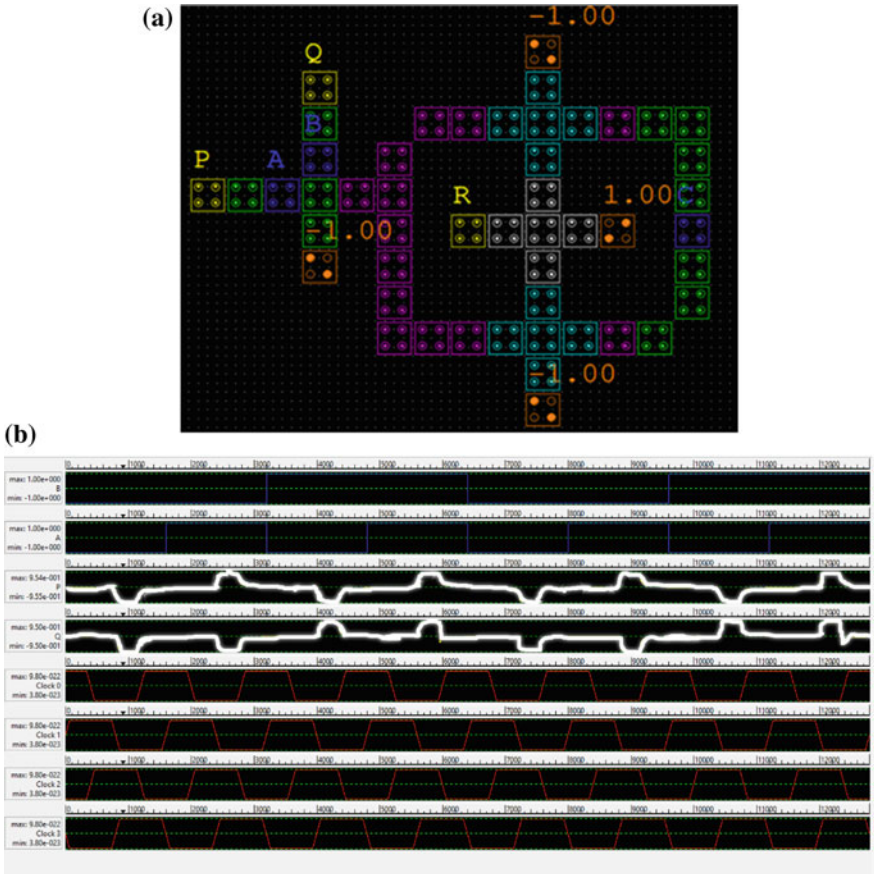


Fig. 6 **a** Toffoli gate circuit implementation **b** simulation output waveform

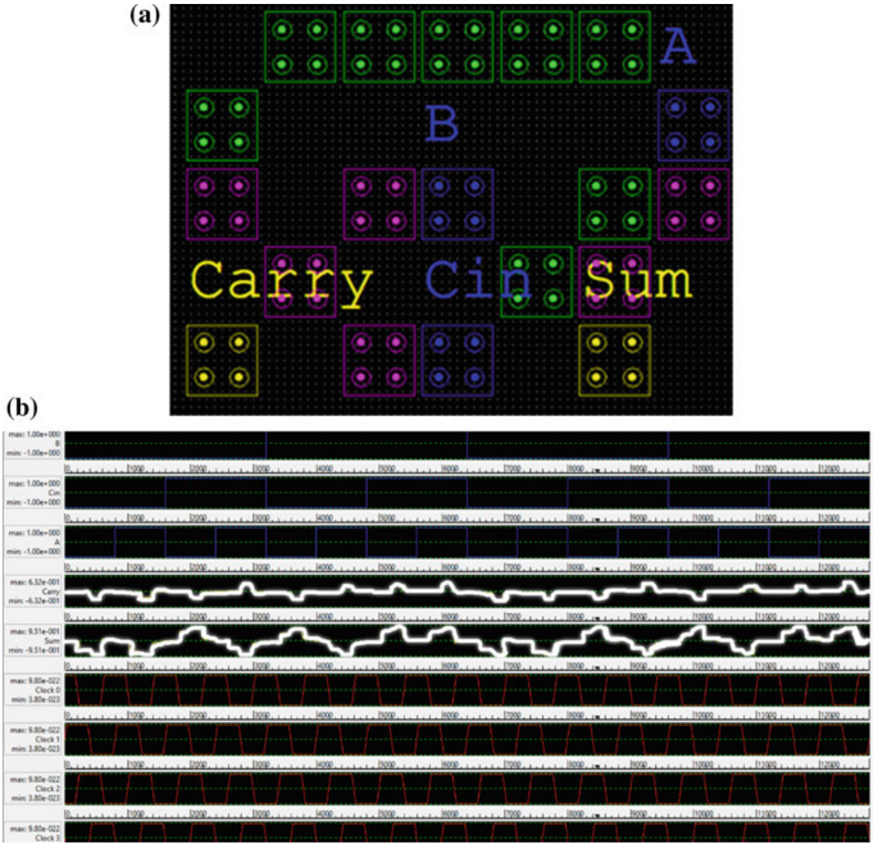


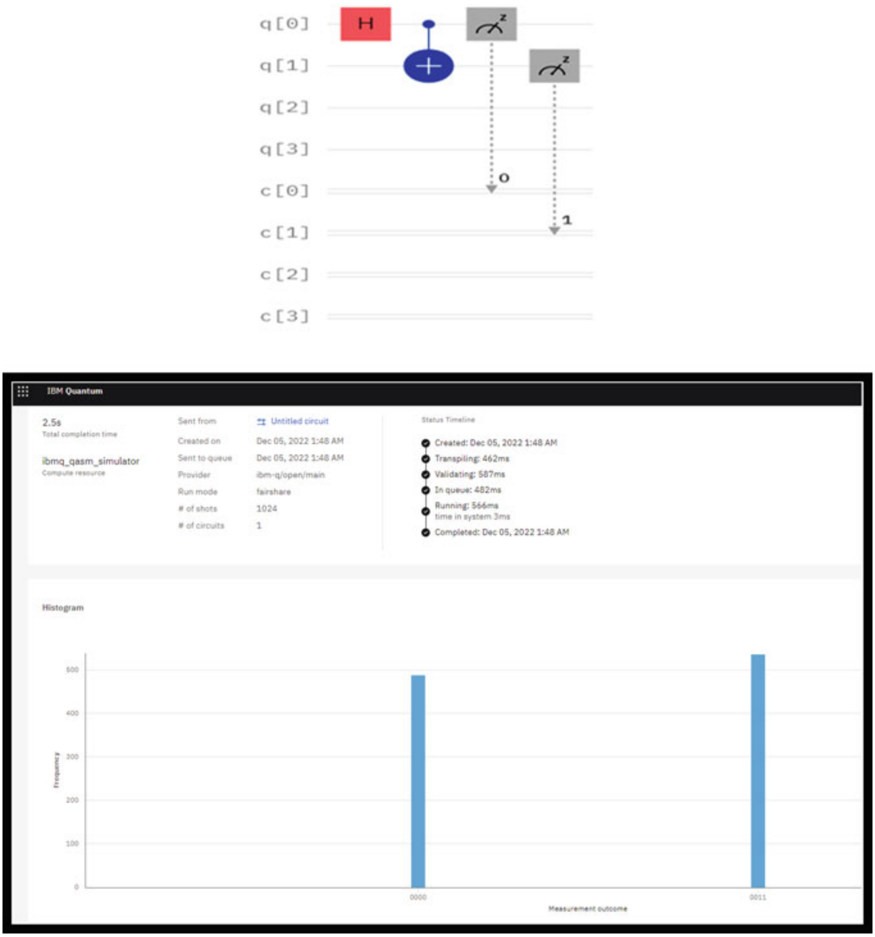
Fig. 7 **a** Full adder circuit implementation **b** simulation output waveforms

It outputs a sum and a carry bit value. As per the simulation graph, depending on the inputs given, the values of sum and carry in and carry out are carried forward and displayed. In the above figure, the output is shown as a yellow color waveform with carry and sum as the initials (Fig. 7).

5 Qiskit Simulation Results

(a) CNOT Gate

CNOT gate is simulated using IBM Quantum Composer. It uses 1024 shots to complete the simulation. Before execution, the state is 0000. After simulation, the frequency values 000 and 011 have probability of 522 and 502 (Fig. 8).



IBM Quantum

2.5s
Total completion time

ibmq_qasm_simulator
Compute resource

Sent from	Untitled circuit
Created on	Dec 05, 2022 1:48 AM
Sent to queue	Dec 05, 2022 1:48 AM
Provider	ibm-q/open/main
Run mode	fairshare
# of shots	1024
# of circuits	1

Status Timeline

- Created: Dec 05, 2022 1:48 AM
- Transpiling: 462ms
- Validating: 587ms
- In queue: 482ms
- Running: 566ms
  time in system 3ms
- Completed: Dec 05, 2022 1:48 AM

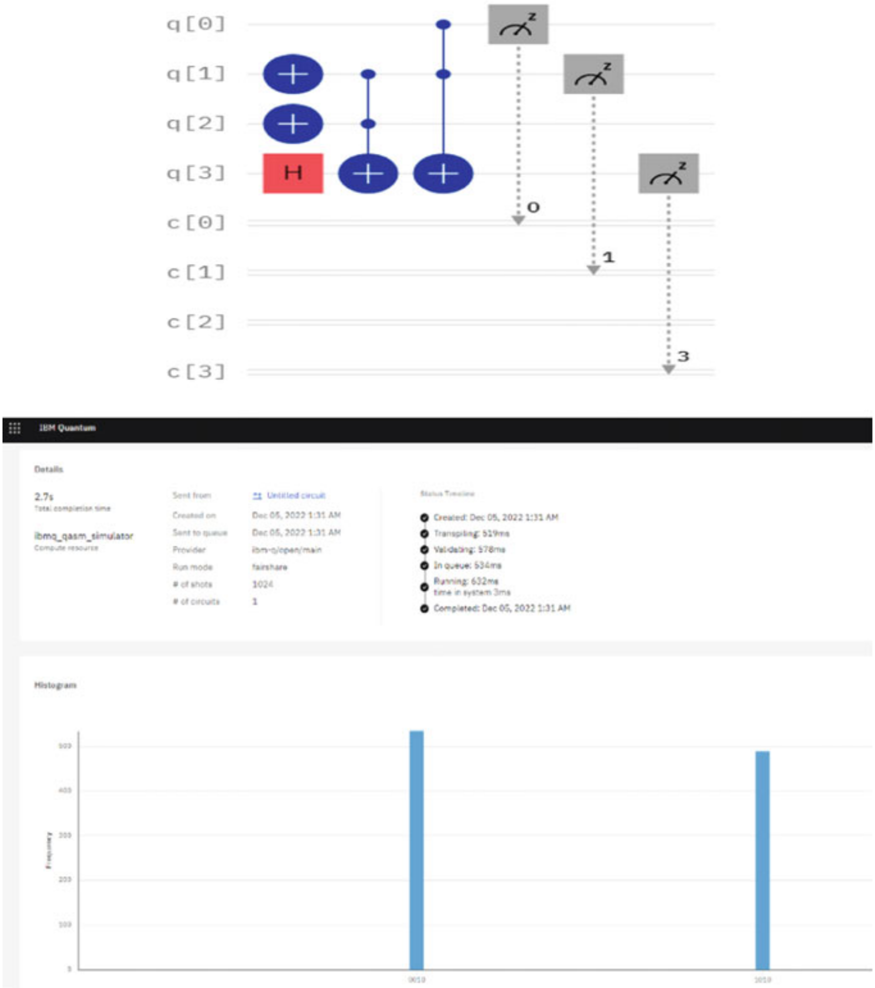
Fig. 8 CNOT gate simulation Qiskit simulator

(b) Toffoli Gate

Toffoli gate is simulated using IBM Quantum Composer. It uses 1024 shots to complete the simulation. Before execution, the state is 1110. After simulation, the frequency values 0010 and 1010 have probability of 522 and 502 (Fig. 9).

(c) Full Adder

Full adder circuit is simulated using IBM Quantum Composer. It uses 1024 shots to complete the simulation. Before execution, the state is 1001. After simulation, the frequency values 1001 and 1024 have probability of 522 and 502.



Details

2.7s
Total completion time

ibmq_qasm_simulator
Compute resource

Sent from	Untitled circuit
Created on	Dec 05, 2022 1:31 AM
Sent to queue	Dec 05, 2022 1:31 AM
Provider	ibm-q/open/main
Run mode	fairshare
# of shots	1024
# of circuits	1

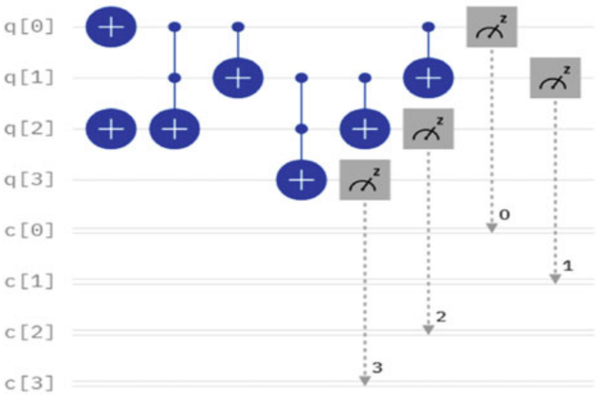
Status Timeline

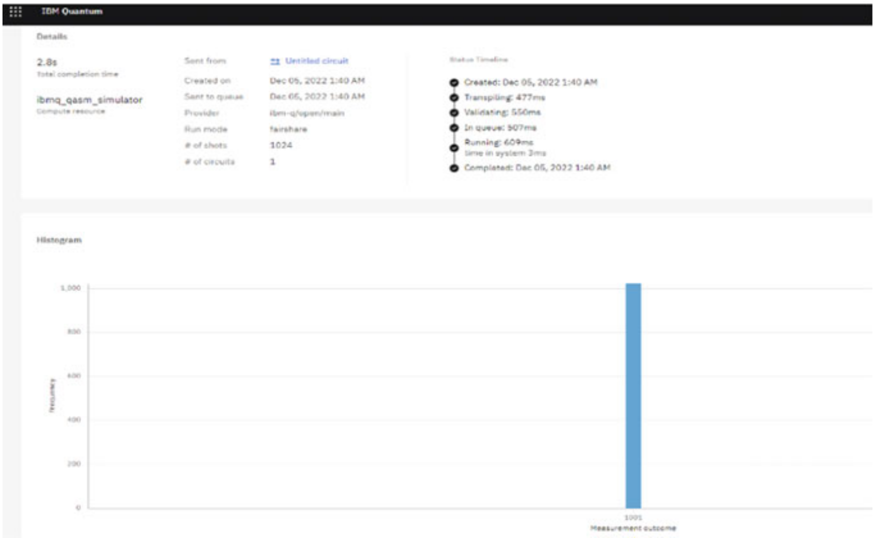
- Created: Dec 05, 2022 1:31 AM
- Transpiling: 519ms
- Validating: 578ms
- In queue: 534ms
- Running: 632ms
  time in system 3ms
- Completed: Dec 05, 2022 1:31 AM

Fig. 9 Toffoli gate simulation Qiskit simulator

The CNOT, Toffoli gate, and full adder are implemented in QCAD and IBM Qiskit simulator (Fig. 10). The below table shows the time taken for simulation in both platforms.

Circuit name	QCAD (in s)	Qiskit (in s)
CNOT gate	1	2.5
Toffoli gate	1	2.7
Full adder circuit	1	2.8





Details

2.8s
Total completion time

ibmq_qasm_simulator
Compute resource

Sent from: Untitled circuit
Created on: Dec 05, 2022 1:40 AM
Sent to queue: Dec 05, 2022 1:40 AM
Provider: ibm-q/open/main
Run mode: fairshare
# of shots: 1024
# of circuits: 1

Status Timeline
- Created: Dec 05, 2022 1:40 AM
- Transpiling: 477ms
- Validating: 550ms
- In queue: 507ms
- Running: 609ms
  time in system 3ms
- Completed: Dec 05, 2022 1:40 AM

Histogram

Fig. 10 Full adder implementation using simulation Qiskit simulator

CNOT Gate

Simulator used:	IBM Quantum Composer (using Qiskit)
State:	0000 (probability: 51.75781%) (before execution) 0011 (probability: 48.24219%) (before execution)
Frequency:	000 (488) (after execution) 011 (536)
No. of shots:	1024 (QASM simulator) 32 qubits

Toffoli Gate

Simulator used:	IBM Quantum Composer (using Qiskit)
State:	1110 (probability: 1) (before execution)

Frequency:	0010 (535) (after execution) 1010 (489)
No. of shots:	1024 (QASM simulator) 32 qubits

Full Adder

Simulator used:	IBM Quantum Composer (using Qiskit)
State:	1001 (probability: 1) (before execution)
Frequency:	1001 (1024) (after execution)
No. of shots:	1024 (QASM simulator) 32 qubits

6 Conclusion

QCA technology is a non-traditional circuit design pattern. CMOS technology drawbacks, primitive growth, and challenges with VLSI chip design are presented in this paper. Simulation is carried out in two software, i.e., QCA and Qiskit. QCA gives the information about the design of the chip, and Qiskit results tell us about the probability occurrence of the qubits. Using the QCAD simulator, we were able to realize the QCA structures of CNOT and Toffoli gates, as well as view the cell structure of a full adder circuit. Simultaneously, we observed similar circuits being executed in a quantum simulator using Qiskit and compared their entire execution time.

References

Bahar AN, Habib MdA, Biswas NK (2013) A novel presentation of Toffoli gate in quantum-dot cellular automata (QCA). Int J Comput Appl. https://doi.org/10.5120/14149-2243

Daddala A, Ramanaiah KV, Sumalatha V (2015) An efficient design of XOR gate and its applications using QCA. https://doi.org/10.26634/jele.5.3.3394

Gassoumi I, Touil L, Mtibaa A (2021) An efficient design of QCA full-adder-subtractor with low power dissipation. J Electr Comput Eng 2021:9, Article ID 8856399. https://doi.org/10.1155/2021/8856399

Gholami M (2019) A novel efficient full adder–subtractor in QCA nanotechnology. Int Nano Lett 9:51–54. https://doi.org/10.1007/s40089-018-0256-0)

https://qiskit.org/

Kalogeiton VS, Papadopoulos DP, Liolis O (2017) Programmable crossbar quantum-dot cellular automata circuits. IEEE Trans Comput Aided Des Integr Circuits Syst. https://doi.org/10.1109/TCAD.2016.2618869

Laajimi R, Ajimi A, Touil L, Bahar AN (2017) A novel design for XOR gate used for quantum-dot cellular automata (QCA) to create a revolution in nanotechnology structure. Int J Adv Comput Sci Appl 8(10). https://doi.org/10.14569/IJACSA.2017.081036

Mohammadi M, Mohammad M, Gorgin S (2016) An efficient design of full adder in quantum-dot cellular automata (QCA) technology. Microelectron J 50:35–43

Mokhtari D, Rezai AH, Rashidi H, Rabiei F, Emadi S, Karimi A (2018) Design of novel efficient full adder circuit for quantum-dot cellular automata technology. In: Series electronics and energetics, vol 31, no 2, pp 279–28. https://doi.org/10.2298/FUEE1802279M

Raman S, Samanvita N, Ahmed T, Aishwaraya A, Karthiganesh D (2022) Implementation of parity generator and checker using quantum-dot cellular automata. In: 2022 IEEE 2nd Mysore sub section international conference. https://doi.org/10.1109/MysuruCon55714.2022.9972362

Safoev N, Lee J-S, Jeon J-C (2017) QCA XOR gate for arithmetic and logic circuit design. Int J Adv Comput Eng Networking. ISSN: 2320-2106

Santra A, Santra S (2015) Design and simulation of quantum cellular automata based XOR gate with optimize complexity and cell count. JETIR. ISSN: 2349-5162

Sharma U, Pradeep K, Samanvita N, Raman S (2022) Implementation and performance evaluation of binary to gray code converter using quantum dot cellular automata. In: Inventive systems and control. Springer, Singapore, pp 299–321

Subashini V, Koteeshwari RS (2018) Design and implementation of full adders using QCA. SSRG Int J VLSI Sig Process SSRG-(IJVSP), special issue ICETSST. ISSN: 2394-2584

Singh G, Sarin RK, Raj B (2016) A novel robust exclusive-OR function implementation in QCA nanotechnology with energy dissipation analysis. https://doi.org/10.1007/s10825-016-0804-7

Thalaimalaivanaraj A, Raj M, Lakshminarayanan G (2020) Energy-efficient coplanar adder and subtractor in QCA. In: ICSSIT 2020. ISBN: 978-1-7281-5821-1

Vahabi M, Lyakhov P, Bahar AN (2021) Design and implementation of novel efficient full adder/subtractor circuits based on quantum-dot cellular automata technology. Adv Inf Process Methods Appl 11:8717. https://doi.org/10.3390/app11188717

GitHub Users Recommendations Based on Repositories and User Profile

R. Nagaraj, G. R. Ramya, and S. Yougesh Raj

1 Introduction

By offering code hosting services and team-based software development frameworks, GitHub has gained the attention of many software engineers globally as a prominent software development platform, a social coding, and a web-based Git repository hosting service that enables the developers to engage in open-source project documentation, design, coding, and testing in a socio-cultural context. The important feature of this platform is that it allows following other users or organizations, so that they may get alerts about their public activity. This feature has enhanced user collaboration by encouraging them to explore more repositories and promoting the conversation on GitHub.

Moreover, GitHub is considered as a social media platform designed specifically for developers where they can duplicate the relevant projects and then customize the features of those projects to match their requirements. Also, users can ask for assistance from or have a conversation with people who are on their following lists when they run into technical difficulties while developing a repository, bug patches, code restructuring, and feature enhancements.

Even though the developers have the option to search for the projects or do the above things, it takes a lot of time and might disappoint them. So, to overcome this we analyzed a method to identify and recommend the followers working on similar domains to the user by using different centrality measures of network analysis, link prediction between the users, and recommending followers based on the content-based filtering technique.

The paper has five sections where Sect. 2 describes the related works and emphasizes the key characteristics used by us to implement the different measures of

R. Nagaraj · G. R. Ramya (✉) · S. Yougesh Raj
Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Coimbatore, India
e-mail: gr_ramya@cb.amrita.edu

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_10

social network analysis and some techniques to identify the prominent followers. Section 3 has a detailed explanation of methodology analysis and the experimental findings from the implementation used to evaluate the effectiveness of the analysis are documented in Sect. 4. Finally the conclusion is given in Sect. 5.

2 Related Works

Initially we need to understand how GitHub works and what features it offers. As a result, we referred to a number of research studies conducted on GitHub with various objectives, where (Joy et al. 2018; Sarwar 2021; Koskela et al. 2018) focused on understanding and analyzing the various parameters of the open-source software GitHub. They were able to identify the key parameters as the number of contributions, programming languages, project size, and project age, and they investigated their significance by developing a conceptual model and experimenting with GitHub data. They were able to confirm that the above parameters have a greater impact on project performance as a result of this.

In paper Radha et al. (2017) proposed the implemented of different centrality measures where the USA road network to attain congestion-free shipment. They analyzed the road network based on the performance indicators and identified ways to reduce traffic issues that could aid in traffic management.

Thangavelu and Jyotishi (2018), Sharma and Mahajan (2017), and Gajanayake et al. (2020) proposed a detailed analysis of GitHub work and its projects. They primarily focused on identifying the characteristics of the firm behind the influence of the performance of the open-source project at different stages. In order to experiment they developed a model with four stages such as issues, forks, pull requests, and releases. Through this, they were able to examine the factors of success of the open-source project at different levels. We were able to gain an understanding of how GitHub works and the role of parameters as a result of these works.

Identification of a key person in the network using the well-known centrality measures such as degree, betweenness, closeness, katz, and Eigenvector centrality was proposed by Landher et al. (2010) and Yadav et al. (2019). They were able to conclude that Katz centrality is better for the network they chose based on the distance between nodes, the number of shortest paths, and ranking by implementing the above measure.

Farooq et al. (2018) implemented the above measures as well as and also the coefficient clustering and page rank to a covert network and were able to visualize the metrics that assisted them in identifying the most influential node. These two works served as a foundation for our research because they provided insight into how to implement those measures in our network.

Liu et al. (2019) proposed a method for predicting links between paper citations based on keywords, times, and author information. A series of experiments were also performed on the real-time dataset. They were able to correlate two papers and create a link in the citation network by assigning a value to a pair that was as accurate as possible. As a result, the dataset validated the feasibility of their proposal. We got an idea for implementing the link prediction to our dataset from this work.

Furthermore, to gain a better understanding of other methods for determining the social network's most significant node, we referred to the work of Ramya and Bagavathi Sivakumar (2021), who demonstrated that DC-FNN classification outperforms fixed clustering and NLP-based sentiment analysis in identifying the influential node in the Twitter dataset based on categories such as pricing, service, and timeliness. They tested these approaches on the likelihood function using iterative logistic regression analysis and the temporal influential model (TIM), and the results show that they are more accurate than other methods in terms of precision, recall, and f-measure. We gained an understanding of the topic's role in a node's influence level as a result of this research.

3 Methodologies of Analysis

3.1 Data Description

We considered the essential GitHub values such as user, followers, following, gists, and numrepos, because we used real-time data from GitHub users. The values were adequate for our analysis, and a more detailed explanation is provided in the following sections.

3.2 Overview of the Work Flow Model

In this diagram, we depicted the flow of methods that assisted us in obtaining the required analysis. There are six divisions in total, with the first two illustrating GitHub access and follower analysis, and the outputs obtained being passed to the other four divisions for detailed analysis. The first division represents the calculation of centrality measures, and the second division represents prediction and follower analysis, which leads to the recommendation of the top 10 followers to the user. Although these two divisions are related, we separated them for clarity. Finally, the fourth division represents the prediction of future links between followers. As needed for our research, the output from each division is considered for analysis in other divisions (Fig. 1).

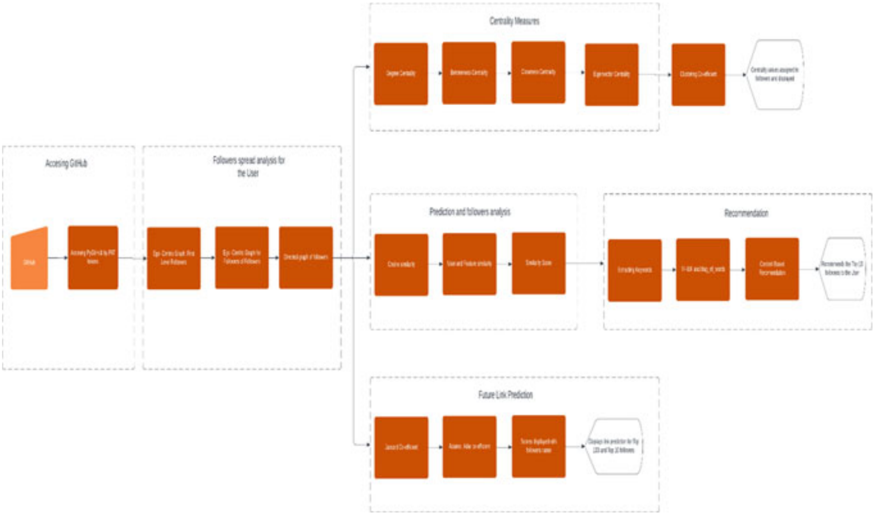

Centrality Measures
Degree Centrality
Betweenness Centrality
Closeness Centrality
Eigenvector Centrality
Clustering Co-efficient
Centrality values assigned to followers and displayed
Accessing GitHub
GitHub
Accessing PyGithub by PAT tokens
Followers spread analysis for the User
Ego-Centric Graph First Level Followers
Ego-Centric Graph for Followers of Followers
Directed graph of followers
Prediction and followers analysis
Cosine similarity
User and Feature similarity
Similarity Score
Recommendation
Extracting Keywords
TF-IDF and bag_of_words
Content Based Recommendation
Recommends the Top 10 followers to the User
Future Link Prediction
Jaccard Co-efficient
Adamic Adar co-efficient
Scores displayed with followers name
Displays the prediction for Top 100 and Top 10 followers

Fig. 1 Overview of the work flow model

3.3 Centrality Measures

Our methods consist of different stages where the results obtained from each stage are used in other stages to obtain an efficient output. Initially, we explored and understood the spread of followers for a particular user or developer by establishing a connection to the GitHub API (PyGitHub) using Personal Access Token, and then the username of a particular user (here the user is 'Anshul Mehta') was given to get the ego-centric graph of the first level of followers in Fig. 2. For better understanding we explored the followers of the followers and plotted the directed graph.

As our first objective, we implemented centrality measures to identify the prominent follower in the network. Basically, centrality is used in a big network to estimate an individual's properties. The degree of interpersonal interaction and communication within a social network is also measured by centrality. More connections between network members result in greater centrality and always the community development in any online social network is mainly driven by individual nodes having high centrality.

3.3.1 Degree Centrality Measure

Degree centrality (DC) is a metric of exposure that evaluates a node's network connectedness using the number of direct contacts to the node.

$$Dv = aiv^{n}i = 1 \tag{1}$$

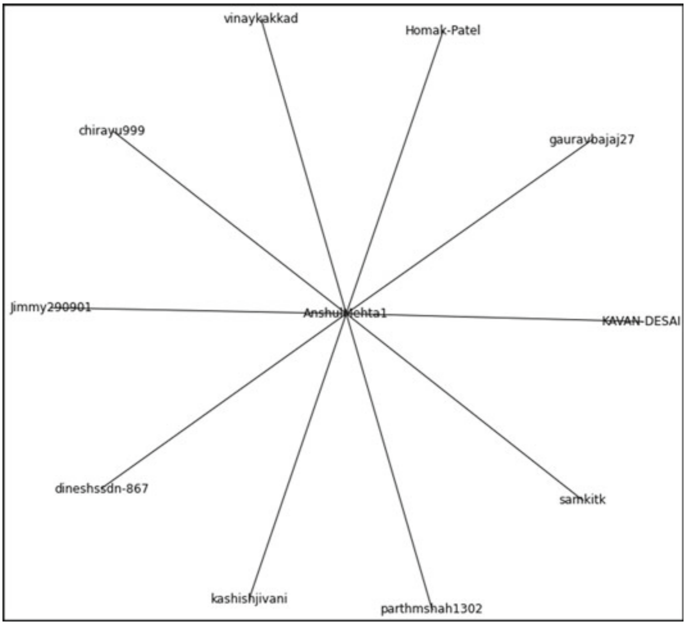


Fig. 2 Followers graph of Anshul Mehta (developer)

In Eq. (1), the adjacency matrix *aiv* was used to determine the degree centrality of node *v*.

3.3.2 Betweenness Centrality Measure

Betweenness centrality (BC) measures the length of time a user or node acts as a bridge between two nodes in a sizable community network while taking into account the quickest path.

$$C_B v = \frac{\sigma_{st}(v)}{\sigma_{st}} \quad s \neq v \neq t \in V \tag{2}$$

In Eq. (2), the terms σ_{st} and $\sigma_{st}(v)$ stand for the total number of shortest routes from node *s* to node *t* and the number of paths that passes through *v,* respectively.

3.3.3 Closeness Centrality Measure

Closeness centrality (CC) is a measure of how long it takes for information to go from one node to every other node in a network and it is also the length of the shortest path between any two nodes in a complicated network graph.

$$Cx = \frac{1}{y^{d(y,x)}} \tag{3}$$

In Eq. (3), $d(x, y)$ is the measured distance between the x and y.

3.3.4 Eigenvector Centrality Measure

Eigenvector centrality (EC) is used to assess a node's effect over other nodes in the network. A node's influence will be stronger if it is connected to significant nodes, but its impact on other nodes will be low if it is connected to less important nodes. So, the significance of the node's neighbours determines its relevance

$$\mathrm{EC}_v = v_x = \frac{1}{\tau_{\max}(A)^{j=^n_1 a_{jx} v_j}} \tag{4}$$

In Eq. (4), $v = (v1, 2, \ldots)$ is the eigenvector for the highest eigenvalue $\tau_{\max}(A)$ of the adjacent matrix A.

3.4 Similarity Measures

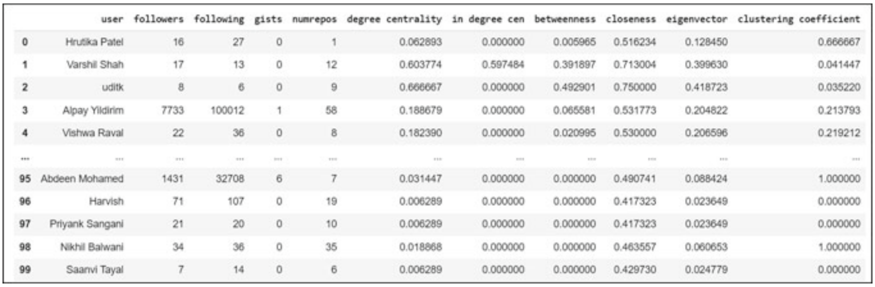
In addition to that coefficient clustering was calculated to identify how closely the followers were present in a network. The obtained values of each follower were stored as a data frame with basic values of a GitHub such as the number of repositories, gists, number of followers, and number of following to get an idea of data spread and analyze the influential parameters.

As a part of the next research, we implemented the link prediction using the cosine distance for the obtained network. Link prediction is used to find a collection of missing or future linkages between the users by forecasting and estimating the likelihood of each non-existing link in the network.

$$\cos(x, y) = \frac{x.y}{||x|| * ||y||} \tag{5}$$

In Eq. (5), the terms $x.y$ and $||x||$ and $||y||$ are the dot product and the length of the vectors x and y, whereas the cross product of the vectors x and y is $||x|| * ||y||$.

Whereas, the cosine of the angle created by the two vectors is used to calculate how similar two users or objects are when they are treated as two vectors in m-dimensional space. During filtering, it is often used to compare two documents, and subsequently more and more regularly to compare two individuals or other aspects instead of simply documents. Since it is often utilized in positive space, it determines how close 0 and 1 are in value.



	user	followers	following	gists	numrepos	degree centrality	in degree cen	betweenness	closeness	eigenvector	clustering coefficient
0	Hrutika Patel	16	27	0	1	0.062893	0.000000	0.005965	0.516234	0.128450	0.666667
1	Varshil Shah	17	13	0	12	0.603774	0.597484	0.391897	0.713004	0.399630	0.041447
2	uditk	8	6	0	9	0.666667	0.000000	0.492901	0.750000	0.418723	0.035220
3	Alpay Yildirim	7733	100012	1	58	0.188679	0.000000	0.065581	0.531773	0.204822	0.213793
4	Vishwa Raval	22	36	0	8	0.182390	0.000000	0.020995	0.530000	0.206596	0.219212
...	...	...	...	...	...	...	...	...	...	...	...
95	Abdeen Mohamed	1431	32708	6	7	0.031447	0.000000	0.000000	0.490741	0.088424	1.000000
96	Harvish	71	107	0	19	0.006289	0.000000	0.000000	0.417323	0.023649	0.000000
97	Priyank Sangani	21	20	0	10	0.006289	0.000000	0.000000	0.417323	0.023649	0.000000
98	Nikhil Balwani	34	36	0	35	0.018868	0.000000	0.000000	0.463557	0.060653	1.000000
99	Saanvi Tayal	7	14	0	6	0.006289	0.000000	0.000000	0.429730	0.024779	0.000000

Fig. 3 Data frame created from the calculated values

Then to get a clear idea of similarities among the followers, we employed the follower similarity which identifies the similarities between the followers based on ratings given by them and also found the feature similarities in which the similar features among the followers were evaluated. The data frame consisting of these values was formed in Fig. 3. In addition, the pairwise cosine was found to get a clear idea of similarities between the followers. By using these data, similar followers were found based on the aggregate similarity score (based on the number of gists, repositories, followers, and following) with a developer.

As we were able to identify similar followers now, we focused on finding the potential followers to recommend the user based on the content filtration technique. Here the keywords of the repositories are considered as content. The reason for the content-based recommendation is that it recommends based on the user preferences and this would help us in recommending the followers to the developer.

To implement the recommendation, we initially found the keywords of repositories by stage-wise like name, description, and language. Here comes the natural language processing (NLP) where the strings were converted to lowercase and by stemming, we found the keywords. This was done in two approaches such as bag of words and TF-IDF.

$$\text{TF-IDF} = \text{TF}(t, d) + \text{IDF}(t) \tag{6}$$

In Eq. (6), TF(t, d) is the term frequency of term 't' that appears in a document 'd' and IDF(t) is the inverse document frequency.

Bag of words is an NLP model used to count the frequency of each word in a corpus for that document, whereas TF-IDF is used to quantify the importance of the relevance of string representation in documents among the collection of documents. As we obtained the words, now again the cosine distance was checked in a bag of words and TF-IDF to find the similarity. These words were then stored and sorted with the corresponding indexes. Now we developed a recommender where the index fetched from the similarity array and sorted the distance in descending order to get the top 10 followers for both bag of words and TF-IDF.

As we found the followers for the user now, we identified the list of followers who can be recommended to the user. In addition, this was done to all the first-level followers of the developer to check whether our method worked properly or not. From this, we obtained the end result of recommending the followers to the user working on similar domains.

Finally, the future prediction between the user's followers was calculated using the Jaccard coefficient and Adamic Adar coefficient, and the results were visualized to get an idea of links. We discovered the network's common triads to gain a better understanding of the links and influence of followers.

4 Experimental Results

This section carries out a thorough experimental and quantitative study of the analysis undertaken and the results were shown with an explanation. Initially, the graph was obtained and then the centrality measures were implemented on the follower's network in Figs. 4 and 5 depicts the centrality of common neighbours.

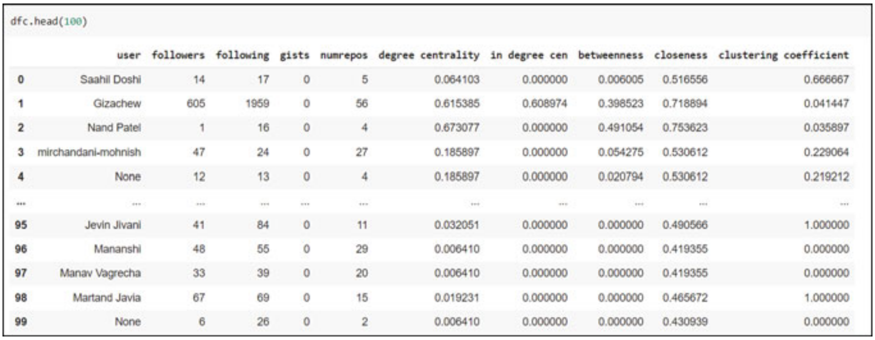
Further, the link prediction between the followers was identified using cosine distance and values for the corresponding followers were stored in the data frame as in Fig. 6.

The cosine similarity was calculated between the followers and the features of the data frame which helped us to get an idea of the relations between the followers as in Fig. 7.

The above values were then added to the data frame as in Figs. 8 and 9.

Based on these values the similar feature of the user was calculated by considering the ratings given and it was mapped to the gists as a similar score. So, when we give the gists number the corresponding features get displayed as in Fig. 10.

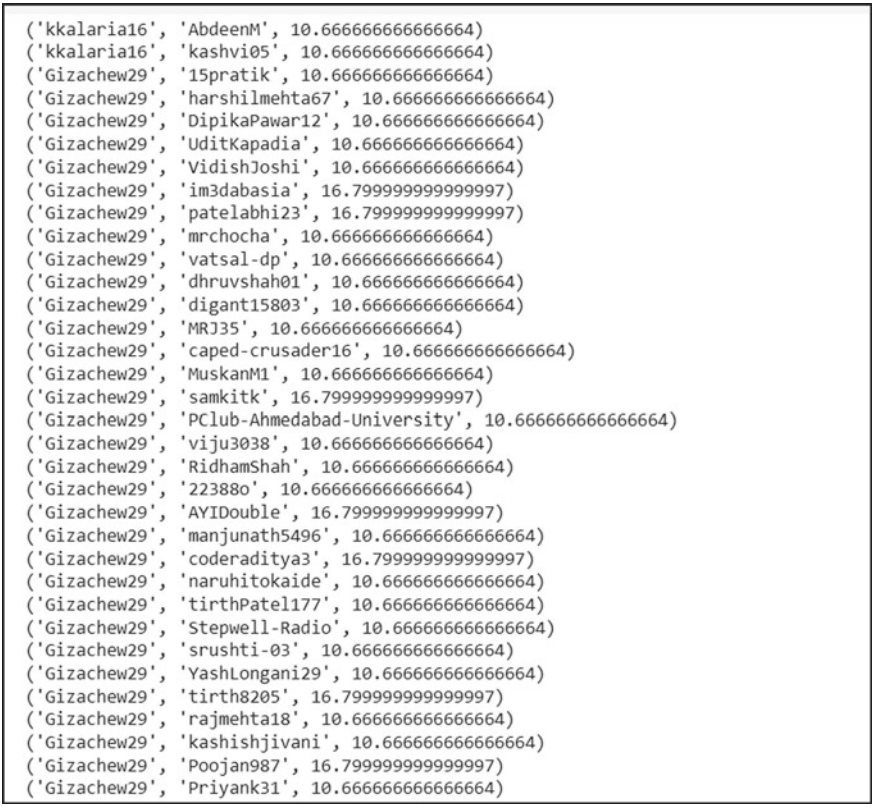
Based on the score of similar features, the aggregate score was calculated which helped us to find similar followers to the user as in Fig. 11.



```
dfc.head(100)
```

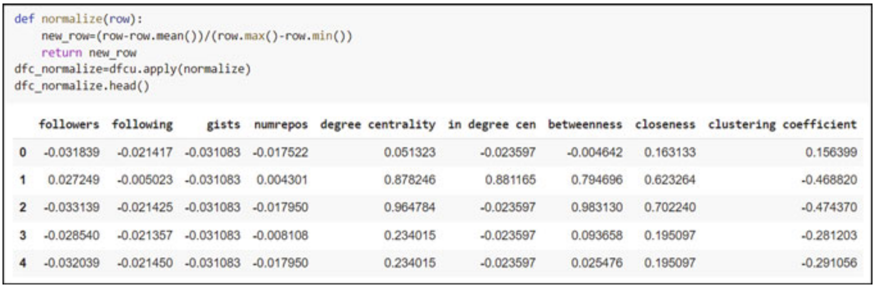
	user	followers	following	gists	numrepos	degree centrality	in degree cen	betweenness	closeness	clustering coefficient
0	Saahil Doshi	14	17	0	5	0.064103	0.000000	0.006005	0.516556	0.666667
1	Gizachew	605	1959	0	56	0.615385	0.608974	0.398523	0.718894	0.041447
2	Nand Patel	1	16	0	4	0.673077	0.000000	0.491054	0.753623	0.035897
3	mirchandani-mohnish	47	24	0	27	0.185897	0.000000	0.054275	0.530612	0.229064
4	None	12	13	0	4	0.185897	0.000000	0.020794	0.530612	0.219212
...	...	...	...	...	...	...	...	...	...	...
95	Jevin Jivani	41	84	0	11	0.032051	0.000000	0.000000	0.490566	1.000000
96	Mananshi	48	55	0	29	0.006410	0.000000	0.000000	0.419355	0.000000
97	Manav Vagrecha	33	39	0	20	0.006410	0.000000	0.000000	0.419355	0.000000
98	Martand Javia	67	69	0	15	0.019231	0.000000	0.000000	0.465672	1.000000
99	None	6	26	0	2	0.006410	0.000000	0.000000	0.430939	0.000000

Fig. 4 Values obtained from the centrality measures



```
('kkalaria16', 'AbdeenM', 10.666666666666664)
('kkalaria16', 'kashvi05', 10.666666666666664)
('Gizachew29', '15pratik', 10.666666666666664)
('Gizachew29', 'harshilmehta67', 10.666666666666664)
('Gizachew29', 'DipikaPawar12', 10.666666666666664)
('Gizachew29', 'UditKapadia', 10.666666666666664)
('Gizachew29', 'VidishJoshi', 10.666666666666664)
('Gizachew29', 'im3dabasia', 16.799999999999997)
('Gizachew29', 'patelabhi23', 16.799999999999997)
('Gizachew29', 'mrchocha', 10.666666666666664)
('Gizachew29', 'vatsal-dp', 10.666666666666664)
('Gizachew29', 'dhruvshah01', 10.666666666666664)
('Gizachew29', 'digant15803', 10.666666666666664)
('Gizachew29', 'MRJ35', 10.666666666666664)
('Gizachew29', 'caped-crusader16', 10.666666666666664)
('Gizachew29', 'MuskanM1', 10.666666666666664)
('Gizachew29', 'samkitk', 16.799999999999997)
('Gizachew29', 'PClub-Ahmedabad-University', 10.666666666666664)
('Gizachew29', 'viju3038', 10.666666666666664)
('Gizachew29', 'RidhamShah', 10.666666666666664)
('Gizachew29', '22388o', 10.666666666666664)
('Gizachew29', 'AYIDouble', 16.799999999999997)
('Gizachew29', 'manjunath5496', 10.666666666666664)
('Gizachew29', 'coderaditya3', 16.799999999999997)
('Gizachew29', 'naruhitokaide', 10.666666666666664)
('Gizachew29', 'tirthPatel177', 10.666666666666664)
('Gizachew29', 'Stepwell-Radio', 10.666666666666664)
('Gizachew29', 'srushti-03', 10.666666666666664)
('Gizachew29', 'YashLongani29', 10.666666666666664)
('Gizachew29', 'tirth8205', 16.799999999999997)
('Gizachew29', 'rajmehta18', 10.666666666666664)
('Gizachew29', 'kashishjivani', 10.666666666666664)
('Gizachew29', 'Poojan987', 16.799999999999997)
('Gizachew29', 'Priyank31', 10.666666666666664)
```

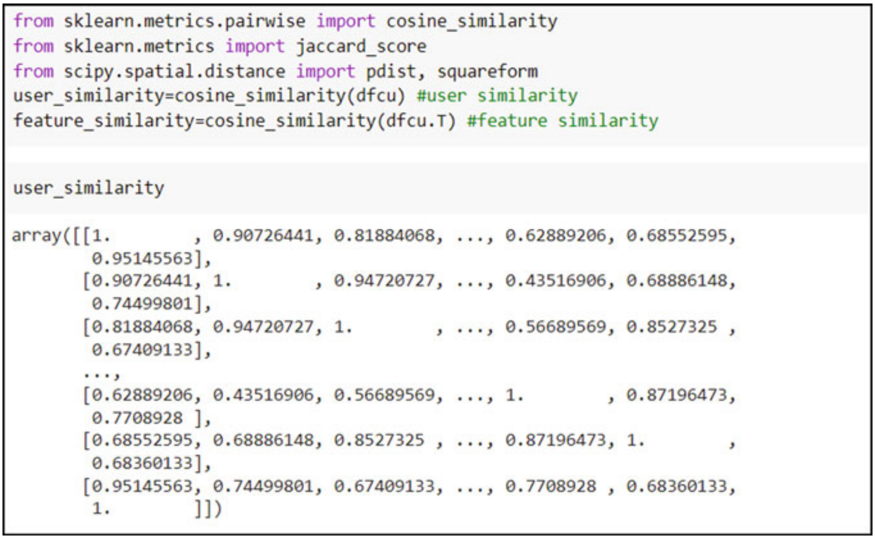
Fig. 5 Common neighbours centrality measure



```
def normalize(row):
    new_row=(row-row.mean())/(row.max()-row.min())
    return new_row
dfc_normalize=dfcu.apply(normalize)
dfc_normalize.head()
```

	followers	following	gists	numrepos	degree centrality	in degree cen	betweenness	closeness	clustering coefficient
0	-0.031839	-0.021417	-0.031083	-0.017522	0.051323	-0.023597	-0.004642	0.163133	0.156399
1	0.027249	-0.005023	-0.031083	0.004301	0.878246	0.881165	0.794696	0.623264	-0.468820
2	-0.033139	-0.021425	-0.031083	-0.017950	0.964784	-0.023597	0.983130	0.702240	-0.474370
3	-0.028540	-0.021357	-0.031083	-0.008108	0.234015	-0.023597	0.093658	0.195097	-0.281203
4	-0.032039	-0.021450	-0.031083	-0.017950	0.234015	-0.023597	0.025476	0.195097	-0.291056

Fig. 6 Values obtained from the cosine distance measure

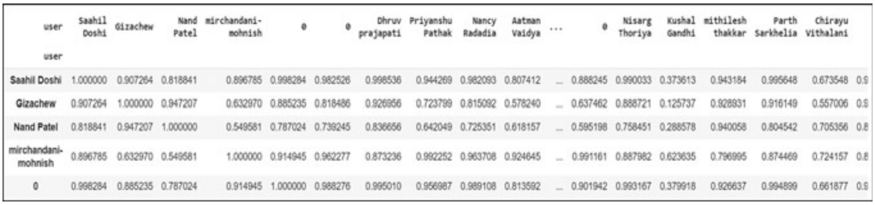


```
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.metrics import jaccard_score
from scipy.spatial.distance import pdist, squareform
user_similarity=cosine_similarity(dfcu) #user similarity
feature_similarity=cosine_similarity(dfcu.T) #feature similarity
```

```
user_similarity
```

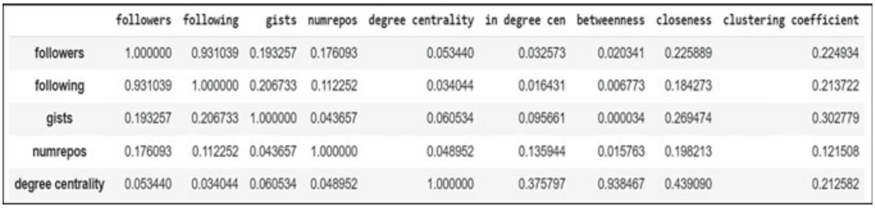
```
array([[1.        , 0.90726441, 0.81884068, ..., 0.62889206, 0.68552595,
        0.95145563],
       [0.90726441, 1.        , 0.94720727, ..., 0.43516906, 0.68886148,
        0.74499801],
       [0.81884068, 0.94720727, 1.        , ..., 0.56689569, 0.8527325 ,
        0.67409133],
       ...,
       [0.62889206, 0.43516906, 0.56689569, ..., 1.        , 0.87196473,
        0.7708928 ],
       [0.68552595, 0.68886148, 0.8527325 , ..., 0.87196473, 1.        ,
        0.68360133],
       [0.95145563, 0.74499801, 0.67409133, ..., 0.7708928 , 0.68360133,
        1.        ]])
```

Fig. 7 Implementation to find user and feature similarity



user	Saahil Doshi	Gizachew	Nand Patel	mirchandani-mohnish	0	0	Dhruv prajapati	Priyanshu Pathak	Nancy Radadia	Aatman Vaidya	...	0	Nisarg Thoriya	Kushal Gandhi	mithilesh thakkar	Parth Sarkhelia	Chirayu Vithalani	
user																		
Saahil Doshi	1.000000	0.907264	0.818841	0.896785	0.998284	0.982526	0.998536	0.944269	0.982093	0.807412	...	0.888245	0.990033	0.373613	0.943184	0.995648	0.673548	0.9
Gizachew	0.907264	1.000000	0.947207	0.632970	0.885235	0.818486	0.926956	0.723799	0.815092	0.578240	...	0.637462	0.888721	0.125737	0.928931	0.916149	0.557006	0.9
Nand Patel	0.818841	0.947207	1.000000	0.549581	0.787024	0.739245	0.836656	0.642049	0.725351	0.618157	...	0.595198	0.758451	0.288578	0.940058	0.804542	0.705356	0.8
mirchandani-mohnish	0.896785	0.632970	0.549581	1.000000	0.914945	0.962277	0.873236	0.992252	0.963708	0.924645	...	0.991161	0.887982	0.623635	0.796995	0.874469	0.724157	0.8
0	0.998284	0.885235	0.787024	0.914945	1.000000	0.988276	0.995010	0.956987	0.989108	0.813592	...	0.901942	0.993167	0.379918	0.926637	0.994899	0.661877	0.9

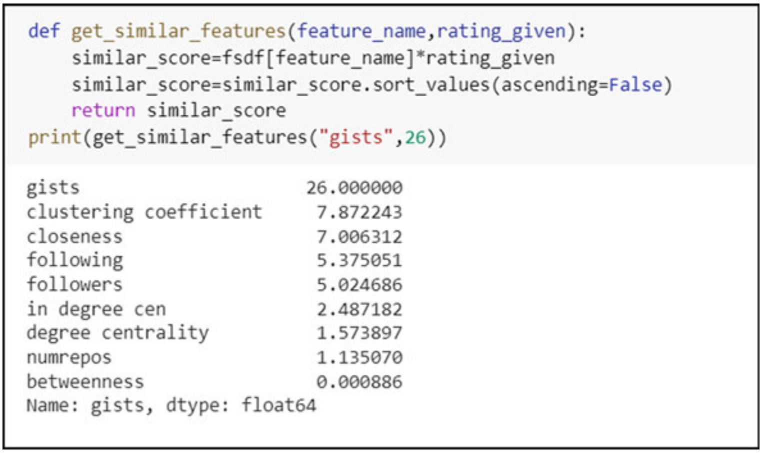
Fig. 8 Values obtained from the user similarity



	followers	following	gists	numrepos	degree centrality	in degree cen	betweenness	closeness	clustering coefficient
followers	1.000000	0.931039	0.193257	0.176093	0.053440	0.032573	0.020341	0.225889	0.224934
following	0.931039	1.000000	0.206733	0.112252	0.034044	0.016431	0.006773	0.184273	0.213722
gists	0.193257	0.206733	1.000000	0.043657	0.060534	0.095661	0.000034	0.269474	0.302779
numrepos	0.176093	0.112252	0.043657	1.000000	0.048952	0.135944	0.015763	0.198213	0.121508
degree centrality	0.053440	0.034044	0.060534	0.048952	1.000000	0.375797	0.938467	0.439090	0.212582

Fig. 9 Values obtained from the feature similarity

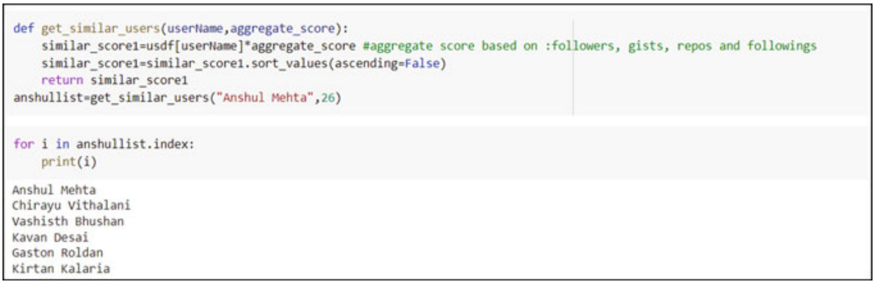
As a next step for recommending the followers, we used the content-based filtering approach where the keywords of repositories, such as name, description, and language were extracted and added to the data frame by converting strings to lowercase and stemming as in Fig. 12.



```
def get_similar_features(feature_name,rating_given):
    similar_score=fsdf[feature_name]*rating_given
    similar_score=similar_score.sort_values(ascending=False)
    return similar_score
print(get_similar_features("gists",26))
```

```
gists                     26.000000
clustering coefficient     7.872243
closeness                  7.006312
following                  5.375051
followers                  5.024686
in degree cen              2.487182
degree centrality          1.573897
numrepos                   1.135070
betweenness                0.000886
Name: gists, dtype: float64
```

Fig. 10 Similar feature of gists 26

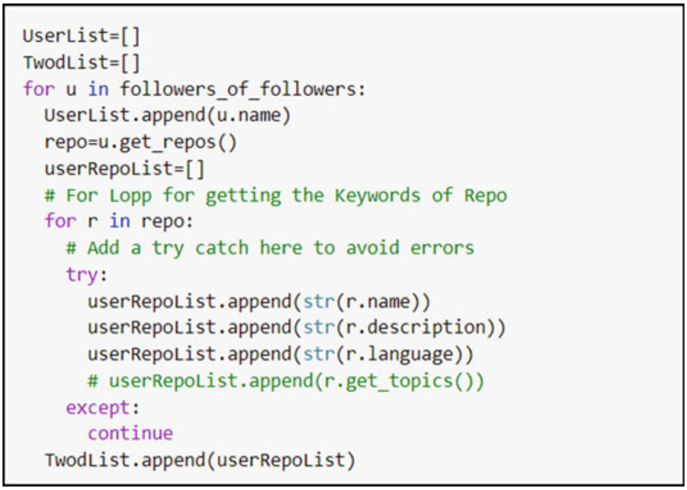


```
def get_similar_users(userName,aggregate_score):
    similar_score1=usdf[userName]*aggregate_score #aggregate score based on :followers, gists, repos and followings
    similar_score1=similar_score1.sort_values(ascending=False)
    return similar_score1
anshullist=get_similar_users("Anshul Mehta",26)
```

```
for i in anshullist.index:
    print(i)
```

```
Anshul Mehta
Chirayu Vithalani
Vashisth Bhushan
Kavan Desai
Gaston Roldan
Kirtan Kalaria
```

Fig. 11 Found similar followers for user Anshul Mehta

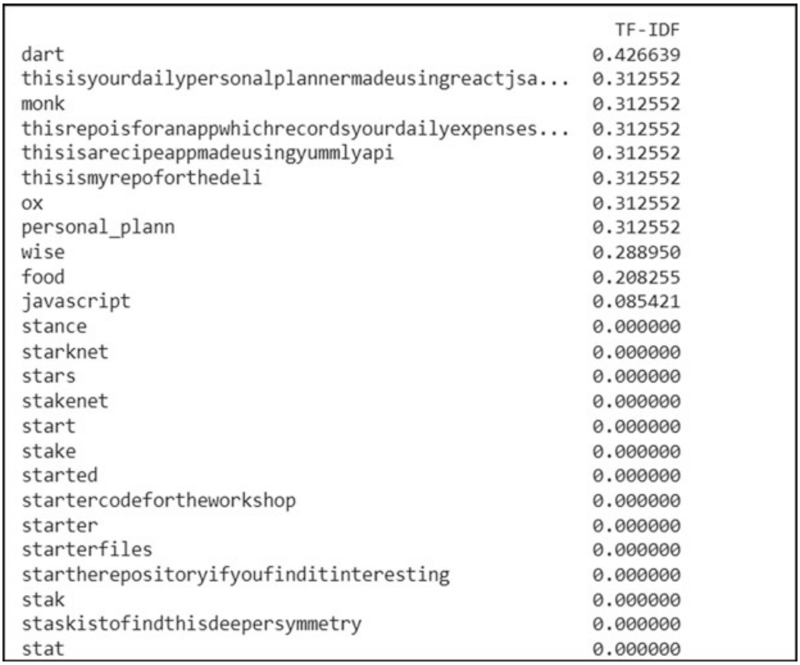


```
UserList=[]
TwodList=[]
for u in followers_of_followers:
  UserList.append(u.name)
  repo=u.get_repos()
  userRepoList=[]
  # For Lopp for getting the Keywords of Repo
  for r in repo:
    # Add a try catch here to avoid errors
    try:
      userRepoList.append(str(r.name))
      userRepoList.append(str(r.description))
      userRepoList.append(str(r.language))
      # userRepoList.append(r.get_topics())
    except:
      continue
  TwodList.append(userRepoList)
```

Fig. 12 Extracting keywords from repositories

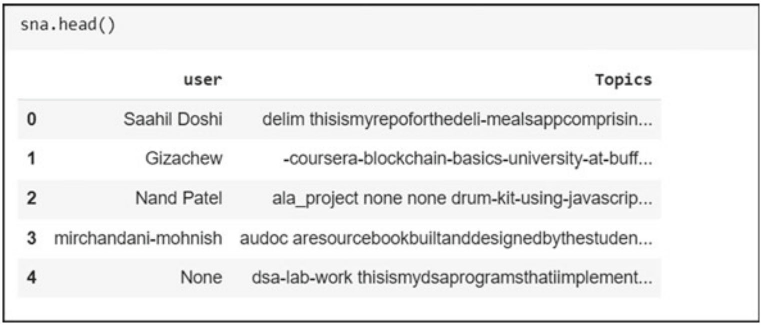
By the method of TF-IDF and bag of words, the frequency of words was found and then cosine similarity was used to identify the similarity between the TF-IDF vectors which was then stored in the sorted list as in Fig. 13.

From the above diagram, we can infer some of the known keywords such as JavaScript, stake, dart, etc. We may thus conclude that extraction was successful since we have now identified the followers who, in light of the list of followers, can be suggested to the user as in Figs. 14 and 15.



```
                                                        TF-IDF
dart                                                  0.426639
thisisyourdailypersonalplannermadeusingreactjsa...    0.312552
monk                                                  0.312552
thisrepoisforanappwhichrecordsyourdailyexpenses...    0.312552
thisisarecipeappmadeusingyummlyapi                    0.312552
thisismyrepoforthedeli                                0.312552
ox                                                    0.312552
personal_plann                                        0.312552
wise                                                  0.288950
food                                                  0.208255
javascript                                            0.085421
stance                                                0.000000
starknet                                              0.000000
stars                                                 0.000000
stakenet                                              0.000000
start                                                 0.000000
stake                                                 0.000000
started                                               0.000000
startercodefortheworkshop                             0.000000
starter                                               0.000000
starterfiles                                          0.000000
starttherepositoryifyoufinditinteresting              0.000000
stak                                                  0.000000
staskistofindthisdeepersymmetry                       0.000000
stat                                                  0.000000
```

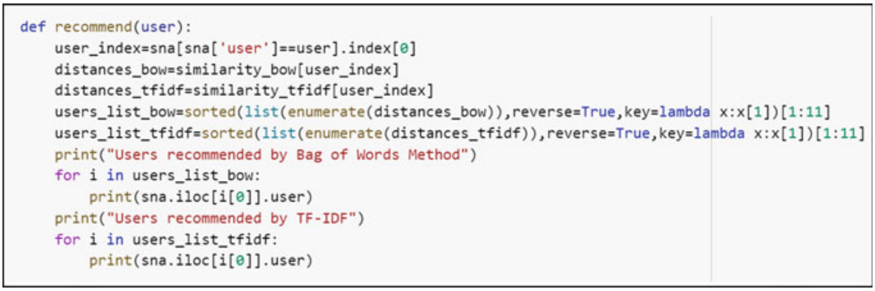
Fig. 13 Frequency of keywords extracted from the TF-IDF



```
sna.head()
```

	user	Topics
0	Saahil Doshi	delim thisismyrepoforthedeli-mealsappcomprisin...
1	Gizachew	-coursera-blockchain-basics-university-at-buff...
2	Nand Patel	ala_project none none drum-kit-using-javascrip...
3	mirchandani-mohnish	audoc aresourcebookbuiltanddesignedbythestuden...
4	None	dsa-lab-work thisismydsaprogramsthatiimplement...

Fig. 14 Keywords obtained were stored to the corresponding user



```
def recommend(user):
    user_index=sna[sna['user']==user].index[0]
    distances_bow=similarity_bow[user_index]
    distances_tfidf=similarity_tfidf[user_index]
    users_list_bow=sorted(list(enumerate(distances_bow)),reverse=True,key=lambda x:x[1])[1:11]
    users_list_tfidf=sorted(list(enumerate(distances_tfidf)),reverse=True,key=lambda x:x[1])[1:11]
    print("Users recommended by Bag of Words Method")
    for i in users_list_bow:
        print(sna.iloc[i[0]].user)
    print("Users recommended by TF-IDF")
    for i in users_list_tfidf:
        print(sna.iloc[i[0]].user)
```

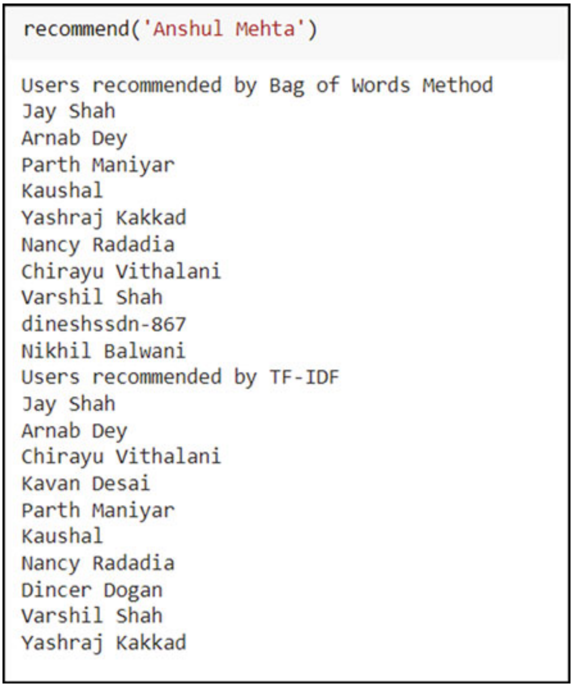
Fig. 15 Finding the top 10 users from both the methods

The follower's list closer to the user is printed from the data frame where we can see some followers are the same in both the methods. Figure 16 shows the list of top 10 followers from both the methods.

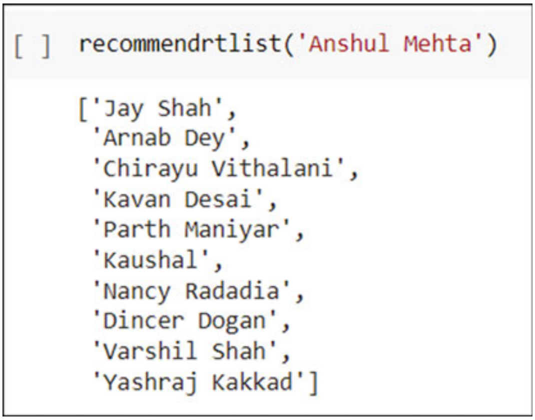
At last, the objective of our research is to recommend the top 10 followers to the user based on the working of followers in similar technologies or domains. So, the recommendation is done as in Fig. 17.

Additionally, in order to identify the future link prediction between the followers we find Jaccard coefficient and Adamic Adar coefficient measures and also visualized

Fig. 16 List of top 10 followers from both methods



```
recommend('Anshul Mehta')

Users recommended by Bag of Words Method
Jay Shah
Arnab Dey
Parth Maniyar
Kaushal
Yashraj Kakkad
Nancy Radadia
Chirayu Vithalani
Varshil Shah
dineshssdn-867
Nikhil Balwani
Users recommended by TF-IDF
Jay Shah
Arnab Dey
Chirayu Vithalani
Kavan Desai
Parth Maniyar
Kaushal
Nancy Radadia
Dincer Dogan
Varshil Shah
Yashraj Kakkad
```



```
[ ]  recommendrtlist('Anshul Mehta')

     ['Jay Shah',
      'Arnab Dey',
      'Chirayu Vithalani',
      'Kavan Desai',
      'Parth Maniyar',
      'Kaushal',
      'Nancy Radadia',
      'Dincer Dogan',
      'Varshil Shah',
      'Yashraj Kakkad']
```

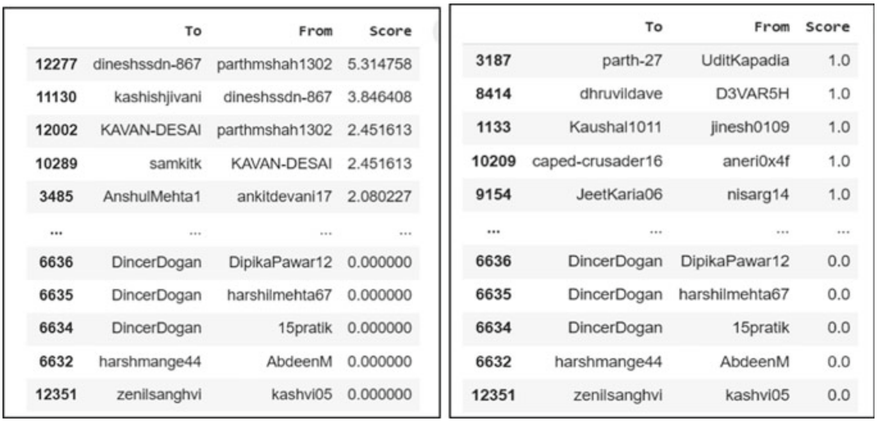
Fig. 17 List of top 10 followers recommended to the developer

the prediction for the top 100 and top 10 followers. Figure 18 depicts scores of Adamic Adar and Jaccard coefficient.

As per the scores, the graph was plotted where the lines between followers shows the future link predictions for the top 100 and top 10 followers. Figures 19 and 20 depict visualization for Adamic Adar and Jaccard coefficient for top 10 followers.

From these graphs, we could be able to infer that there are high chances of future links between the followers of users. As the outliers are the same in these graphs, all graphs are nearly giving the same prediction of links. Figure 21 depicts the scores between the followers.

The outer clusters remain the same and all the inner clusters are concentrated for a few users forming the bigger circle. On basis of scores obtained from the Jaccard coefficient and Adamic Adar coefficient, the top 10 most predicted links between the user and his followers were evaluated.



	To	From	Score
12277	dineshssdn-867	parthmshah1302	5.314758
11130	kashishjivani	dineshssdn-867	3.846408
12002	KAVAN-DESAI	parthmshah1302	2.451613
10289	samkitk	KAVAN-DESAI	2.451613
3485	AnshulMehta1	ankitdevani17	2.080227
...	...	...	...
6636	DincerDogan	DipikaPawar12	0.000000
6635	DincerDogan	harshilmehta67	0.000000
6634	DincerDogan	15pratik	0.000000
6632	harshmange44	AbdeenM	0.000000
12351	zenilsanghvi	kashvi05	0.000000

	To	From	Score
3187	parth-27	UditKapadia	1.0
8414	dhruvildave	D3VAR5H	1.0
1133	Kaushal1011	jinesh0109	1.0
10209	caped-crusader16	aneri0x4f	1.0
9154	JeetKaria06	nisarg14	1.0
...	...	...	...
6636	DincerDogan	DipikaPawar12	0.0
6635	DincerDogan	harshilmehta67	0.0
6634	DincerDogan	15pratik	0.0
6632	harshmange44	AbdeenM	0.0
12351	zenilsanghvi	kashvi05	0.0

Fig. 18 Scores of Adamic Adar and Jaccard coefficient

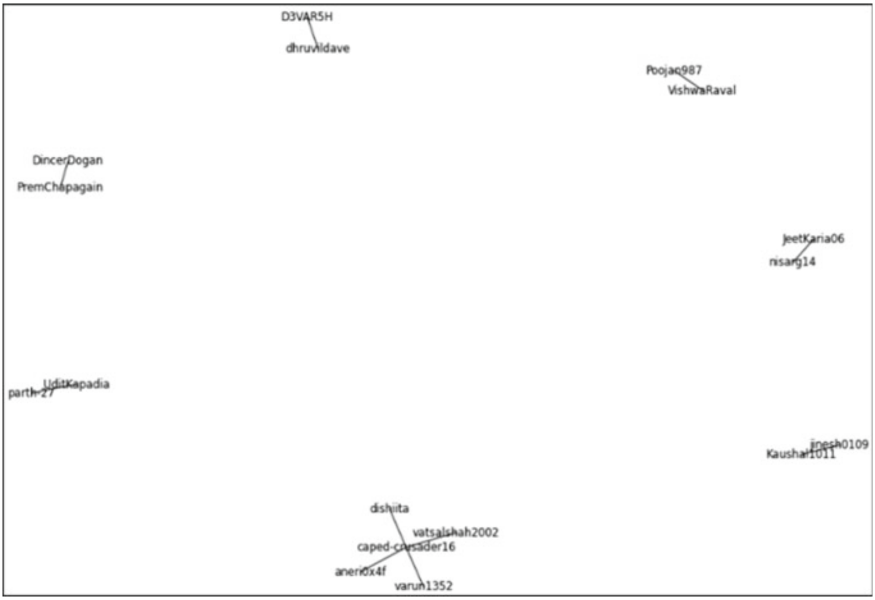


Fig. 19 Adamic Adar coefficient for top 10 followers

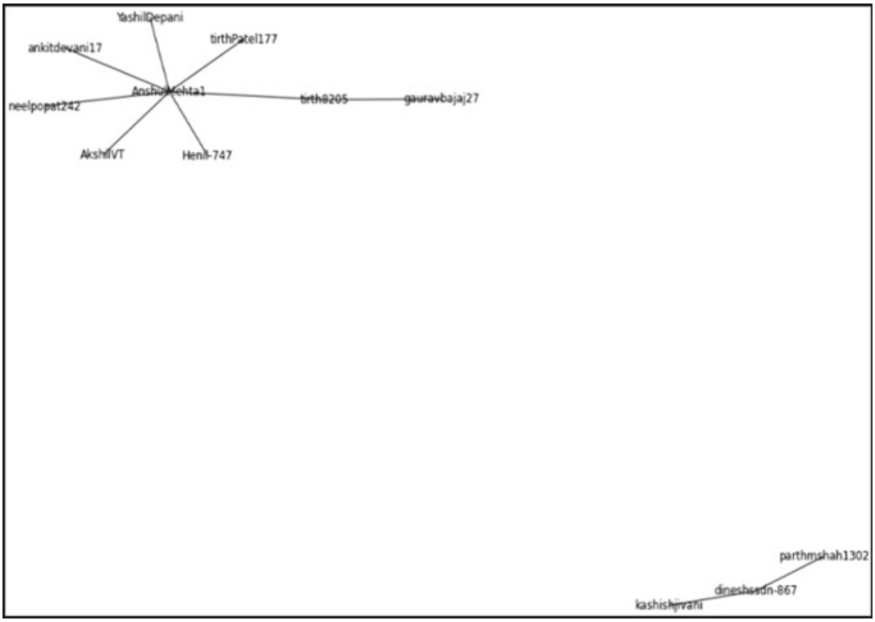
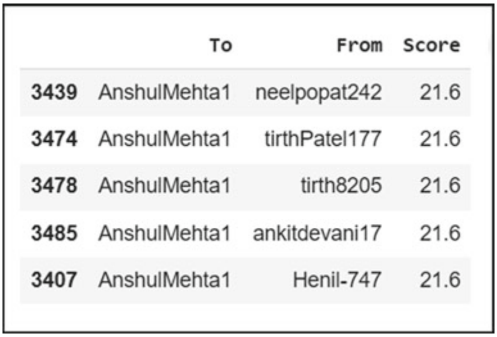


Fig. 20 Jaccard coefficient for top 10 followers



	To	From	Score
3439	AnshulMehta1	neelpopat242	21.6
3474	AnshulMehta1	tirthPatel177	21.6
3478	AnshulMehta1	tirth8205	21.6
3485	AnshulMehta1	ankitdevani17	21.6
3407	AnshulMehta1	Henil-747	21.6

Fig. 21 Scores between the followers

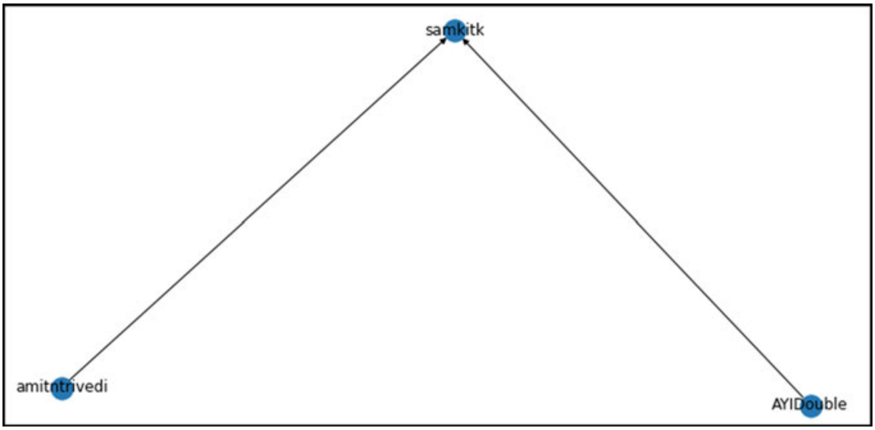


Fig. 22 Triad of developer and followers

The common triads in the network were found to get an idea of links and influence followers. Triads are a set of three actors and the smallest community possible in a network. Through this, we were able to identify that user and closest follower present in all triads as in Fig. 22.

5 Conclusion

Every day, more people join GitHub, but identifying users working in a similar domain is critical for developers so that they can collaborate and work together. We discussed the recommendation system designed specifically for the user and their followers in this paper. We were able to achieve our goals by employing approaches such as centrality measures, link prediction with standard algorithms, and content-based recommendations based on the developer's real-time data. The results of the tests and a thorough quantitative analysis indicate that the metrics used are effective in suggesting notable followers to the network user.

References

Farooq A, Joyia GJ, Uzair M, Akram U (2018, Mar) Detection of influential nodes using social networks analysis based on network metrics. In: 2018 International conference on computing, mathematics and engineering technologies (iCoMET). IEEE, pp 1–6

Gajanayake RGUS, Hiras MHM, Gunathunga PIN, Supun EJ, Karunasenna A, Bandara P (2020, Dec) Candidate selection for the interview using GitHub profile and user analysis for the position of software engineer. In: 2020 2nd International conference on advancements in computing (ICAC), vol 1. IEEE, pp 168–173

Joy A, Thangavelu S, Jyotishi A (2018, Feb) Performance of GitHub open-source software project: an empirical analysis. In: 2018 Second international conference on advances in electronics, computers and communications (ICAECC). IEEE, pp 1–6

Koskela M, Simola I, Stefanidis K (2018, Sept) Open-source software recommendations using GitHub. In: International conference on theory and practice of digital libraries. Springer, Cham, pp 279–285

Landherr A, Friedl B, Heidemann J (2010) A critical review of centrality measures in social networks. Bus Inf Syst Eng 2(6):371–385

Liu H, Kou H, Yan C, Qi L (2019) Link prediction in paper citation network to construct paper correlation graph. EURASIP J Wirel Commun Netw 2019(1):1–12

Radha D, Kavikuil K, Keerthi R (2017, Dec) Centrality measures to analyze transport network for congestion free shipment. In: 2017 2nd International conference on computational systems and information technology for sustainable solution (CSITSS). IEEE, pp 1–5

Ramya GR, Bagavathi Sivakumar P (2021) An incremental learning temporal influence model for identifying topical influencers on Twitter dataset. Soc Netw Anal Min 11(1):1–16

Sarwar MU (2021) Recommending whom to follow on GitHub. Doctoral dissertation, North Dakota State University

Sharma S, Mahajan A (2017) Suggestive approaches to create a recommender system for GitHub. Int J Inf Technol Comput Sci 9(8):48–55

Thangavelu S, Jyotishi A (2018, June) Determinants of open source software project performance: a stage-wise analysis of GitHub projects. In: Proceedings of the 2018 ACM SIGMIS conference on computers and people research, pp 41–42

Yadav AK, Johari R, Dahiya R (2019, Oct) Identification of centrality measures in social network using network science. In: 2019 International conference on computing, communication, and intelligent systems (ICCCIS). IEEE, pp 229–234

Social Network Hashtag Analysis for the 75th Year of India's Independence

A. Veeramanohar, A. J. Nishanth, S. Vishvajit, and G. R. Ramya

1 Introduction

Twitter is a free social networking platform which essentially aims to connecting people. It allows users to publish posts known as tweets. Users can include text, videos, photos or links in the tweets (Nair et al. 2015). Twitter is widely used by people to get the latest news and events, to communicate with fellow users; and follow world leaders, politicians, sports personalities and celebrities (Ramya and Bagavathi Sivakumar 2021).

This notable fast spread of news in twitter is because of the grouping of topics. Topics are grouped using hashtags (#). Twitter came up with the hashtag with the purpose of categorizing discussions and online exchanges which helps in gathering activity that spans globally. These hashtags play an important role in Twitter as it helps in finding the relevant content for the user. Hashtags are also used to identify the trending topics that are being discussed currently on twitter. While there are a number of hashtags present, the key is to identify the right ones so that it matches the content and reaches more number of people (Otte and Rousseau 2002).

This study involved extracting tweets containing the official hashtag #HarGharTiranga of the 75th Indian Independence campaign. "Har Ghar Tiranga" is an event which is a part of the campaign Azadi Ka Amrit Mahotsav, inviting people to take a Tiranga home and lift it to celebrate India's 75th year of independence. The idea behind this drive is to instil patriotism in people's minds and promote awareness of the Indian flag.

A. Veeramanohar · A. J. Nishanth
Department of Electrical and Electronics Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India

S. Vishvajit · G. R. Ramya (✉)
Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Coimbatore, India
e-mail: cb.en.u4eee19145@cb.students.amrita.edu

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_11

This study employs social network analysis based on around 12,000+ Twitter users and above 35,000 tweets of Twitter. All these tweets were posted on the occasion of the 75th year of Indian Independence. The dataset obtained was analyzed for several social network parameters like degree, assortativity, centrality measures like betweenness centrality, closeness centrality, eigenvector centrality and sentiment analysis (Broniatowski et al. 2014).

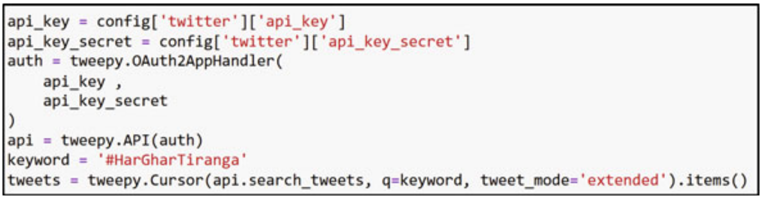
The paper has four sections where Sect. 2 describes the related works and emphasizes the key characteristics used by us to implement the various measures of social network analysis. Section 3 contains a detailed explanation of the methodology analysis used to assess effectiveness. Section 4 concludes with a conclusion.

2 Related Works

Various extensive research papers have been on the highlights of the proposed system that includes social network analysis based on Twitter posts. The paper (Yum 2020) describes the analysis of tweets during COVID-19. Study finds out that Donal J Trump is the central node that connects all other nodes (nodes are twitter account) since he was the President of United States. Centrality measures done on the dataset showed the same concluding that people followed him most during COVID-19 and he was most influential as he held the power and lead the country.

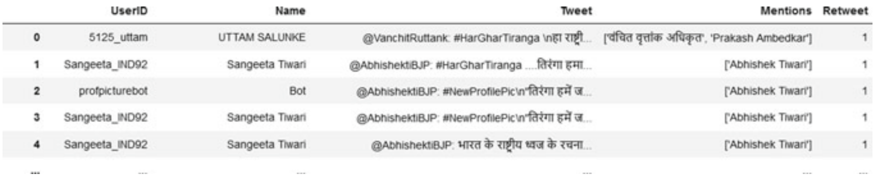
Ramya and Bagavathi Sivakumar (2021) uses sentiment analysis to identify influencers on twitter. It describes what a sentiment analysis is and who all are an influential user. Twitter data was preprocessed by prompting weighted partition around medoids (WPAM) along with artificial cooperative search (WPAM-ACS) which helped in extracting topics from Twitter data through dynamic clustering (DC). Data contained the tweets for which polarities were checked. This was done using deep learning. Natural language processing (NLP) was supposed to be used but it pointed out that tweets were in user's own language for which NLP was difficult. A set of measures were taken and for each tweet, they were calculated and in the end a graph was produced in which highly influential users were found.

During a time of crisis, or during election times, twitter is faster in spreading news than news channels. Nair et al. (2015) tell us how twitter was used during the 2015 floods in Chennai. Social media plays a vital role in the spread of information during disasters. Nair et al. (2015) contain a study regarding how the people of Chennai, Tamil Nadu used twitter in response to the one of the country's worst floods ever. The tweets collected were examined using machine learning algorithms such as Naive Bayes, random forests and decision tree. The performance of all the three algorithms were compared and it was found that random forest was the best algorithm that can be relied on, during a disaster.



```
api_key = config['twitter']['api_key']
api_key_secret = config['twitter']['api_key_secret']
auth = tweepy.OAuth2AppHandler(
    api_key ,
    api_key_secret
)
api = tweepy.API(auth)
keyword = '#HarGharTiranga'
tweets = tweepy.Cursor(api.search_tweets, q=keyword, tweet_mode='extended').items()
```

Fig. 1 Data extraction from Twitter



	UserID	Name	Tweet	Mentions	Retweet
0	5125_uttam	UTTAM SALUNKE	@VanchitRuttank: #HarGharTiranga \nहा राष्ट्री...	['वंचित वृत्तांक अधिकृत', 'Prakash Ambedkar']	1
1	Sangeeta_IND92	Sangeeta Tiwari	@AbhishektiBJP: #HarGharTiranga ....तिरंगा हमा...	['Abhishek Tiwari']	1
2	profpicturebot	Bot	@AbhishektiBJP: #NewProfilePic\n"तिरंगा हमें ज...	['Abhishek Tiwari']	1
3	Sangeeta_IND92	Sangeeta Tiwari	@AbhishektiBJP: #NewProfilePic\n"तिरंगा हमें ज...	['Abhishek Tiwari']	1
4	Sangeeta_IND92	Sangeeta Tiwari	@AbhishektiBJP: भारत के राष्ट्रीय ध्वज के रचना...	['Abhishek Tiwari']	1
...	...	...	...	...	...

Fig. 2 Sample dataset

3 Methodology

3.1 Data Extraction

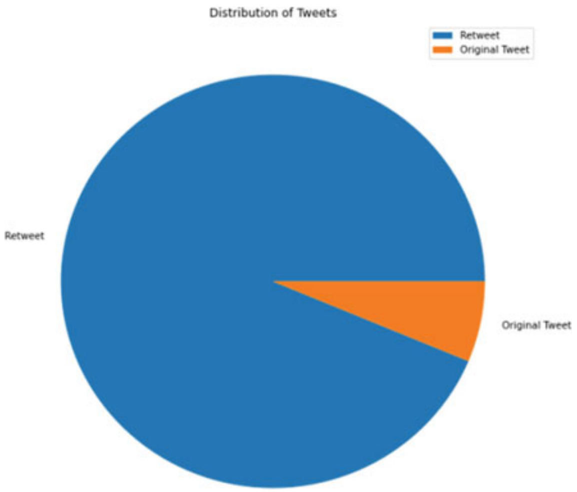
The tweets from twitter were extracted using "tweepy" library in Python. It is an easy-to-use Python library which helps in accessing the Twitter API.

The tweepy library requires Twitter API key and then fetches the raw tweet data by using a keyword. Figure 1 displays the code for extracting data from Twitter. Once the raw data is fetched, it is then processed and the final dataset is formed. A total of 35,947 tweets were extracted from the platform. The data was collected on 3rd September, 2022. Figure 2 is the dataset. The dataset has information like user ID, name of the user, tweet, users mentioned in a tweet and a Boolean indicator to show whether the tweet is an original tweet or a retweet.

3.2 Generalized Insights

After collecting the data, the next step was to pull out the basic measures from the dataset. A total of 35,847 tweets by 12,253 users were obtained. It was observed that there was a total of 33,680 retweets and 2267 direct tweets. From the dataset, the top 10 most mentioned users were tabulated and the most mentioned user was Narendra Modi, the Prime Minister of India with 2609 mentions, followed by NYKS India, a youth organization established during the Rajiv Gandhi government. The other notable mentions in the most mentioned user list included the Indian actors Pawan

Fig. 3 Distribution of tweets



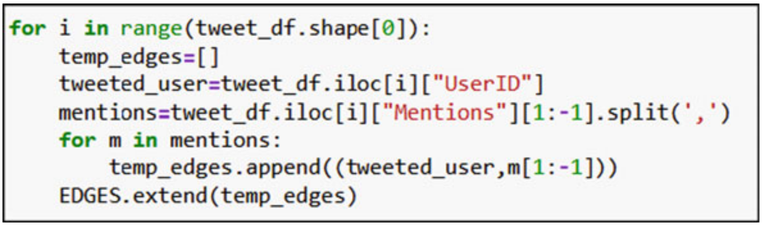
Kalyan and Mahesh Babu. Retweet in Twitter means to share a tweet from another user.

By observing Fig. 3, we can infer that retweets were more when compared with original tweets.

3.3 *Conversion of Data into a NetworkX Graph*

All the users who tweeted or retweeted were considered as nodes. When a user mentions another user, an edge is formed and these nodes, edges were used to form a network (Bastian et al. 2009).

From the collected dataset, we iterate through each tweet and an edge is formed between the users who have tweeted and the each of the users who has been mentioned in that specific tweet. Figure 4 shows the code for forming edges between nodes in the network.



```
for i in range(tweet_df.shape[0]):
    temp_edges=[]
    tweeted_user=tweet_df.iloc[i]["UserID"]
    mentions=tweet_df.iloc[i]["Mentions"][1:-1].split(',')
    for m in mentions:
        temp_edges.append((tweeted_user,m[1:-1]))
    EDGES.extend(temp_edges)
```

Fig. 4 Edges between the nodes in the network

The number of nodes increased after adding edges since the users who have not tweeted were also mentioned by the users who have tweeted. After adding edges, the number of tweets increased from 12,253 to 16,742. The network formed consisted of a total of 16,742 nodes and 25,550 edges.

3.4 Basic Social Network Measures

Statistical measures, like the density, degree, assortativity, page rank of the system are determined then from the obtained data. The density of the system given by the following formula gives use of the possible connections actually present in the system.

$$d(G) = \frac{L}{L_{\max}} \tag{1}$$

In Eq. 1, L is the actual number of connections present in the system and $L_{\max}$ is the total possible connections in the system. Usually, density lies between the 0 and 1. In this network, the density of the system is obtained as 0.0001823190.

The degree of the system gives us the number of connections a node has to other nodes. From Fig. 5, the node with the highest degree is Narendra Modi with a degree of 2362. Following Narendra Modi, the node "Amrita Mahotsav" has the second highest degree value of 693. Indian actors Pawan Kalyan and Mahesh Babu have a degree value of 353 and 315, respectively.

The degree assortativity gives us the ability of nodes of higher degree to connect to higher degree nodes with respect to, low degree nodes. In our analysis the degree assortativity for the data is obtained as $-$ 0.1211564434. The degree assortativity helps in understanding the formation of the network. A negative value of degree assortativity means that the higher degree nodes have more tendency to attach to lower degree nodes.

A node may have a high degree score which means it has numerous connections but a relatively low eigenvector centrality score, as the connections maybe with other low scored nodes. Also, a node may have a high betweenness centrality score but a low eigenvector centrality score if it is distant from the centres of importance in the network. Page Rank helps us in assigning a score to every node, indicating the level of importance a node holds in the network.

PageRank differs from eigen-centrality in terms of bringing link direction into the picture. The outcome is that nodes with many incoming connections are influential, and the nodes to which they are connected share some of that influence. PageRank is an adjustment of Katz centrality that takes into consideration this issue.

The PageRank helps in uncovering the most important and influential node in the network. From Fig. 6, it can be observed that Narendra Modi is node with the highest PageRank value in the network and we can conclude that he is most influential person in the entire network.

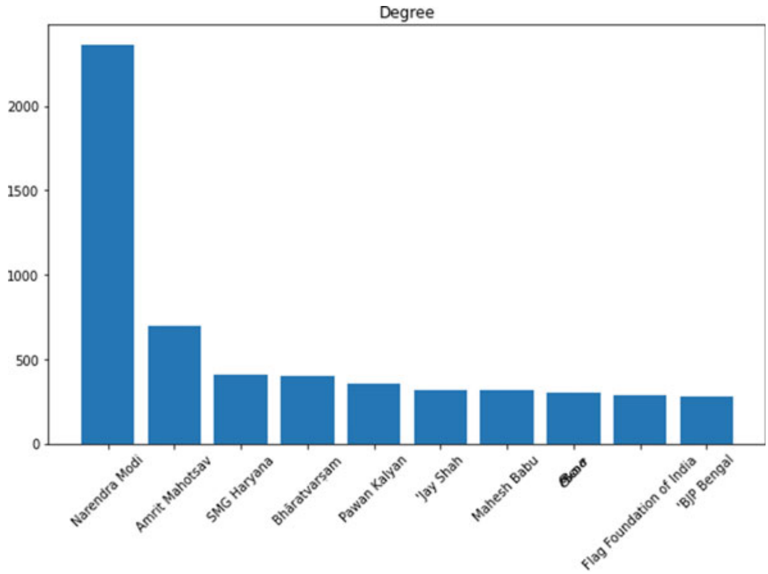


Fig. 5 Degree of the nodes

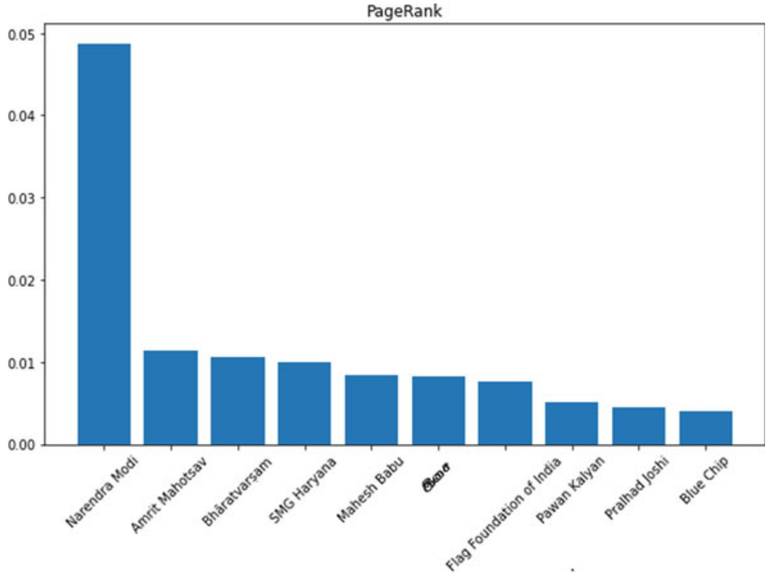


Fig. 6 PageRank centrality

3.5 Social Network Centrality Measures

In social network analysis, the centrality measures play an important role in identifying the important nodes of a network. The centrality metrics/measure gives us the information of how central a node is in the network. There are several centrality measures like betweenness centrality, closeness centrality, eigenvector centrality each of which gives us the importance of a node from different perspectives.

3.5.1 Betweenness Centrality

The betweenness centrality is metric based on the path between the nodes of a network. This measure shows us which nodes act as the connections between the nodes of a network. Nodes with high betweenness centrality score will have more influence within the network than the ones with lower betweenness centrality score.

$$g(v) = \sum_{a \neq v \neq b} \frac{\sigma_{ab}(v)}{\sigma_{ab}} \tag{2}$$

The betweenness centrality is calculated using Eq. (2), where σ_{ab} is the total number of shortest paths from node a to node b and $\sigma_{ab}(v)$ is the number of paths that pass across v.

From Fig. 7, we can clearly deduce that Narendra Modi, the Prime Minister of India acts as the bridge between nodes in most parts of the network. Other notable personalities in the list include Amit Shah, the former president of the Bharatiya Janata Party and Jay Shah, the president of BCCI.

3.5.2 Closeness Centrality

The closeness centrality is a measure which is based on the distance between two nodes. The average of the shortest distance of a node from all the other nodes in the network is calculated. The closeness centrality gives us information on how close a node to all the other nodes in the network. While betweenness centrality gives us a measure of how others are dependent on a particular node, the closeness centrality gives the access each node has to the other nodes in the network. The closeness centrality is measured using the following Eq. 3, where N is the number of nodes and $i \neq j$ and d_{ij} is the length of the shortest paths between nodes i and j.

$$\mathrm{CC}(i) = \frac{N-1}{\sum_j d(i,j)} \tag{3}$$

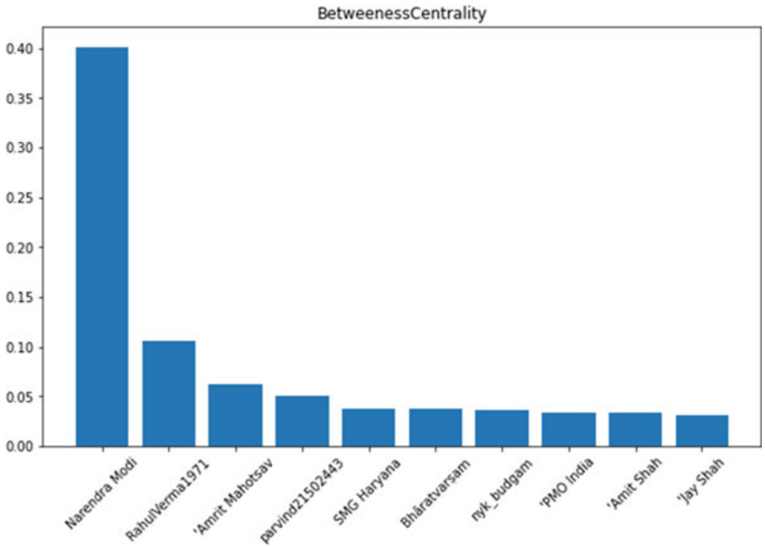


Fig. 7 Betweenness centrality

From Fig. 8, we can say that the spread of information between the nodes of the network has been faster and more efficient through the node Narendra Modi. We can conclude that Narendra Modi is the most important/central person in the network.

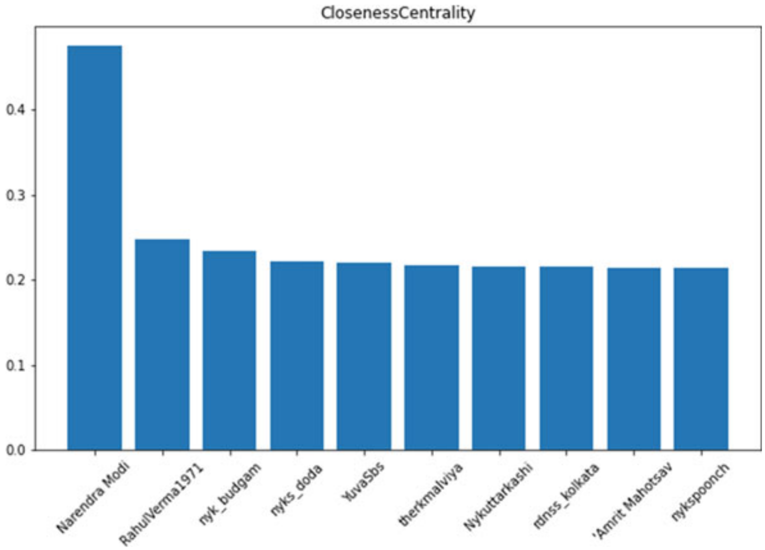


Fig. 8 Closeness centrality measure

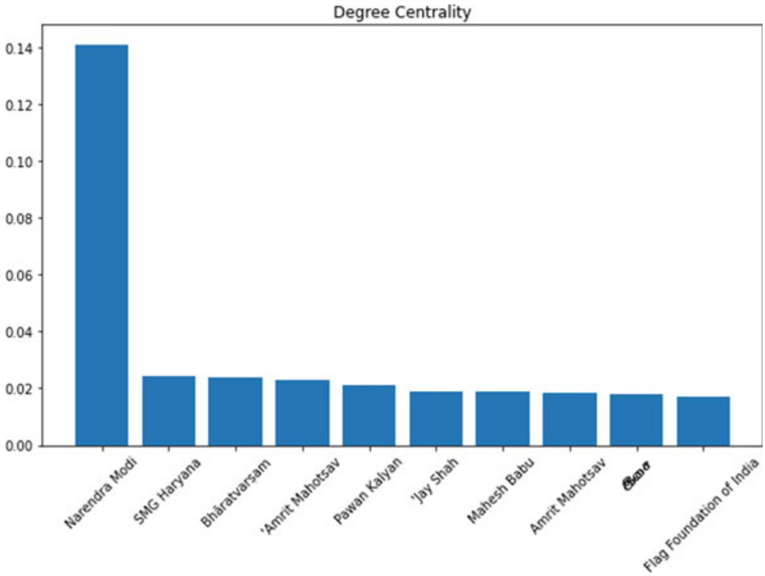


Fig. 9 Degree centrality

3.5.3 Degree Centrality

Degree centrality is one of the measures to analyze the influential node in a network. It is simply the count of total number of connections to a node. But a disadvantage with this measure is it cannot differentiate between quality and quantity. Degree centrality cannot differentiate between the prime minister and a normal student. The degree centrality is measured using the following Eq. 4.

$$C_d(j) = \sum_{j=1}^{n} A_{ij} \tag{4}$$

In our network, we got Narendra Modi as the node with highest degree centrality score, which is clearly visible in the graphical representation of degree centrality in Fig. 9. We also got Pawan Kalyan and Mahesh Babu, two famous actors in the top 10 list. The top 10 had few normal two users whose followers were very less when compared with the others in list.

3.5.4 Eigenvector Centrality

The eigenvector centrality helps in measuring the influence of a node in the network. Relative scores are given to all the nodes in the network by finding the largest eigenvalue in the network and its associated eigenvector of the adjacency matrix. It is a

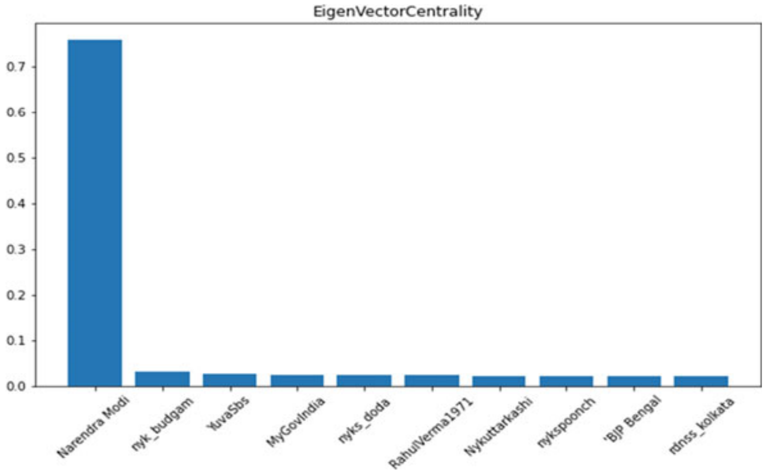


Fig. 10 Eigenvector centrality

generalized version of the degree centrality. The eigenvector centrality is measured using the following Eq. 5.

$$x_v = \frac{1}{\lambda} \sum_{t \epsilon M(v)} x_t = \frac{1}{\lambda} \sum_{t \epsilon V} a_{V,t} x_t \tag{5}$$

The above formula for eigenvector can be rewritten as, $Ax = \lambda x$ where λ is a constant.

From Fig. 10, it is clearly evident that “Narendra Modi” is the most influential in the entire network as observed through eigenvector centrality.

From all the centrality measures calculated, we can deduce that Narendra Modi is the most important person/node in the network and also the most influential in the network.

3.6 Word Cloud

A word cloud is a cluster of words in different sizes. The size of the word in the word cloud is directly proportional to the importance of the word and the number of times the word has been used. There were a total of 2088 unique words used in the tweets. From the above word cloud, we can see that the word “HarGharTiranga” is the most used word in the observed tweets. Other words making a mark in the word cloud include “Azadi” and also “Indian Army”, which were mentioned in 305 tweets. Figure 11 represents the word cloud formed from the tweets in our network.

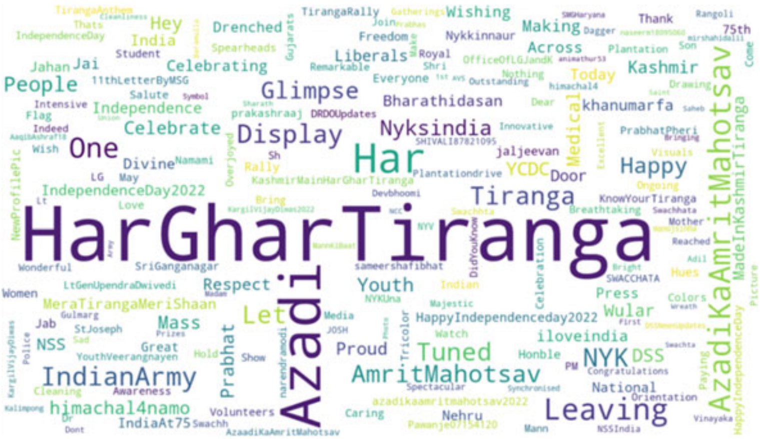


Fig. 11 Word cloud for tweets

3.7 Sentiment Analysis

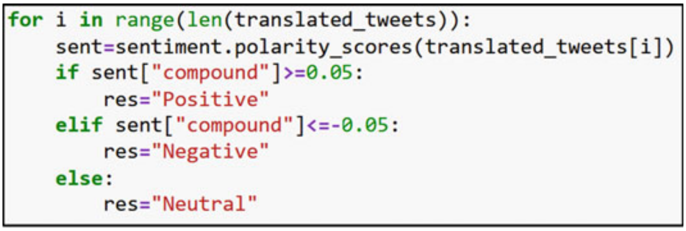
Sentiment analysis, also known as opinion mining, is process which involves computationally determining whether an opinion is positive, negative or neutral (Nooy et al. 2018). In politics, sentiment analysis is used to keep track of the view about oneself among the people. It helps in finding the consistency and inconsistency between the statements and actions made. It also serves as an important tool in the prediction of election results. It is used in several other fields like business, where companies use it to develop new strategies based on the feelings of their customers. It acts like a feedback mechanism in business. It is also used to analyze social phenomena and understand it for predicting potentially dangerous situations (Landherr et al. 2010; Yadlapalli et al. 2020).

Valence Aware Dictionary and Sentiment Reasoner (VADER) sentiment analysis is a rule-based sentiment analysis tool and a lexicon (Hutto and Gilbert 2014). It gives the opinion a score which tells us about the positivity and negativity of the sentiment. It assesses the sentiment of a given data without any previous training unlike machine learning, where we have to train models before deploying them.

A composite score is calculated by adding the valence scores of each word in the lexicon, adjusting them in consonance with the rules, and then normalizing it to a value between − 1 (very negative) and + 1 (very positive) increase. This is a most useful metric when you want a one-dimensional measure of sentiment for a given sentence (Hutto and Gilbert 2014).

Normalization is usually calculated by using the below Eq. 6,

$$x = \frac{x}{\sqrt{\alpha + x^2}} \tag{6}$$



```
for i in range(len(translated_tweets)):
    sent=sentiment.polarity_scores(translated_tweets[i])
    if sent["compound"]>=0.05:
        res="Positive"
    elif sent["compound"]<=-0.05:
        res="Negative"
    else:
        res="Neutral"
```

Fig. 12 Code for sentiment analysis

where x = sum of valence scores of constituent words and α = normalization constant.

The compound score is a measure that calculates the sum of all ratings in the lexicon, normalized from − 1 (very negative) to 1 (very positive) (Hutto and Gilbert 2014).

$$\text{Positive sentiment} \rightarrow \text{Compound score} \geq 0.05$$

$$\text{Neutral sentiment} \rightarrow \text{Compound score} > -0.05 \text{ and Compound score} < 0.05$$

$$\text{Negative sentiment} \rightarrow \text{Compound score} \leq -0.05$$

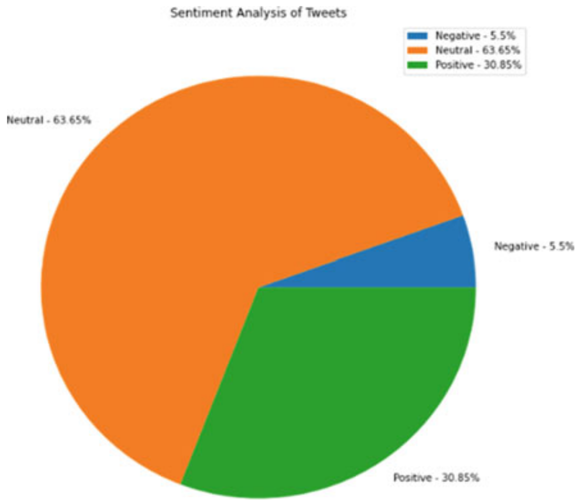
Figure 12 shows the code for implementing the logic of threshold for each sentiment. Based on the above-mentioned threshold values for each sentiment by analyzing the 35,847 tweets, around 63.65% of tweets were neutral and 30.85% were positive. Only 5.5% of tweets were negative. The sentiments in the tweets reflect the emotions or mindset of the tweeters. Thus, we can understand that majority of the tweeters were having a positive or neutral impression on the HarGharTiranga campaign. Figure 13 visualizes the distribution of positive, neutral and negative tweets from the dataset.

3.8 Visualizing the Network Using Gephi

Gephi is open-source visualization software for graphs and network. It helps in displaying large networks in real time using a 3D rendering engine for faster exploration. A flexible multitasking architecture provides new ways to process complex datasets and deliver valuable visual results. By using this software, our network has been visualized.

The ForceAtlas2 layout has been used, which is the standard choice for this type of analysis. It places the nodes simulating the physical forces of attraction and repulsion between them. When two knots are tied, they are closer together; if they are not connected, they are further apart. This arrangement creates an informative graph of communities, because users belonging to the same community are grouped together, while users from different communities are located in different regions.

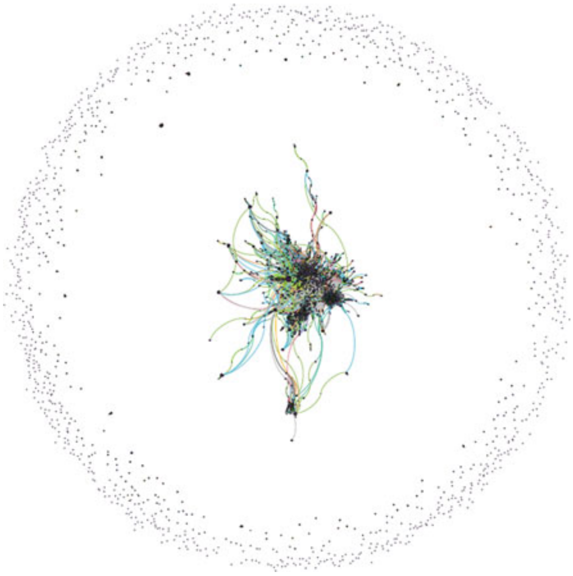
Fig. 13 Sentiment analysis of tweets



From the above network graph Fig. 14, we can visually see that most of the nodes are left unconnected and a part of the network is tightly connected in the centre. The poor density observed earlier can be now visually seen through this visualization of the network.

The yellow-coloured node in Fig. 15 is "Narendra Modi". It is clearly visible that his node is the most connected in the network. Figure 16 shows the labelled version of Fig. 16, where the central yellow-coloured node is labelled as "Narendra Modi".

Fig. 14 Visualization of the network



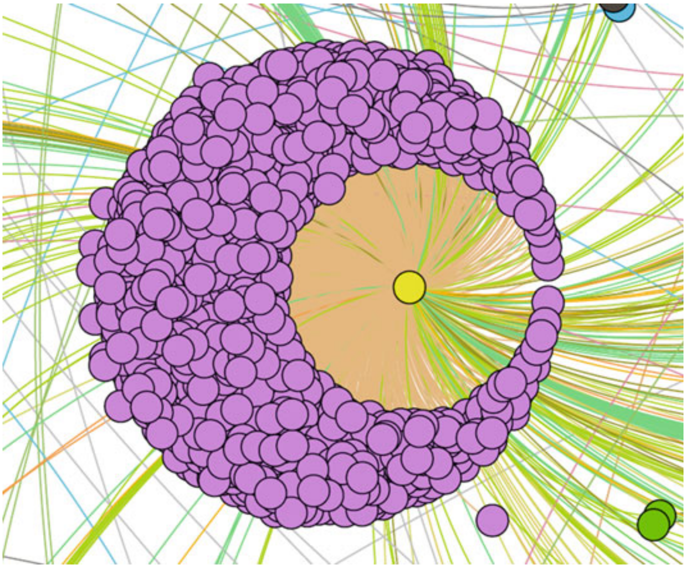


Fig. 15 Visualization for the node "Narendra Modi" (yellow colour)

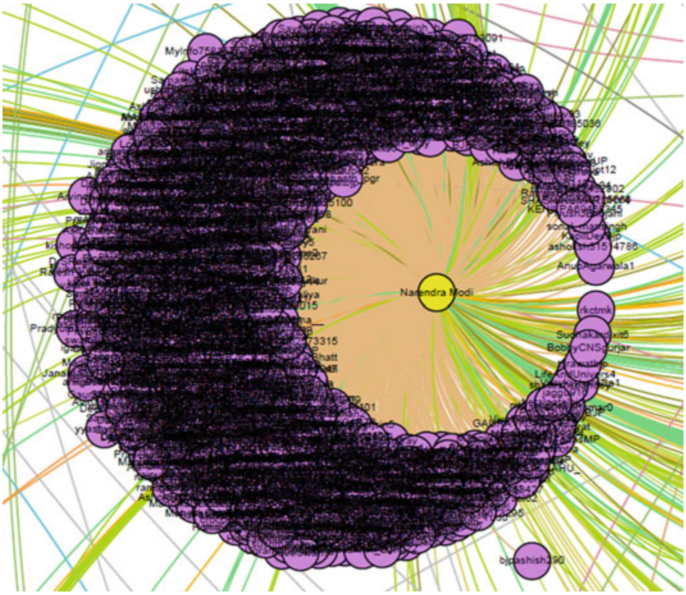


Fig. 16 Labelled version for Fig. 15

4 Conclusion

This study explores the influential nodes for HarGharTiranga hashtag in twitter by employing social network analysis. This study suggests few important findings, first most of the tweets in the dataset was retweets. This shows that majority of the users tend to retweet tweets of influencers rather than tweeting their own tweet. Second, almost in all the centrality measures, Narendra Modi—the prime minister of India emerged as the top scorer. This reflects the influence of Narendra Modi in Twitter. Third, while considering the top 10 scorers in all the measures few famous personalities, like Amit Shah, Jay Shah, Pawan Kalyan and Mahesh Babu have been found to be influential too. But Narendra Modi had a huge difference in scores compared with the others. This shows how Narendra Modi has emerged as a monotonous leader in India.

Fourth, HarGharTiranga, Azadi Ka Amrit Mahotsav and Indian Army were the most frequent occurring words in the tweets. The occurrence of "Indian Army" in the tweets reflects the gratitude of people towards the army by mentioning them in their tweets related to Independence Day. Fifth, by analyzing the sentiments of the tweets, majority of the tweets were either positive or neutral. So, this shows the positive impact and peoples' acceptance towards the HarGharTiranga campaign.

References

Aisha TS, Wok S, Manaf AMA, Ismail R (2015) Exploring the use of social media during the 2014 flood in Malaysia. Procedia Soc Behav Sci 211:931–937. ISSN 1877-0428

Bashir S, Bano S, Shueb S, Gul S, Mir AA, Ashraf R, Shakeela, Noor N (2021) Twitter chirps for Syrian people: sentiment analysis of tweets related to Syria chemical attack. Int J Disaster Risk Reduction 62:102397. ISSN 2212-4209

Bastian M, Heymann S, Jacomy M (2009) Gephi: an open source software for exploring and manipulating networks. In: International AAAI conference on weblogs and social media

Broniatowski DA, Paul MJ, Dredze M (2014) Twitter: big data opportunities. Science 345(6193):148

De Nooy W, Mrvar A, Batagelji V (2018) Exploratory social network analysis with Pajek. University Press, Cambridge

Hrishiah M, Safar M, Mahdi K (2016) Modeling Twitter as weighted complex networks using retweets. In: 2016 IEEE/ACM International conference on advances in social networks analysis and mining (ASONAM)

Hutto C, Gilbert E (2014) VADER: a parsimonious rule-based model for sentiment analysis of social media text. In: Proceedings of the international AAAI conference on web and social media, vol 8(1)

Iglesias JA, García-Cuerva A, Ledezma A, Sanchis A (2016) Social network analysis: evolving Twitter mining. In: 2016 IEEE International conference on systems, man, and cybernetics (SMC), pp 001809–001814

Landherr A, Friedl B, Heidemann J (2010) A critical review of centrality measures in social networks. Bus Inf Syst Eng 2:371–385

Medhat W, Hassan A, Korashy H (2014) Sentiment analysis algorithms and applications: a survey

Mishra P, Rajnish R, Kumar P (2016) Sentiment analysis of Twitter data: case study on digital India. In: 2016 International conference on information technology (InCITe)—the next generation IT summit on the theme—internet of things: connect your worlds, pp 148–153. https://doi.org/10.1109/INCITE.2016.7857607

Mukherjee S, Malu A (2012) TwiSent: A multistage system for analyzing sentiment in Twitter. In: Proceedings of the 21st ACM international conference on Information and knowledge management

Nair MR, Ramya GR, Bagavathi Sivakumar P (2017) Usage and analysis of Twitter during 2015 Chennai flood towards disaster management. Procedia Comput Sci 115:350–358. ISSN 1877-0509

Nair AJ, Veena G, Vinayak A (2021) Comparative study of Twitter sentiment on COVID—19 Tweets. In: 2021 5th International conference on computing methodologies and communication (ICCMC), pp 1773–1778

Naveenkumar KS, Vinayakumar R, Soman KP (2019) Twitter dataset for sentimental analysis and application of classical machine learning and deep learning

Otte E, Rousseau R (2002) Social network analysis: a powerful strategy, also for the information sciences. J Inf Sci 28(6):441–453

Ramya GR, Bagavathi Sivakumar P (2021) An incremental learning temporal influence model for identifying topical influencers on Twitter dataset. Soc Netw Anal Min 11:27. https://doi.org/10.1007/s13278-021-00732-4

Sachin Kumar S, Anand Kumar M, Kp S (2017) Sentiment analysis of tweets in Malayalam using long short-term memory units and convolutional neural nets. https://doi.org/10.1007/978-3-319-71928-3_31

Tanuja U, Gururaj HL, Janhavi V (2019) An exploratory analysis on data features and analysis techniques in social networks. In: 2019 11th International conference on communication systems & networks (COMSNETS)

Yadlapalli SS, Rakesh Reddy R, Sasikala T (2020) Advanced Twitter sentiment analysis using supervised techniques and minimalistic features. In: Hu YC, Tiwari S, Trivedi M, Mishra K (eds) Ambient communications and computer systems. Advances in intelligent systems and computing, vol 1097. Springer, Singapore

Yum S (2020) Social network analysis for coronavirus (COVID-19) in the United States. Soc Sci Q 101(4):1642–1647. https://doi.org/10.1111/ssqu.12808. Epub 28 May 2020. PMID: 32836475; PMCID: PMC7283848

A Survey Analysis on Dental Caries Detection from RVG Images Using Deep Learning

P. Nageswari, Piyush Kumar Pareek, A. Suresh Kumar, Pai H. Aditya, M. S. Guru Prasad, and Manivel Kandasamy

1 Introduction

In the medical field, computer-supported evaluation software has been utilized to obtain second opinions. Deep learning techniques are difficult to use, but they have been incorporated with encouraging results for a number of health applications. With teeth, mouth is vital component of body, and each person's smile is unique. Poor oral health care leads to gum disease, dental caries, tooth loss, bone loss, and other dental issues. Oral illnesses impact ~4 billion people around the world (Singh et al. 2022). A developing area of study in the healthcare industry is medical imaging. Medical imaging is crucial for the early identification, diagnosis, and treatment of disorders. Dental caries is one of the most common conditions affecting people of all ages globally. Using radiovisiography (RVG) or dental X-ray images (Tomer et al. 2020), it could be challenging to spot dental caries in its early stages. Tooth decay,

P. Nageswari
Department of Mathematics, Mahendra Institute of Technology, Namakkal, Tamil Nadu 637503, India
e-mail: nageswarip@mahendratech.org

P. K. Pareek
Computer Science and Engineering, NITTE Meenakshi Institute of Technology, Bangalore 560064, India

A. Suresh Kumar (✉) · P. H. Aditya · M. S. Guru Prasad
School of Computer Science and Engineering, JAIN (Deemed-to-be University), Bengaluru, Karnataka 562112, India
e-mail: falsesuresh79.slm@gmail.com

M. S. Guru Prasad
e-mail: guruprasad.cse@geu.ac.in

M. Kandasamy
Unitedworld School of Computational Intelligence, Karnavati University, Gandhinagar, Gujarat 382422, India

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_12

often known as dental caries, is a widespread chronic ailment that affects people all over the world and is caused by the erosion of dental enamel. The bacteria that cause dental decay produce acid that damages tooth enamel, eventually causing a small hole in the tooth. Formation of dental caries is though multifarious contact among acid producing bacteria and fermentable carbohydrate, teeth, saliva (Arora et al. 2020). Digital radiography has been used in dentistry for the past 20 years, and dental professionals are using it more frequently every day. Dental radiography displays tooth decay and bone loss. The presence of a hidden dental structure is not visible during a physical examination (Singh et al 2021).

Deep learning has had a substantial impact on the medical industry for a number of years, yielding results that are well acknowledged (Chandrappa et al. 2021). Despite the database's small size, dental surveys show that several studies are carried out to detect and cure dental caries. There is no publicly accessible or industry-standard database that contains dental X-ray pictures (Naveen Kumar et al. 2019). A database of 1336 dental X-ray images were produced by the authors (Mohd et al. 2021; Rubidha et al. 2022; Gupta et al. 2022). Since dental caries can range in size from tiny to large, it can be difficult to characterize patterns using dental X-ray images for caries identification (Nisha Chandran and Gangodkar 2021). In order to accurately diagnose dental cavities from 2D dental radiographs, CNN, a deep learning architecture, must be used. The difficult task of image segmentation and feature extraction from dental X-ray pictures is characterizing the area of a tooth with caries (Haseena 2022; Rajesh et al. 2020).

In light of the importance of dental caries crowd detection and analysis, numerous reports are recommended (Jawahar et al. 2022). These articles frequently discuss computer vision-based CNN models to detect dental caries. In this study, the three methodologies are density-based, regression-based, and detection-based will be contrasted (Kirubakaran et al. 2021).

2 Methodology

Convolution neural network is designed to identify between dental caries photographs with a 2D shape and performs better in this area than other deep learning models (Balamurugan et al. 2022). The ability of CNN to extract features is quite good. CNN maintains the spatial interaction of pixels, in contrast to vector image processing techniques. We have gathered a collection of roughly 1336 radiovisiography (RVG) X-ray pictures from dental clinic in India, since the typical dental X-ray database is not online (Balamurugan et al. 2022). X-rays of the patient's mouth are taken using dental imaging sensors, and RVG images are stored using Kodak's Care Stream 6100 RVG software. The dimensions of each RVG image are 748 × 512. To convert .rvg files into the .jpg format, which is supported by practically all dental clinics, one needs a license copy of the RVG software (Parasuraman et al. 2023). From the X-ray image, each tooth was clipped independently. These teeth sample images have been cropped and shrunk to 224 × 224. Every image is reduced in size and saved in a folder labeled "caries" or "no caries."

Database size is increased by applying image augmentation methods like rotate image positive by 15°, rotate image negative by 15°, rotate image positive by 60°, rotate image negative by 60°, rotate image positive by 90°, rotate image negative by 90°, vertical flip, horizontal flip, and mixture of horizontal, vertical, and rotating images by angle 15. A sample dental database with teeth with and without caries is shown in Fig. 1a, b.

A. *VGG-16 Architecture*

VGG-16 would be a convolutional neural network plan that won the ILSVR challenge in 2014. This is now recognized as among the top optical prototype plans on the market. The much more notable feature of VGG-16 would be that, rather than expanding the number of hyper-limits, researchers focused upon creating 3 × 3 channel convolution layers with stage 1 using similar padding and max pool layer as stage 2's 2 × 2 channel. Throughout the architecture, it continually follows the convolutional and maximum pool layers' game plan (Fig. 2).

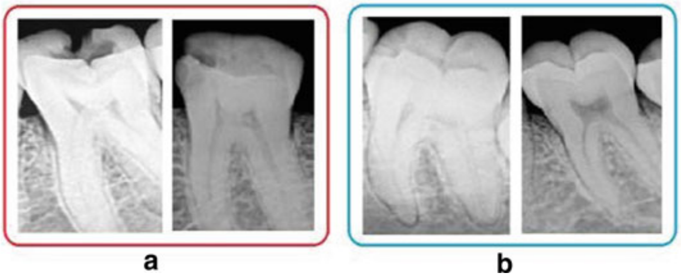


Fig. 1 Sample RVG images **a** caries, **b** no caries

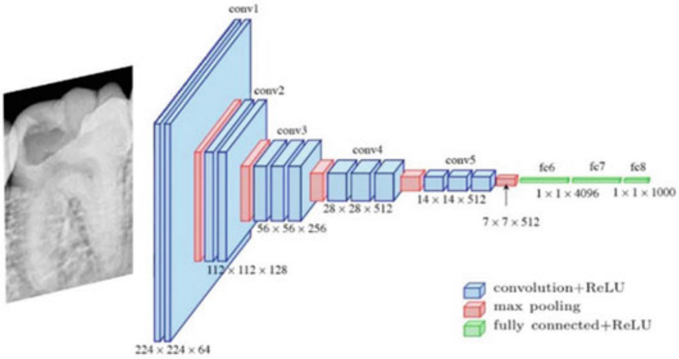


Fig. 2 VGG-16 architecture

Image Data Generator from the Keras. Preprocessing library to do VGG-16 model on dental X-ray images. The goal of Image Data Generator is to input information with names into the model as accurately as possible. It is a crucial class since it contains multiple capabilities such as rescaling, turning, zooming, flipping, and so on. The best part about this class is that it has no bearing on the information on the plate. This class hustles information while passing it on to the show. As a result, Image Data Generator will call all of the information inside the caries coordinator caries and the no-caries envelope no-caries. As a result, data can be successfully transmitted to a neural network.

The VGG-16 was designed using 1104 dental X-ray pictures of size 224 × 224 and tested on 121 images for 37 emphases, achieving a 75.3% testing accuracy. After the 38th emphases, the model exhibition begins to disintegrate.

B. *Inception V3*

A starting network is a deep neural network with a construction plan that includes repeated components known as beginning modules. It used a unique set of techniques to push execution, both in terms of speed and precision. Its continuous progress has resulted in the creation of a network with few variants.

(i) Inception v1

The size of striking parts in an image can range from exceedingly small to incredibly large. Because there can be such a wide range of information, choosing the proper piece size for convolution movement can be difficult. The more obtrusive piece is chosen for knowledge that is passed on all more locally, and the bigger portion is liked for information that is distributed all more throughout the world. Overfitting is a risk with really deep networks. It is also difficult to depart because the trend has resurfaced across the entire network. Stacking massive convolution exercises in a haphazard manner is computationally expensive. If channels of varying sizes had the choice to chip away at a comparative level, the network would essentially get nothing "more widespread" rather than "deeper." What do you think? demonstrates a "naive" conception module. It conducts convolution on commitment with three different channel sizes (1 × 1, 3 × 3, 5 × 5). Furthermore, maximum pooling has been completed. The results are connected and forwarded to the next module of the start-up process. Figure 3 shows the Naïve version of inception module.

To make it less expensive, deep neural networks are computationally expensive. Before the 3 × 3 and 5 × 5 convolutions, a 1 × 1 convolution layer is added. Although it may appear counterintuitive to add more activity, 1 × 1 convolutions are undeniably less expensive than 5 × 5 convolutions, and a lower number of information channels also help. Instead, then being shown before the max pooling layer, 1 × 1 convolution is shown after it.

(ii) Inception v2

Reduce the size of the illustrative bottleneck. When convolutions do not change components of information significantly, neural networks operate well, according to

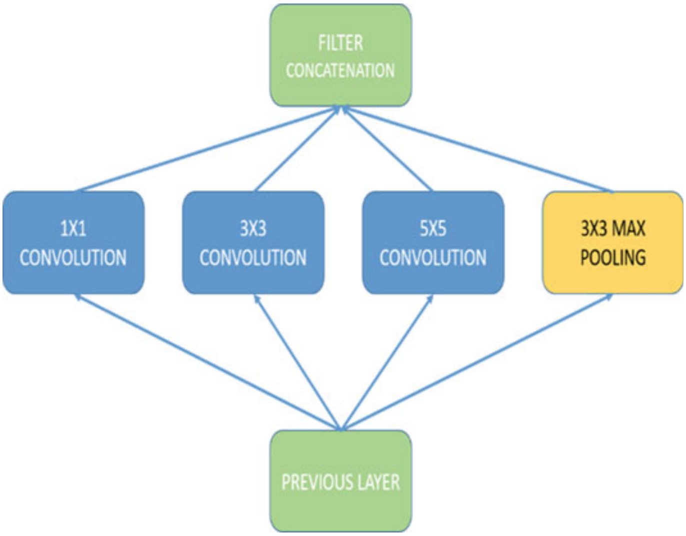
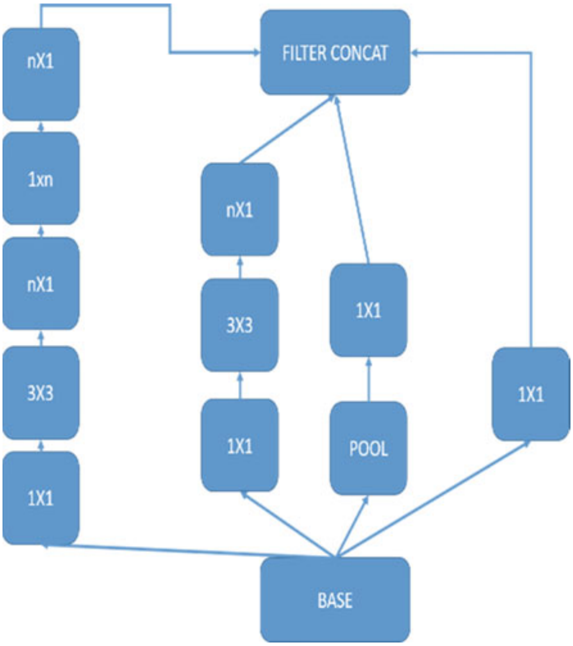


Fig. 3 Inception module Naïve version

Fig. 4 Stacking method causes lift in execution



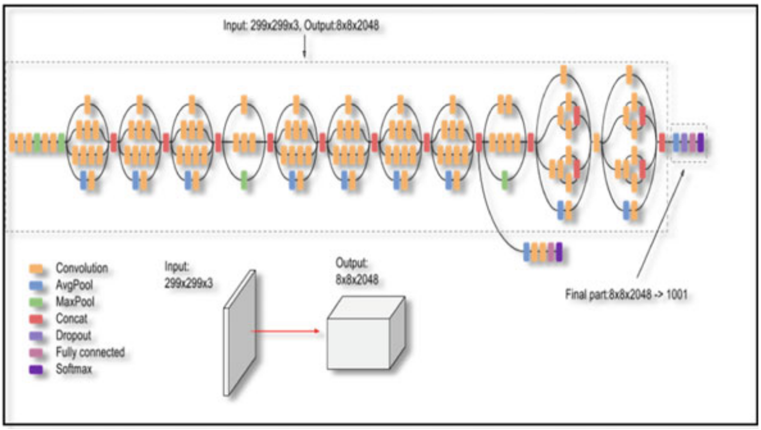


Fig. 5 Stacking method causes lift in execution

instinct. Excessive usage of lessening characteristics can result in data loss, which is regarded as “an example of a bottleneck” (Fig. 4).

(iii) *Inception v3*

Until near the end of the training procedure, when correctness was approaching immersion, auxiliary designations did not make much of a difference. It was decided to look into possible results to improve on Beginning v2 without necessarily modifying modules. As a result, version 3 was released, which included the RMS Prop Streamlining agent, factorized 7 × 7 convolutions, batch norm in auxiliary distributions, and name smoothing overfitting is avoided (Fig. 5).

C. *AlexNet*

AlexNet engineering consists of five convolutional layers and three completely associated layers, as seen in Fig. 6. A number of Convolutional Portions (channels) extract fascinating visual highlights. There are usually several parts of identical size in a single convolutional layer. AlexNet’s primary Conv Layer, for example, has 96 bits of 11 × 11 × 3 size. A section’s width and height are usually equivalent, and its depth is the same for the range of channels. Following the initial two convolutional layers are Covering Max Pooling layers, which are shown next. The third, fourth, and fifth convolutional layers are directly linked. The Covering Max Pooling layer follows the fifth convolutional layer, and the consequence is the advancement of two entirely associated layers. Softmax categorization is handled by a second entirely related layer with 1000 class marks. After all convolution and fully related layers, ReLU nonlinearity is applied. Prior to doing pooling, ReLU nonlinearity of the first and second convolution layers is followed by a neighborhood standardization venture.

AlexNet’s presentation will be ruined if any of the convolutional layers are removed. Engineering was done on dental X-ray pictures for 67 different ages, with a precision of 53.98%.

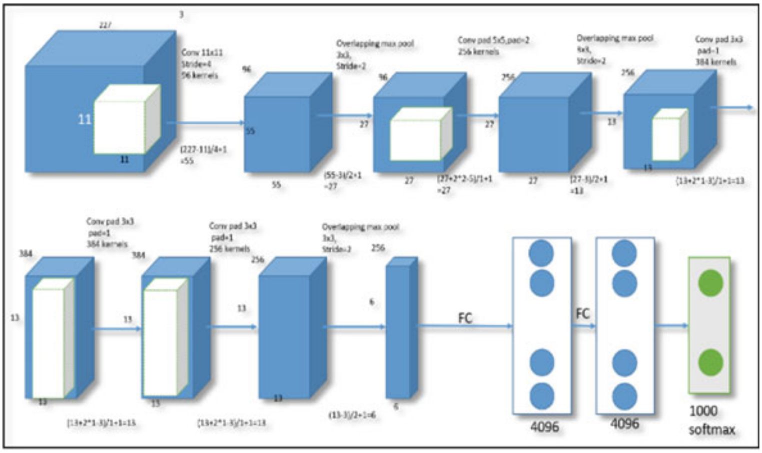


Fig. 6 Stacking method causes lift in execution

3 Proposed Architecture

Proposed architecture of CNN for dental caries detection (CCNN DCD) is given in Fig. 7. Convolutional neural networks, often known as CNNs or ConvNets, are a type of neural network that spends a lot of time processing data with lattice-like geography, such as images. CNN automatically distinguishes relevant elements with little to no human intervention. CNNs, like traditional neural networks, were energized by neurons in human and animal brains. Three important benefits of CNN were found by Singh et al (2021): similar representations, insufficient connections, and parameter allocation. In contrast to conventional fully associated (FC) networks, CNN uses shared loads and neighborhood linkages to use 2D information structures such as picture signals. This activity uses a small number of borders, which simplifies both preparation interaction and network velocities. A typical type of CNN, known as a multilayer perceptron (MLP), consists of several convolution layers preceding sub-testing (pooling) levels, with FC layers serving as completion layers. The advantages of employing CNNs over other classic neural networks in computer vision are almost impossible to ignore. A multilayer perceptron is a form of CNN that has numerous convolution layers before the sub-testing (pooling) levels, with FC layers functioning as completion layers.

4 Results and Discussion

In order to test and trained 1300 images out of 6000 RVG scanned images received from medical center in India. Out of those, 1300 images are scrutinized for training, and 100 images for verification and validation to access the performance of a system.

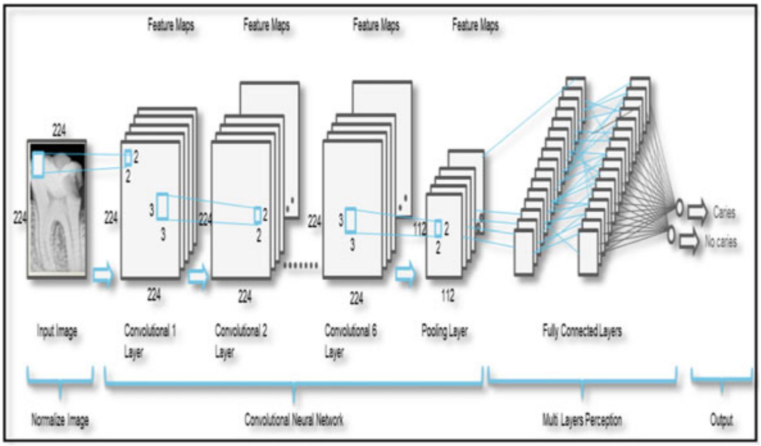


Fig. 7 CCNN architecture

This system is developed using Keras Python and TensorFlow libraries with the help of trained with Google Colab. The comparative analysis has been trained with Inception3, AlexNet, VGG-16, and proposed model.

Table 1 its showing compares the study of deep learning models with the suggested CCNN DCD model. Our model has accurately categorized 100 test photos. As a result, we were able to attain an acceptable accuracy of 93.2% achieved using given Eqs. 1, 2 and it is shown in Fig. 8.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} * 100, \tag{1}$$

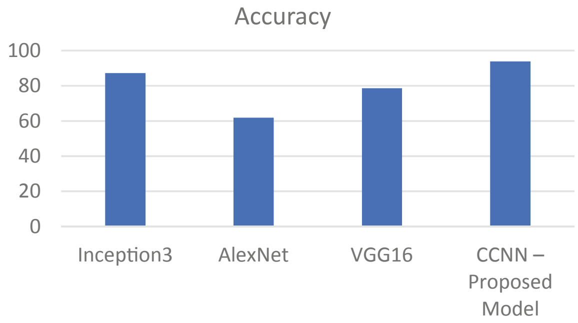
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} * 100. \tag{2}$$

Figure 9 shows the testing and trained accuracies of proposed system model with 100 epochs.

Table 1 Comparative study with CCNN

DL model	No. of epochs	Practice database size	Testing database size	Accuracy (%)
Inception3	80	1300	100	87.2
AlexNet	69	1300	100	61.89
VGG-16	37	1300	100	78.6
CCNN—proposed model	100	1300	100	93.8

Fig. 8 CCNN for accuracy comparatives



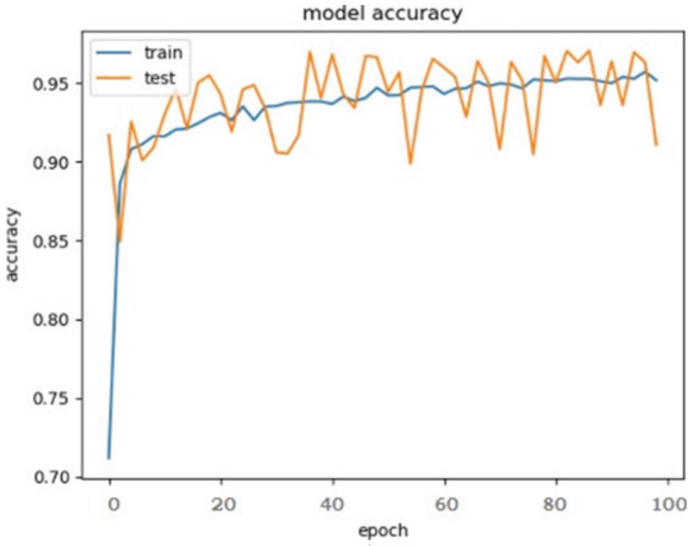


Fig. 9 CCNN for test and train accuracy models

5 Conclusion

In this paper, RVG pictures are classified using a specific, unique customized convolution network to detect for dental issues. In order to increase datasets and improve the image's visual readability, this section defines picture improvement approaches with a number of enhancement processes that highlight distinct differences between the image's attributes. The image is then double thresholder and used as a framework example to designate the focuses around the jaw. The unnecessary component was then removed after separating the upper and lower jaws in order to separate the individual teeth. A CCNN computation for dental disease distinguishing proof was employed to successfully detect impacted caries, abscesses, impacted third molars, and root fragments with the accuracy rate of 93.8% has been achieved.

References

Arora YK et al (2020) A Novel approach for detection and classification of rice leaf diseases. In: Springer—Advance in Intelligent systems and computing, vol 1155. https://doi.org/10.1007/978-981-15-4029-5_10

Balamurugan M, Kumaresan M, Haripriya V, Annamalai S, Bhuvana J (2022) Secured cloud computing for medical database monitoring using machine learning techniques. In: JeenaJacob I, Selvanayaki KS, Bestak R (eds) Notes in networks and systems, vol 444. Springer, Singapore, pp 91–109

Chandrappa S et al (2021) A novel approach to glaucoma screening using optic nerve head through image fusion and fractal geometry. J Math Comput Sci 11:4284–4300

Gupta AD et al. (2022) A framework of internet of things (IoT) for the manufacturing and image classification system. In: International conference on advance computing and innovative technologies in engineering (ICACITE), pp 293–297

Haseena S et al (2022) (2022) Prediction of the age and gen and gender based on human face images based on deep learning algorithm. Comput Math Methods Med 2022:1413597

Jawahar M, Anbarasi LJ, Ravi V, Prassanna J, Graceline Jasmine S, Manikandan R, Sekaran R, Kannan S (2022) CovMnet–deep learning model for classifying coronavirus (COVID-19). Health Technol 12:1009–1024

Kirubakaran J, Venkatesan GKDP, Sampath Kumar K, Kumaresan M, Annamalai S (2021) Echo state learned compositional pattern neural networks for the early diagnosis of cancer on the internet of medical things platform. J Ambient Intell Human Comput 12(3):3303–3316

Mohd N et al (2021) A survey: heart disease prediction using machine learning techniques. Webology 18(4):441–452

Naveen Kumar HN et al (2019) Optimization in feature extraction schemes on static images to improve the performance of automatic facial expression recognition systems. Int J Comput Sci Eng 7(6):1104–1109

Nisha Chandran S, Gangodkar D (2021) A novel image retrieval technique based on semi supervised clustering. Multimed Tools Appl 80:35741–35769

Parasuraman M, Munnangi AK, Rajeyyagari S, Sekaran R, Ramachandran M (2023) 5G wireless network-based cybersecurity analysis using software defined Phy_HetNets and Boltzmann encoder convolutional basis neural network. In: Hassanien AE, Castillo O, Anand S, Jaiswal A (eds) International Conference on Innovative Computing and Communications. ICICC 2023. Lecture Notes in Networks and Systems, vol 703. Springer, Singapore. https://doi.org/10.1007/978-981-99-3315-0_10

Rajesh E et al (2020) Analysis and Prediction of Liver Disease using Deep Learning. Int J Adv Sci Technol 29(11):2271–2278

Rubidha Devi D et al (2022) Enhancing orthopedic surgery and treatment using artificial intelligence and its application in health and dietary welfare. Hindawi J Food Qual 2022:7734650. https://doi.org/10.1155/2022/7734650

Singh AS et al (2021) Colour detection in objects using NIN implemented CNN. Special Ugdymas 1(43):3989–4001

Singh S et al (2022) Automated deep learning based disease prediction using skin health recored, issues, challenges and future directions. In: International conference on Electronics and renewable systems, pp 638–643. https://doi.org/10.1109/ICEARS53579.2022.9752422

Tomer V et al (2020) A selection method for computing the ensemble size of base classfier in multiple classifier system. In: Applied vision and Image processing. Springer—Advance in Intelligent systems and computing, vol 1155. https://doi.org/10.1007/978-981-15-4029-5_23

IoT-Based Smart Agriculture Monitoring System Using Web Application

K. S. Shashidhara, M. Aakaash, Aman Kumar, P. Girish, B. D. Parameshachari, and Veerendra Dakulagi

1 Introduction

This agriculture is an integral part of the world's economy especially in the case of developing countries. It is regarded as one of the primary source of income, employment and food, the basic needs for some. According to World Bank, agricultural development is one of the most important tools to end poverty and feed a projected 9.7 billion people by 2050 (https://www.worldbank.org/en/topic/agriculture/overview), and according to Food and Agriculture Organization, the share of agricultural population is 67% of the total population in the world (https://www.tractorjunction.com/blog/top-10-agricultural-producing-countries-in-the-world/). The top agriculture producing countries in the world are China, USA, Brazil and India (in the 4th place) (https://www.tractorjunction.com/blog/top-10-agricultural-producing-countries-in-the-world/).

Current scenario of agriculture and farming in India is going on a steady pace with lots of additional support especially the use of technologies like IoT, machine learning, image processing and so on. But still a majority of the farmers rely on traditional techniques. Some areas have lesser access to modern machinery, logistics, storage and do not get regular updates on the data of weather patterns of that area, crop health and other data that can be obtained from the fields.

K. S. Shashidhara (✉) · M. Aakaash · A. Kumar · P. Girish · B. D. Parameshachari
Department of ECE, NitteMeenakshi Institute of Technology, Bengaluru, Karnataka 56064, India
e-mail: sks.nmit@gmail.com

B. D. Parameshachari
e-mail: paramesh@nmit.ac.in

V. Dakulagi
Department of ECE, Guru Nanak Dev Engineering College, Bidar, Karnataka 585403, India
e-mail: veerendra@ieee.org

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_13

Adverse climatic changes, not monitoring crop health, inadequate or excess use of irrigation and other natural sources on the agricultural land can lead to destruction of plant life and subsequently reduce the yield.

In this project, with the use of sensors and IoT the agriculture field is monitored by looking into water content in soil, water level in the tank and smoke detection. Along with these, a crop disease detection system is used to detect the disease the plant is suffering as timely awareness of this might save lots of crops thereby helping the farmer. Apart from this, monitoring the weather is an important aspect as sudden and unpredictable climatic changes are also prevailing issues. Hence, the feature of monitoring the weather of the location of the field has also been included. A web application has been created for the user that provides the outcomes of all these features mentioned above along with the option to choose language in order to overcome language barrier.

2 Related Works

Puranik et al. (2019) have implemented automation in agriculture by using smart irrigation system and monitoring some aspects of the farm using sensors and sending the data obtained from field remotely. Varghese and Sharma (2018) have designed a system for pesticide deployment based on area density, used SVM for crop health and smart sensors for monitoring through a raspberry pi and SMS API for alerts. Deepa et al. (2020) have explained the importance of smart farming. They have proposed a model using IoT and cloud computing. Soil health monitoring system is implemented and measurement of different parameters like soil temperature, soil moisture, pH value and humidity. Abdu et al. (2020) have pointed out the importance of detection of diseases on plants. They have compared the performance of SVM and deep learning and concluded that the deep leaning model outperforms SVM by a substantial margin in terms of classification accuracy. Wiangtong and Sirisuk (2018) have implemented precision farming, control of temperature and humidity. They have installed timer, condition and hysteresis control. They have put more emphasis on optimizing plantation process. Data is shown in web application. Nagaraja et al. (2019) proposed smart agriculture management to increase crop production using machine learning and IoT and put more emphasis on reducing resource wastage. They have made a system for crop prediction using soil sample, sensory data and lab tests result. Haque et al. (2020) used two machine algorithms: support vector machines and linear regression to analyze the crop yield. It depends on parameters like water, UV rays, pesticides, fertilizer and area of land covered in that region and considered crops like wheat, corn, sugar beet and soybean. These parameters are chief key factors affecting the yield.

3 Hardware Operation

The components included in the hardware equipment of the model are—Raspberry Pi 3B+ and the microcontroller board that acts as the processing unit. All the required sensors for monitoring the agricultural field are connected to the microcontroller board. The sensors include soil moisture sensor, water level sensor, gas sensor, solenoid valves and pump. All these sensors are considered to be data acquisition units, since they are keenly responsible for the information received from the field. Relay module is as a control device that is connected to the pump and the solenoid valve. The pump and solenoid valve are activated at the required instance, with the help of the relay. The Raspberry Pi does not include an inbuilt ADC. Hence MCP3008 has been utilized which is an analog to digital converter. A battery is used as the power source for the Pi.

As shown in the block diagram (Fig. 1), all the sensors are placed on the agriculture field with the soil moisture sensor in the soil for detecting water content in the soil, the water level sensor in the tank for monitoring the irrigation and the gas sensor functioning as a smoke detector. Apart from these sensors, a motor has been placed inside the tank for pumping out water, and solenoid valve which performs the open and close functioning for providing water supply for the area of the field required, all of these controlled by the Raspberry Pi.

The functioning of the model beings as thus:

1. It starts with the soil moisture sensor. This sensor detects the water content level in the field wherever it is situated depending on a pre-set threshold which classifies as whether there is enough water content at that place. When this sensor detects that the water content level is lesser than the threshold, it sends a command or a value to the Raspberry Pi indicating that the water content is low.
2. After receiving its value, the Raspberry Pi checks the water level sensor that is placed in the tank, used for irrigating the field. If the water sensor returns a value indicating that the water tank contains sufficient water, then the Pi activates the relay module which triggers the motor in the tank and then the solenoid valve.
3. The relay module as described in the hardware description works as a switch performing open and close operations for circuits. When it receives the command from Pi, it activates the submerged motor placed in the tank to begin pumping water to the field. When the water is being pumped out, it activates the solenoid valve, which works as a mechanical tap to open and supply water to that part of the field where the water content is less.
4. The gas sensor functions as a smoke detector, monitoring the field for any accidental fires, which can cause lot of harm and damage the entire field and even neighbouring areas if not curbed properly. This gas sensor sends an output value if it detects smoke. With the value displayed on the web-app, the farmer can take appropriate measures before the smoke can spread into a bigger fire, destroying the crops and causing the farmer irreparable loss.
5. Now for sending the data from Raspberry Pi to the web application through cloud, a platform called ThingSpeak is utilized. It is an open IoT platform that is used

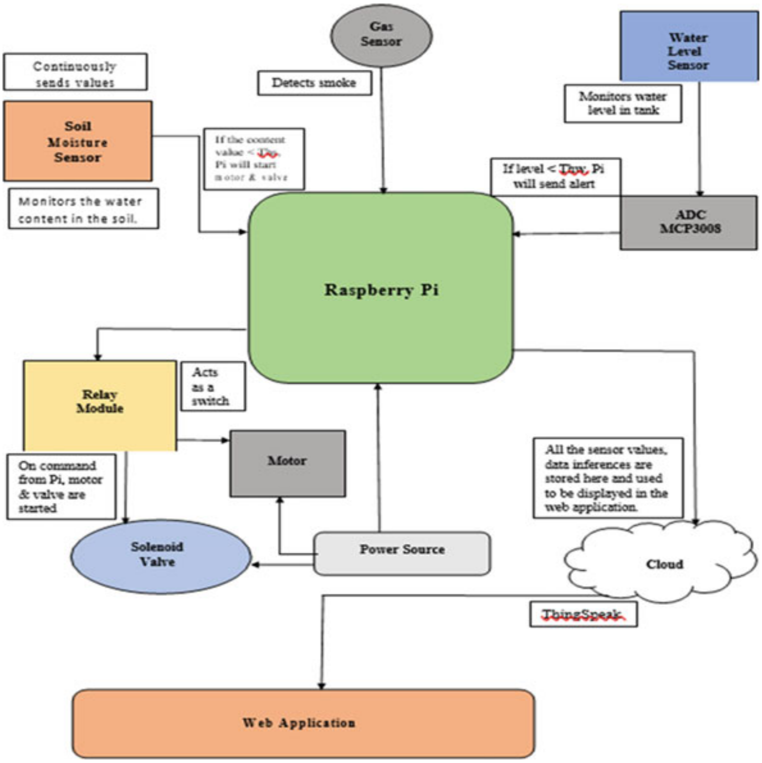

Gas Sensor
Water Level Sensor
Continuously sends values
Detects smoke
Monitors water level in tank
Soil Moisture Sensor
If the content value < Thn, Pi will start motor & valve
If level < Thw, Pi will send alert
Monitors the water content in the soil.
ADC MCP3008
Raspberry Pi
Relay Module
Acts as a switch
Motor
On command from Pi, motor & valve are started
All the sensor values, data inferences are stored here and used to be displayed in the web application.
Solenoid Valve
Power Source
Cloud
ThingSpeak
Web Application

Fig. 1 Block diagram of hardware configuration

to monitor the data online. In ThingSpeak channel (a place where date is sent and stored), the data can be categorized as public or private depending on the requirements. ThingSpeak takes around 15 s to update the readings in the page. Initially creation of a new channel in ThingSpeak is done by entering the name and description of the data to be uploaded on this channel.

6. To send data to ThingSpeak, a unique ThingSpeak API key is used, which is added in the Python master code for the on-field equipment to upload the CPU data to its website. After execution, the outputs are received accordingly. Now the values obtained in ThingSpeak are reflected on a webpage (created with the help of “Creative Tim”, an open source design site) and connected to the main web application.

4 Crop Disease Detection

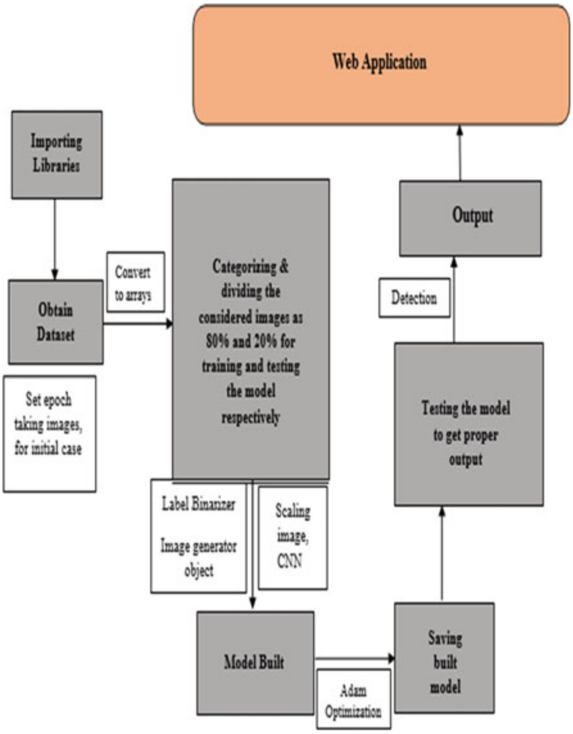
Crop disease detection is one of the most explored part right now in the use of technologies like artificial neural networks, machine learning, deep learning and CNN in agriculture. Crop loss due to diseases can be minimized. The block diagram below describes the flow of how crop disease detection procedure works.

As observed in the Fig. 2, the crop disease implementation goes through stages such as using the necessary libraries, accessing the dataset, dividing it into training and testing before building the model. Once the model is built, it is saved for future use so that it can be accessed when deploying the web application. Then it is tested to determine its accuracy and get the required output.

Here is the procedure of how the crop disease model is constructed. CNN sequential model has been used for training the network, since it involves operation using image data.

1. Importing and accessing all necessary libraries and packages for the implementation of the model. The libraries used are–NumPy, OpenCV-Python, Matplotlib, Keras and TensorFlow.

Fig. 2 Crop disease detection flow


Web Application
Importing Libraries
Obtain Dataset
Convert to arrays
Set epoch taking images, for initial case
Categorizing & dividing the considered images as 80% and 20% for training and testing the model respectively
Label Binarizer
Image generator object
Scaling image, CNN
Model Built
Adam Optimization
Saving built model
Testing the model to get proper output
Detection
Output

2. Acquiring the dataset (important step in any deep learning/machine learning project), “Plant Village”, will be accessed here. Google Colaboratory (Colab) is used to execute this model.
3. After accessing the dataset, the images are loaded by selecting at random some number of images from each of the category of the dataset (description in about the dataset), these images will be given as input later to the CNN model.
4. These loaded images are divided into training data and testing data. This is divided as such because, for working with any model to obtain output a training data must be provided that will help the model understand the features it must consider in order to give a correct result. Testing data will then be used to validate how better the model is giving the result.

5. Specifying the necessary parameters for the CNN model and building it like image parameters (height, width, depth of each image), number of epochs (number of times the model is going to be trained), then name of the CNN model—sequential model, the layers necessary like convolution layer, max pooling layer are added, batch normalization criteria, activation functions (ReLU, Softmax) are specified, dropout parameter is defined which is used to reduce overfitting in neural networks and flatten is a function used to flatten the output from the CNN layers to connect to the other layers.
6. After specifying the parameters and defining the structure of the CNN model, the loss function which is binary cross entropy and the type of optimizer—Adam optimizer to be used are specified.
7. The sequential model constructed for the function of crop disease detection as specified in the figure below consists of different layers, used to extract the features from the input image provided to the model.

- Figure 3 is the sequential model which is used for training to detect the crop diseases.
- The layer 1, layer 2 and layer 3 perform the feature extracting operation on the input image provided. Within each layer, the image undergoes convolution and returns a feature map which in turn undergoes max pooling, followed by dropout.
- After each layer (layer 1, layer 2 and layer 3), dropout of 25% takes place where 25% of neurons are dropped so that the network doesn’t depend on only one set of neurons each time it trains.
- Batch normalization stabilizes the learning process, it reduces the number of epochs for neurons depending on their training.
- At the end of the feature extracting operation of all the three layers, the classification part begins when the feature map is flattened and passed through dense neural network layers with ReLU and Softmax activation functions.
- These help in classifying the input, and Softmax returns probabilistic values for its input since it acts as a loss function.

This is the structure of the CNN sequential model used and its working when given an image as input.

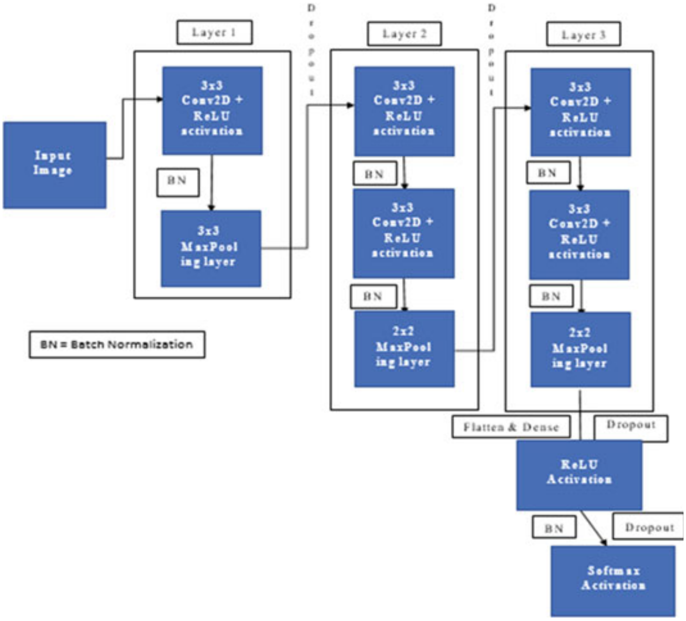


Fig. 3 CNN sequential model structure

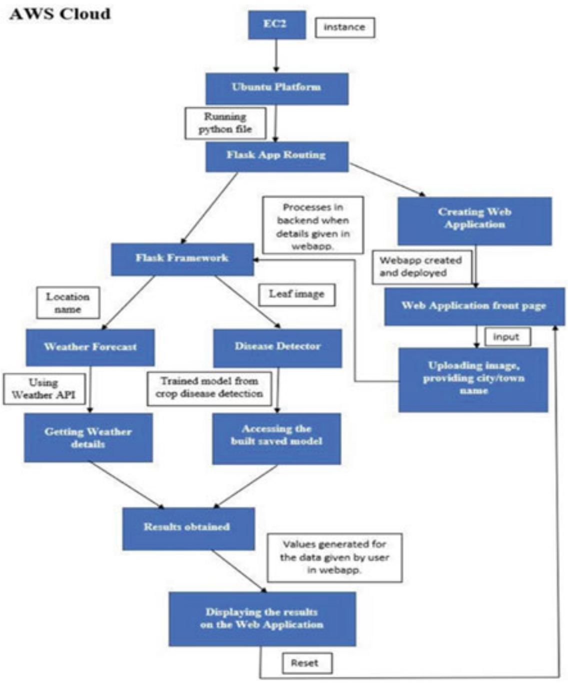
8. Then the model is trained and once it has finished training, it is saved for future use since, this saved model is accessed in the backend when operating with the web application for getting the output results.

5 Web Application

With the completion of hardware system and crop disease detection model, the chapter is the incorporation of both the modules and developing a web application. Finally, the deployment of web application in AWS Cloud services.

The web application is deployed in AWS using EC2 instance on Ubuntu server. The program starts from flask framework where app routings maps action from the web application and gets the data from the web-app, on receiving the data, city name is passed to weather forecast function where coordinates of that location is extracted and weather report is pulled from open weather portal using API, output is extracted and returned back to main function. Similarly, image is sent to disease detector function, where loaded model diagnoses the image and returns the output to main function. Finally all the outputs are rendered back to final page of web-app. The web-app again resets internally and waits for response again (Fig. 4).

Fig. 4 Web application flow


AWS Cloud
EC2
instance
Ubuntu Platform
Running python file
Flask App Routing
Creating Web Application
Processes in backend when details given in webapp.
Flask Framework
Webapp created and deployed
Web Application front page
Location name
Leaf image
Weather Forecast
Disease Detector
input
Uploading image, providing city/town name
Using Weather API
Trained model from crop disease detection
Getting Weather details
Accessing the built saved model
Results obtained
Values generated for the data given by user in webapp.
Displaying the results on the Web Application
Reset

Below are the detailed steps of the backend implementation:

1. User uploads the diseased leaf image along with location name on the landing page.
2. Flask app route is mapped to that part of web-app where input is taken and then it is stored in local drive.
3. Get weather details function is called with the parameter city name.
4. Get weather details function uses city name to get the coordinates of the place using a geopy, a Python library.
5. These coordinates are then fed into open weather API URL along with an API key, this will fetch the weather data in JSON format. Open Weather API is a weather forecast provider. It can access current weather data for any location including over 200,000 cities.
6. From the JSON file all the required data is extracted and finally returned.
7. Similarly, get prediction function is called with the image as a parameter. First the image is being loaded from the local drive then using cv2 module then we resize the image to default image size which is (256, 256) and converted to array using sklearn image to array module.
8. Labels are prepared using the function Label Binarizer.
9. Stored model is loaded into the program.
10. Result obtained from the model is returned to the main function.

Finally, all the results like prediction, prediction probability, city name, temperature, minimum and maximum temperature, pressure and visibility are rendered back to the web-app using render template.

AWS Deployment

For using cloud service AWS EC2 (Elastic Compute) instance is used. On Ec2 instance is running Ubuntu Server 20.04 on it where the web-app will be running. It has 1 GB of RAM and 15 GB of SSD storage. This a free-tier so there is a little performance limitation.

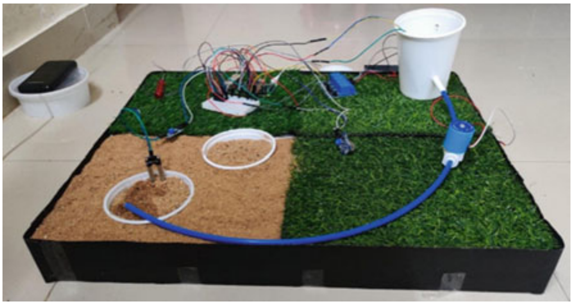
1. After creating an Ubuntu instance and providing all the credentials, Ubuntu storage is accessed using a software called WinSCP and all the web-app files are transferred into Ubuntu storage. It is a free and open source SSH file transfer protocol mainly used to transfer files from local computer to remote computer.
2. Ubuntu terminal is accessed through putty, a free and open source terminal emulator. It supports various network protocols including SCP, SSH.
3. All the libraries are installed in Ubuntu environment for execution of web-app Python file.
4. To overcome the RAM requirements for the application to run, swap memory of 3 GB is created and finally application is executed.
5. On execution, using AWS link followed by the port number landing page is loaded.

6 Results and Analysis

With the completion of the hardware, crop disease detection and web application deployment procedures, the results obtained in each of them are discussed in this section.

As observed in the Fig. 5, the solenoid valve allows flow of water from the tank to the part of the field where the soil moisture sensor detects less water content in the soil. The motor is situated inside the tank to pump out water.

Fig. 5 Prototype hardware model



The results of the sensor data obtained from here are displayed on the web application created, in a plot format, so that the user can understand by observing the varying range of the graph that the sensors are detecting and operation is going on the field.

Using the data obtained from the graphs plotted in Fig. 6, it can be observed that on training the crop disease detection model as explained in Sect. 4, the testing accuracy is 89.75% and the resulting loss is minimized. This explains that the model can detect the crop diseases efficiently.

The results obtained in the hardware and the crop disease detection operations are displayed on the web application in a user understandable efficient way. The web application consists of options to upload image of diseased leaf, enter city name and set the language as per the user requirement with the help of google translate language option. Along with these, there are two more options to view more extra detailed weather forecast results and to view the outputs of the hardware components.

As observed in Fig. 7, this is the landing page of the web application. Here the user can upload the image under the “Upload Image” option by either choosing the image from device or by taking a picture through the camera, enter area name under the “Enter City” option and on clicking the search button the user can view the results.

As observed in Fig. 8, the output for the entered data is obtained with detected crop disease as “Tomato Bacterial Spot” for the uploaded image and with entering location name as “Bangalore”, the weather forecast details are obtained in the table on the left hand side. The results are shown in English language. There is an option to change language as per the users’ requirements using the “Google Language” feature at the top left corner of the web application page as seen in Fig. 9.

From figures, it can be understood that using the Google language feature added in the web application, the user can view in the language that is suitable and easily understood by him/her. This feature is mainly implemented for avoiding the language barrier.

Further there are two more options available in the web application, “More Details” and “Hardware Portal” (Fig. 10).

On clicking the “More Details” option in the web application, the user is directed to this “AccuWeather” which is a third party website where, on entering the location

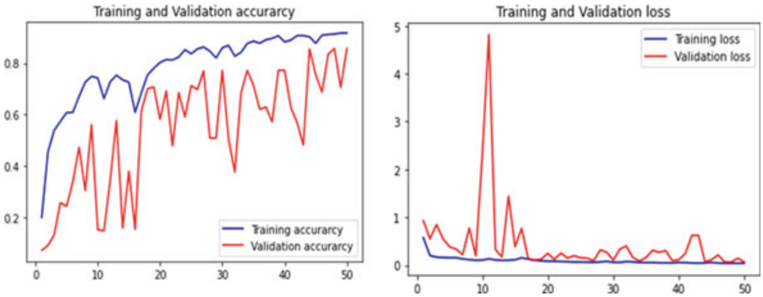


Fig. 6 Accuracy and loss of training and testing data

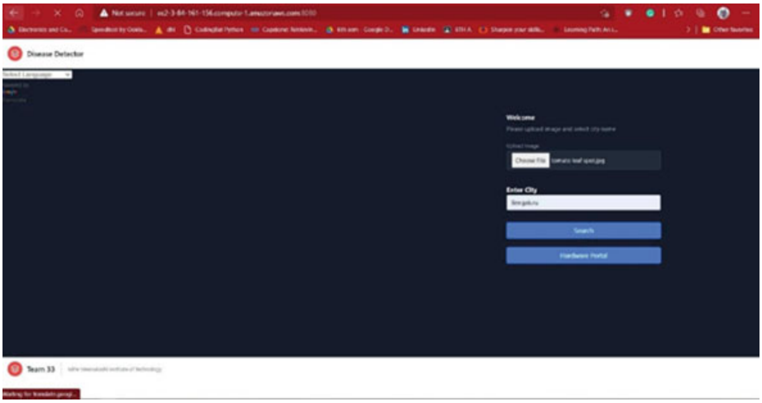


Fig. 7 Landing page of web application

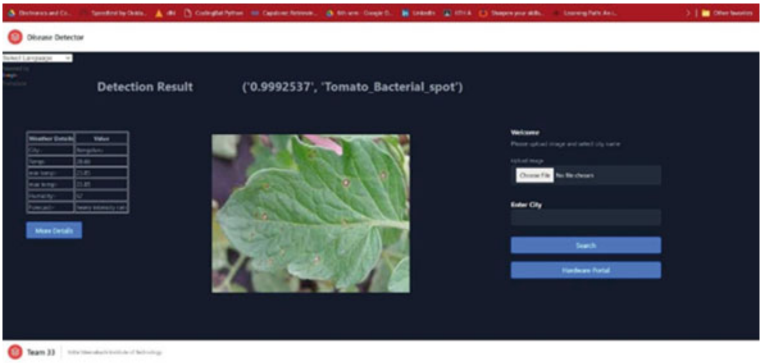

Detection Result ('0.9992537', 'Tomato_Bacterial_spot')

Fig. 8 Result view

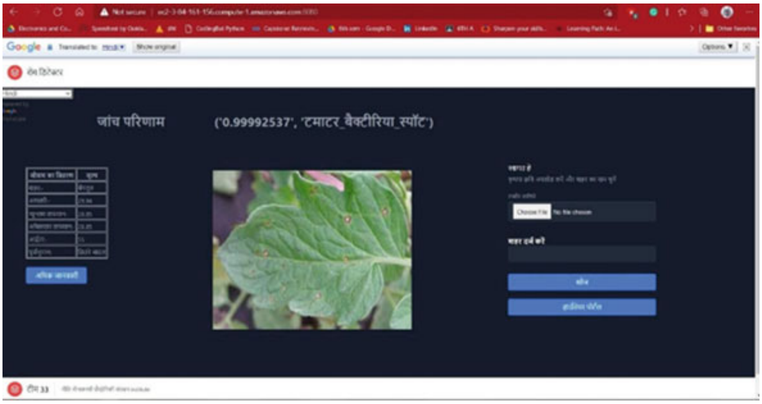

जांच परिणाम ('0.99992537', 'टमाटर_वैक्टीरिया_स्पॉट')

Fig. 9 Result view in Hindi

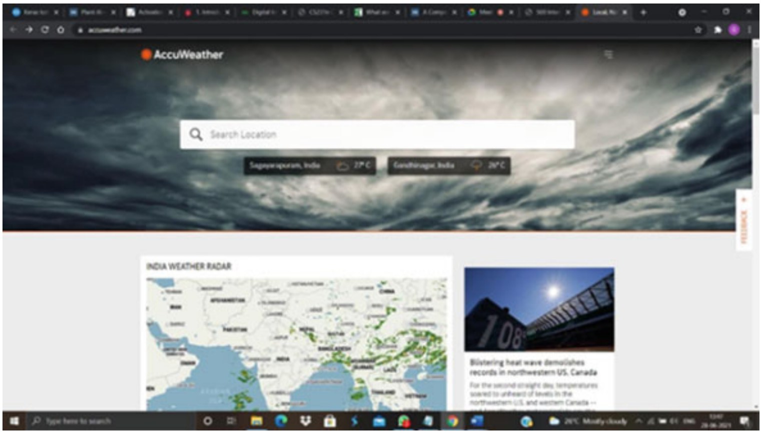

AccuWeather
Search Location
INDIA WEATHER RADAR
Blistering heat wave demolishes records in northwestern US, Canada
FEEDBACK

Fig. 10 AccuWeather page

name, further more details regarding the weather forecast can be obtained if required by the user (Fig. 11).

This “Hardware Portal”, displays the outputs obtained from the sensors placed on the field as shown in the prototype hardware model in Fig. 5, so that the user can monitor the details and decide the necessary things and do the needful that is required. The outputs are displayed here in graph format, this can help the user to understand by looking at the shape and trend in the graph regarding the water content in the soil, availability of water for irrigation when water content in soil has been discovered to be less and smoke detection action if any fire accidents.

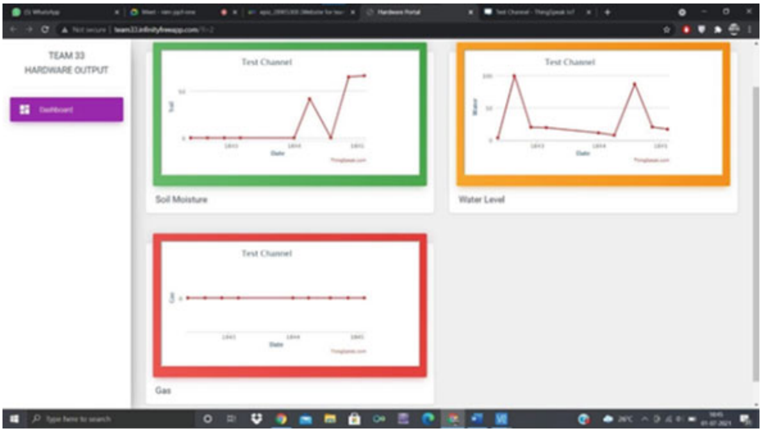


Fig. 11 Hardware portal

7 Conclusion and Future Enhancements

The project proposed here by us, depicts the usage of IoT, deep learning and other technologies in agriculture. As observed, the model consisting of the on-field equipment such as soil moisture sensor, water level sensor, gas sensor and solenoid valves helped in monitoring the agriculture field and its irrigation system. The implementation of the web application displays all these values in easily comprehensible form, together with the crop disease detection system and getting weather forecast information specific to the area of the farm. It can be noted that the model gives values of all the above said features making its functioning to be efficient and user friendly.

Apart from the current model, there are some more advancements that can be done on this in future such as adding intruder detection system for securing the crops from either humans or animals, crop detection based on the type of soil being used not only based on previous data but with the help of sensors determining soil quality and its mineral content and adding more sensors for deep understanding of field, providing farmers more exposure and information regarding market prices. Improvements such as these on the current model could result in an entirely completed product that can be used as a single module for agriculture throughout the country, thereby making an even better contribution towards digitization of agriculture and helping India also take a step towards becoming Digital India.

References

Abdu AM, Mokji MM, Sheikh UU (2020) Machine learning for plant disease detection: an investigative comparison between support vector machine and deep learning. IAES Int J Artif Intell 9(4):670–683

Deepa R, Moorthy V, Venkataraman R, Kundu SS (2020) Smart farming implementation using phase based IoT system. In: International Conference on Communication and Signal Processing (ICCSP)

Haque FF, Abdelgawad A, Yanambaka VP, Yelamarthi K (2020) Crop yield analysis using machine learning algorithms. In: 2020 IEEE 6th World Forum on Internet of Things (WF-IoT)

Nagaraja GS, Soppimath AB, Soumya T, Abhinith A (2019) IoT based smart agriculture management system. In: 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS). RV College Of Engineering, Bangalore

Puranik V, Sharmila, Ranjan A, Kumari A (2019) Automation in agriculture and IoT. In: 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU). Krishna Engineering College, Ghaziabad.

Varghese R, Sharma S (2018) Affordable smart farming using IoT and machine learning. In: Second International Conference on Intelligent Computing and Control Systems (ICICCS). VIT University, Chennai

Wiangtong T, Sirisuk P (2018) IoT-based versatile platform for precision farming. In: 18th International Symposium on Communications and Information Technologies (ISCIT)

An Analysis of the Effects of the COVID-19 Pandemic on Women's Anxiety and Depression Symptoms

K. Jothimani, Vaishali R. Kulkarni, and S. Punitha

1 Introduction

The COVID-19 infection began in December 2019, in China. The World Health Organization (WHO) has classified the COVID-19 epidemic as a worldwide pandemic and declared it a public medical crisis of international significance. A beta coronavirus called COVID-19 spreads between people when they come into close physical contact (Lotfi et al. 2020). People suffer several other health problems such as chills, cough, coryza, sore throat, trouble breathing, myalgia, nausea, vomiting, etc.

The coronavirus disease has a deep impression on women's insanity, causing increased levels of anxiety and depression symptoms. This is due to a variety of factors, including the fear of getting sick or losing loved ones, financial insecurity, and stress due to immense involvement in the care of children and other family members while also working from home or dealing with job loss. The pandemic has also heightened preexisting mental health challenges for many women, such as those related to domestic violence and sexual harassment. Addressing these issues and providing support for women's mental health are crucial for overcoming the pandemic's effects. Several factors are responsible for the effect of COVID-19 on the problems of women's depression and anxiety. Some of the factors are given below.

- Reduced health quality due to stress and anxiety.

K. Jothimani (✉) · V. R. Kulkarni · S. Punitha
Department of Computer Science Engineering, Graphic Era Deemed to be University,
Uttarakhand, Dehradun 248002, India
e-mail: jothimani.k@geu.ac.in

V. R. Kulkarni
e-mail: vaishali.kulkarni@geu.ac.in

S. Punitha
e-mail: punitha.s@geu.ac.in

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes
in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_14

- Financial instability due to the loss of jobs and business leads to reduced income. Many women have undergone financial insecurity and emotional distress.
- Increased care and other family responsibilities. Women are often the primary caregivers in their families and face additional burdens for taking care of children and other family members.
- Work pressure and challenges due to working from home and lack of social meetings.
- Social isolation gives challenges such as loneliness and other mental health challenges. There have been many cases of domestic violence for women that have caused trauma and emotional distress.
- Several sexual harassment cases have been reported during the pandemic.
- Preexisting mental health challenges are increased due to COVID-19 infection.

There has been a lot of research on the study on how the global pandemic is affecting people psychologically, particularly women. Addressing the effect of COVID-19 on women's anxiety and sadness has attracted several therapies, treatment options, and the use of machine learning paradigms.

Deep learning models have been used for understanding the insanity of COVID-19 (Shorfuzzaman et al. 2021). These models can analyze large amounts of data. Data includes survey responses and social media posts. The different patterns of symptoms can be identified and the likelihood of anxiety in individuals can be predicted. These models can also be used to understand the effect of several other factors, such as age, gender, and income. Convolutional neural network (CNN) is a commonly used deep learning technique that has been used in several cases for classifying irregularities and disease anomalies (Yamashita et al. 2018).

In many applications, decision trees have been successfully used in prediction. The decision tree model is the fundamental method where a model is represented as a binary tree. This method is used for classifying patients suffering from fever due to other infections and COVID-19. The different levels of infection, such as mild, severe, or no infection can also be determined using the SVM classifier. K-nearest neighbors algorithm (KNN) is a simple prediction method where the value of K determines the nearest neighbor data value. The KNN method is used for the classification of datasets based on the attributes.

These algorithms are used to improve symptom detection using the image, age, and several other data parameters and arrive at precise diagnoses. Several graphical-based diagnostic methods are used for classifying the symptoms of a patient. The data used in these methods include CT-SCAN images, chest X-rays, etc. (Zhang 2021).

A CNN algorithm is proposed in this paper for the prediction of anxiety and sadness issues in women. These problems are based on the data obtained in the COVID-19 infection database. This method is compared with four other algorithms from the paradigms of machine learning. The main contributions of this paper are:

1. Obtain cleaned and processed data for patients infected by COVID-19.

2. The patient's records are classified using a CNN and compared with four other classifiers, namely four other algorithms, namely decision trees (DT), KNN, SVM, and ANN.
3. The dataset includes the date, place, age bracket, gender, detected city status, nationality, type of transmission, status change date, source, etc.
4. The prediction results of the algorithms are evaluated for the negative predictive value, f score, accuracy rate, and confusion matrix.

The rest of the paper is organized as: in Sect. 2, past research in the prediction of several COVID-19-related classification methodologies is discussed. The proposed work, dataset, classification algorithms, and dynamics are discussed in Sect. 3. The analysis of results and discussion is given in Sect. 4. Finally, the conclusion and future work is discussed in Sect. 5.

2 Related Works

A novel work is proposed using a new pipeline using recurrent neural networks (LSTM) and convolutional neural networks to accurately identify depressive tweets with 99.42% accuracy in (Javed et al. 2020). Data such as depressive emotions in the tweets and natural language processing, were used for prediction. Between February and May 2020, 571 thousand tweets were analyzed by 482 users, showing a significant increase in depressing tweets.

A deep neural network (DNN) in predicting depression among aged people during COVID-19 is proposed in Nguyen and Byeon (2022). They have considered social factors such as stress levels, health status, daily changes, and physical distancing as factors for the prediction. With the 2020 Community Health Survey data, they trained the DNN model and combined it with an explainable model built with LIME to make the prediction more interpretable. The model was found to have a prediction accuracy of 89.92% with many factors.

Prediction of anxiety and sadness indications along with personal activities among the common people is proposed in the work (Galindo-Vázquez et al. 2020). The authors surveyed over three weeks using non-probability sampling and participants were evaluated using PHQ-9, GAD-7, and a self-care behavior scale. Results showed that 20.8% of the 1508 participants had severe anxiety symptoms and 27.5% had severe depression symptoms.

A psychological effect on survivors of COVID-19, considering both clinical and inflammatory factors is given in Mazza et al. (2020). The study surveyed 402 adult COVID-19 survivors for psychiatric symptoms and found that a substantial number of patients reported symptoms in the psychopathological range for PTSD, depression, anxiety, obsessive–compulsive symptoms, and insomnia. The study found a positive correlation between the systemic immune-inflammation index and the scores for depression and anxiety at follow-up.

Young people's indications related to COVID-19 in the US as per the study given in Liu et al. (2020). The authors have taken online assessments which involved several people aged between, who reported high levels of depression (43.3%), anxiety (45.4%), and PTSD symptoms (31.8%). The results showed that loneliness, COVID-19-related worry, and low distress tolerance were significant predictors of the current scenario with all factors.

Investigation of the effectiveness of various algorithms to predict recovery and death issues by a coronavirus is given in Theerthagiri et al. (2021). The k-fold cross-validation method is used to evaluate the performance of the prediction models, and the prediction accuracy, precision, recall, mean square error, confusion matrix, and kappa score are used as evaluation metrics. The results showed that the KNN classification achieved 80.4% prediction accuracy and outperformed other algorithms by 1.5–3.3%. In terms of deceased cases, the KNN algorithm had a 92% true positive rate and a low misclassification rate of 0.077%.

A model that uses SVM classification to predict COVID-19 infection in individuals is presented by Guhathakurata et al. (2021). The dataset was constructed by thoroughly examining all possible infection conditions based on symptoms. It is challenging to make accurate predictions because individuals without symptoms can be affected by the virus. The symptoms constantly vary and it is difficult to predict. Therefore, only the most common and severe symptoms were considered in the dataset creation process.

A choropleth map for finding the influence of coronavirus among the countries and visualizing it with relevant features (Yadav et al. 2020). Most suitable machine learning algorithms were used to adherent the prediction and accuracy. The authors observed the outcome through the visualization using Matplotlib in Python.

The effect of transfer learning on the classification accuracy of COVID. Patient X-Ray images is presented in Poonam et al. (2021). The authors used several deep learning models such as VGG-16, ResNet-50, Inception V3, EfficientNet, and Transfer Learning in the CNN and applied the learning rates (0.0001, 0.0004). To address overfitting, different learning rates were experimented with. The authors proposed a two-layered model, which achieved an accuracy of 90.45% with reduced classification time.

3 Materials and Methods

The proposed system classifies women suffering from anxiety and depression based on severity levels. The dataset is divided into three different subsets called training, testing, and validation set. A three cross-validation scheme has been applied using the feature subsets that are derived from the data collected. Five different machine learning algorithms, such as DT, KNN, SVM, ANN, and feed-forward were used for classification.

3.1 Dataset

This proposed system focused on detecting anxiety, and depression symptoms of COVID-affected women. Part of the data was collected from different participants via Google forms. The rest of the data was collected from various online resources, public surveys, and online questionnaires. The independent variables collected for the classification are given in Table 1. The collected data is further classified using five machine learning algorithms. After data is collected, the data is encoded using integer values of 0–2. Once the data is encoded, they were labeled based on the severity as seen in Table 2. For variables such as ES, CID, FIL, SSL, and LSL, the values are encoded as 0 for low, 1 for medium, and 2 for high. For the BMI variable, 0 represents normal, 1 represents overweight, and 2 represents obese.

Table 1 Independent variables used in the proposed classification system

Variable name	Description	Value	Label
COVID symptoms	COVID symptoms	Mild, severe	Feature1
Age	Age	10–15 years	Feature 2
ES	Economic status	Low, medium, or high	Feature 3
LP	Living place	Urban or nonurban	Feature 4
BMI	Body mass index	Normal, overweight, or obese	Feature 5
Smoking	Tobacco risk	Yes or no	Feature 6
CID	COVID-19-infected duration	Low, medium, or high	Feature 7
PAL	Physical activity	Low, medium, or high	Feature 8
FIL	Family income level	Low, medium, or high	Feature 9
SSL	Stress symptoms level	Low, medium, or high	Feature 10
DSL	Depression symptom level	Normal or abnormal	Feature 11
LSL	Life satisfaction level	Low, medium, or high	Feature 12
Anxiety	Anxiety symptoms level	Normal or abnormal	Feature 13

Table 2 Severity levels

Severity	Anxiety	Depression
Low	0–5	10–15 years
Medium	6–9	Low, medium, or high
High	10–15	Urban or nonurban
Normal	18.5–24.9	18.5–24.9
Overweight	25.0–29.9	25.0–29.9
Obese	30.0 or higher	30.0 or higher

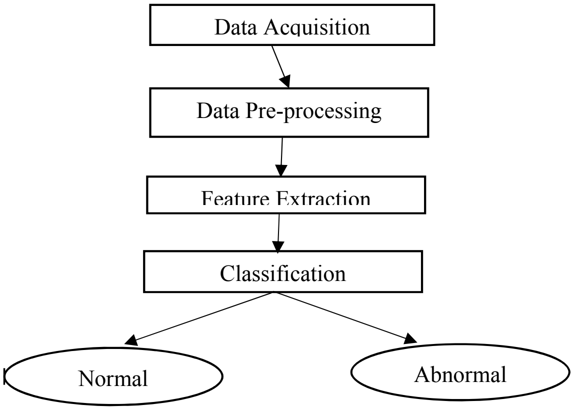


Fig. 1 Proposed anxiety and depression classification system

3.2 Proposed Methodology

The proposed classification system analyzes and classifies the anxiety and depression data of the COVID-19 disease. The classification process proposed in this paper consists of steps as mentioned below.

- Data Collection: Collect relevant data on women's experiences relevant to anxiety and depression.
- Data Preprocessing: Clean and preprocess the data which includes removing missing values, duplicates, and irrelevant information.
- Feature Extraction: Extracting relevant features from the data that can be used to predict anxiety levels, such as demographics, health status, job status, and social support.
- Model Selection: Appropriate classification algorithm must be adopted.
- Training and Validation: Train the model using a portion of the data and evaluate its performance on a separate validation set. Tune the model's hyperparameters to optimize its performance.
- Deployment: Use the trained model for predicting the new data, such as the anxiety levels of women. Figure 1 describes the steps involved in the proposed classification system.

3.2.1 Decision Tree (DT)

The proposed system used a DT classifier. Decision trees are involved in the classification process using flowchart-like structures called decision trees. Those trees consist of different nodes where each node represents an examination of instances based on specific criteria with the help of attributes. Each instance of the dataset is checked against the nodes and sorted from the topmost till the leaf node is reached. The top node of the decision tree corresponds to the best predictor node. Each node is associated with a binary division. The dataset used in the proposed classification system

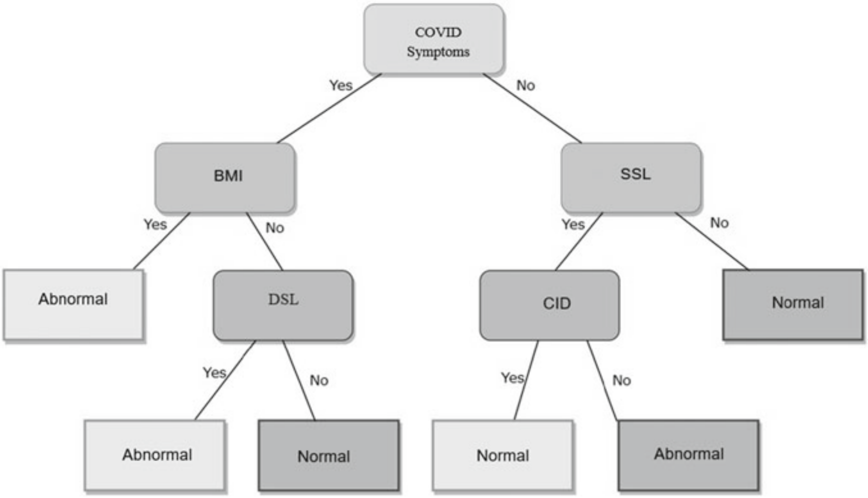


Fig. 2 Decision tree classifier used by the proposed system

is partitioned into smaller subsets incrementally using the J48 decision tree classifier. Information gain is calculated for each attribute and the attribute that has high information gain is involved in the prediction process. The pictorial representation of a decision tree classifier used by the proposed system is shown in Fig. 2.

3.2.2 K Nearest Neighbor (KNN)

The proposed system used a K-nearest neighbor classifier which is a supervised learning-based classification algorithm. The steps involved in the KNN classification of the proposed system is explained using Algorithm 1.

Algorithm 1: KNN classification process.

```
1: Select K neighbors.
2: Estimate the Euclidean distance of K neighbors.
3: Consider K nearest neighbors using the estimated Euclidean distance.
4: Count the data points in every class among the k selected neighbors.
5: Classify the new instance to the category in which the neighbors are maximum.
```

3.2.3 Support Vector Machine (SVM)

The proposed system uses an SVM which is a supervised learning-based classification system. It separates the instances using an optimal line called decision boundary

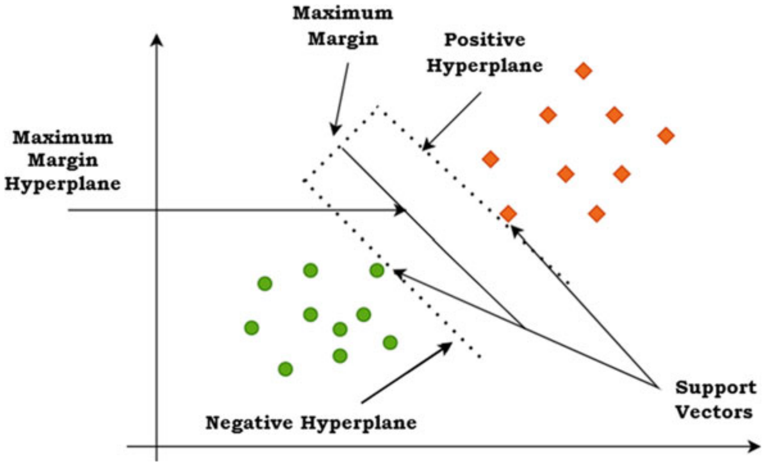


Fig. 3 SVM classifier

in an n-dimensional space into different classes. Each instance is classified into its correct category using the decision boundary. This optimal decision boundary is otherwise called a hyperplane. The vectors that help in building the hyperplane are called support vectors. The pictorial representation of a support vector machine is shown in Fig. 3.

3.2.4 Artificial Neural Network (ANN)

The proposed system used an ANN classifier for the diagnosis of anxiety and depression symptoms in COVID-affected women. A neural network consists of artificial neurons built on different layers in sequence. The input layer accepts inputs in various formats depending on the application. The hidden layer is built as the middle layer. It calculates hidden patterns available in the data instances. The inputs fed are transformed in the hidden layer and the final prediction is forwarded. The ANN estimates the weighted sum of the inputs along with a bias using Eq. 1. The pictorial representation of a support vector machine is shown in Fig. 4.

$$\sum_{i=1}^{n} W_i * X_i + b \tag{1}$$

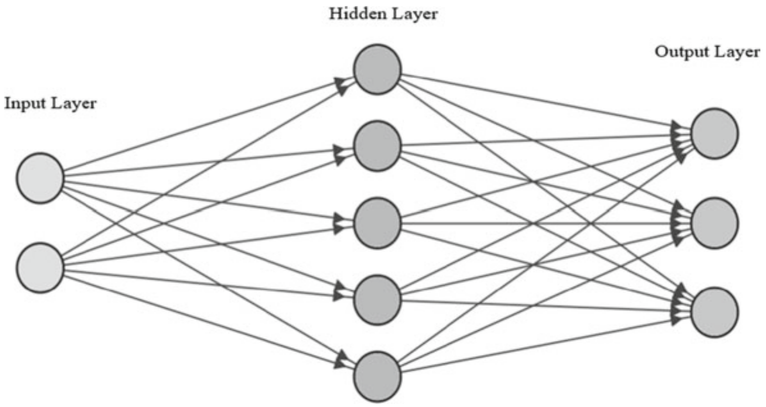


Fig. 4 ANN architecture

3.2.5 Convolution Neural Network (CNN)

The proposed system used a CNN which is a deep learning architecture utilized for classification tasks. It consists of convolutional layers, pooling layers, and fully connected layers. The input layer gets the raw input. The convolution layer calculates the output volume. The activation function layer applies the activation function to each element. The pool layer is inserted in the converts to reduce the volume size, to make computation fast, to reduce memory and to avoid overfitting. Two of the pooling layers are max pooling and average pooling. The pictorial representation is shown in Fig. 5.

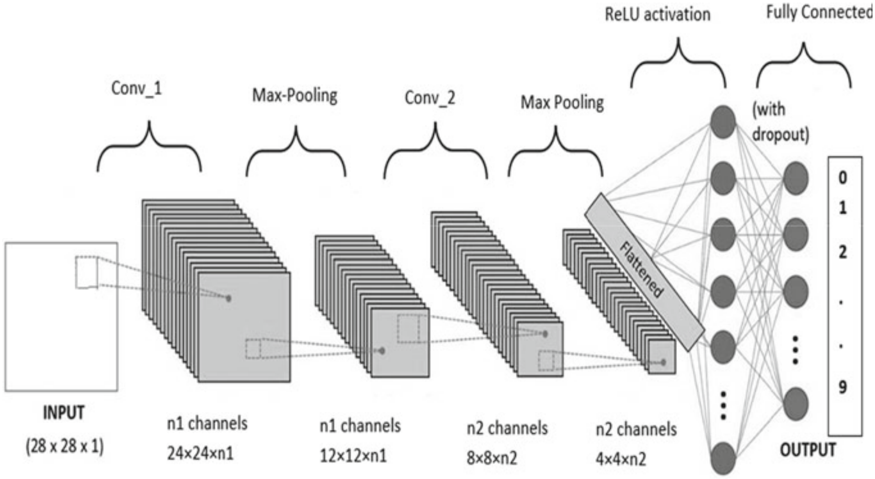


Fig. 5 Convolution neural network

4 Performance Analysis and Discussion

The performance of the proposed system is evaluated in terms of sensitivity, specificity, accuracy, precision, negative predictive value, and F-measure. The true positive, true negative, false positive, and false negative values are calculated. Decision tree, support vector machine, KNN, ANN using MLP, and CNN architecture were used for classification. The feature set used for classification is given in Table 1. The implementation of KNN, SVM, and DT uses default training parameters. ANN and CNN parameter settings are given in Table 3.

The performance metrics used to evaluate the proposed classification system are listed in Table 4. The sensitivity and specificity of CNN are high followed by ANN, SVM, DT, and KNN. The proposed classifier system has been tested using all the features collected.

CNN when evaluated under different iterations attained the highest accuracy for ten runs which is 95.54% when compared with other machine learning algorithms. Followed by CNN, the ANN attained the next highest which is 91.05%. Next to

Table 3 Parameter settings of ANN and CNN classifier

Parameter	Value
ANN-training	Backpropagation
Input nodes size	14 (full feature size)
Output nodes size	2 (normal, abnormal)
Initial weights	Random
Hidden node size	4
Activation function for hidden node	Hyperbolic tangent
Activation function for the output node	Pure linear
CNN-Number of layers	4
CNN-Convolution kernels	64
Training set samples	450 (50%)
Validation set samples	225(25%)
Testing set samples	225(25%)

Table 4 Performance of the proposed classification system

Metrics	DT	KNN	SVM	ANN	CNN
Sensitivity (%)	85.08	82.02	88.17	90.26	95.02
Specificity (%)	84.81	81.54	87.63	90.01	94.88
Accuracy (%)	85.46	82.58	88.52	91.05	95.54
Precision (%)	85.69	82.23	88.38	91.00	95.12
Negative predictive value (NPV) (%)	85.45	82.81	88.90	90.09	95.45
F-measure	0.8538	0.8243	0.8877	0.9102	0.9532

ANN, the SVM classifier attained an accuracy of 88.52% when evaluated under various iterations. Next to SVM, the accuracy of the decision tree is 85.46% and 82.58% for the KNN classifier. A comparison of the machine learning models used in the proposed classification system is shown in Fig. 6.

The confusion matrix of the proposed classification system is given in Table 5. A total of 400 abnormal cases and 500 normal cases were considered for analysis and suitable values of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) were noted.

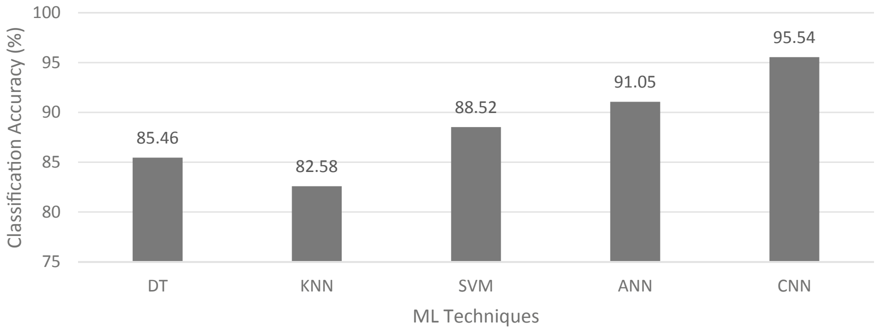


Fig. 6 Comparison of the machine learning models used in the proposed system

Table 5 Confusion matrix of the proposed classification system

Methods	Actual	Total cases	Predicted outcome	
			Abnormal	Normal
DT	Abnormal	400	350 (TP)	50 (FN)
	Normal	500	20 (FP)	480 (TN)
KNN	Abnormal	400	335 (TP)	65 (FN)
	Normal	500	20 (FP)	480 (TN)
SVM	Abnormal	400	382 (TP)	18 (FN)
	Normal	500	10 (FP)	490 (TN)
ANN	Abnormal	400	390 (TP)	10 (FN)
	Normal	500	10 (FP)	490 (TN)
CNN	Abnormal	400	395 (TP)	5 (FN)
	Normal	500	5 (FP)	495 (TN)

5 Conclusion

The efficiency of five different machine learning algorithms such as DT, KNN, SVM, ANN, and CNN is compared for diagnosing anxiety and depression in COVID-19 patients in this paper. The strength of the proposed work lies in using a simple wrapper approach. The performance has been analyzed using various performance metrics. It has been studied that CNN attained higher when compared with other models. The same dataset can be tried on various deep learning models along with hyperparameter tuning. This can be considered as future work to prove better results.

References

Galindo-Vázquez O, Ramírez-Orozco M, Costas-Muñiz R, Mendoza-Contreras LA, Calderillo-Ruíz G, Meneses-García A (2020) Symptoms of anxiety and depression and self-care behaviors during the COVID-19 pandemic in the general population. Gac Med Mex 156(4):294–301. https://doi.org/10.24875/GMM.M20000399

Guhathakurata S, Kundu S, Chakraborty A, Banerjee JS (2021) A novel approach to predict COVID-19 using support vector machine. In: Data Science for COVID-19. Academic Press, New York, pp 351–364. https://doi.org/10.1016/B978-0-12-824536-1.00014-9

Javed B, Sarwer A, Soto EB, Mashwani ZU (2020) The coronavirus (COVID-19) pandemic's impact on mental health. Int J Health Plann Manage 35(5):993–996. https://doi.org/10.1002/hpm.3008. Epub 2020 Jun 22

Liu CH, Zhang E, Wong GTF, Hyun S (2020) Factors associated with depression, anxiety, and PTSD symptomatology during the COVID-19 pandemic: Clinical implications for US young adult mental health. Psychiatry Res 290:113172. https://doi.org/10.1016/j.psychres.2020.113172

Lotfi M, Hamblin MR, Rezaei N (2020) COVID-19: transmission, prevention, and potential therapeutic opportunities. Clin Chim Acta 508:254–266. https://doi.org/10.1016/j.cca.2020.05.044. Epub 2020 May 29

Mazza MG, De Lorenzo R, Conte C, Poletti S, Vai B, Bollettini I et al (2020) Anxiety and depression in COVID-19 survivors: role of inflammatory and clinical predictors. Brain Behav Immun 89:594–600. https://doi.org/10.1016/j.bbi.2020.07.037

Nguyen HV, Byeon H (2022) Explainable deep-learning-based depression modeling of elderly community after COVID-19 pandemic. Mathematics. https://doi.org/10.3390/math10234408

Shorfuzzaman M, Masud M, Alhumyani H, Anand D, Singh A (2021) Artificial neural network-based deep learning model for COVID-19 patient detection using X-ray chest images. J Healthcare Eng 2021:5513679. https://doi.org/10.1155/2021/5513679

Theerthagiri P, Jeena Jacob I, Usha Ruby A, Yendapalli V (2021) Prediction of COVID-19 possibilities using KNearest Neighbour classification algorithm. Int J Curr Res Rev Res 13:06. https://doi.org/10.31782/IJCRR.2021.SP173

Verma P, Tripathi V, Pant B (2021) Comparison of different optimizers implemented on the deep learning architectures for COVID-19 classification. Mater Today Proc 46:11098–11102. https://doi.org/10.1016/j.matpr.2021.02.244

Yadav D, Maheshwari H, Chandra U, Sharma A (2020) COVID-19 analysis by using machine and deep learning. Springer, Singapore. https://doi.org/10.1007/978-981-15-8097-0_2

Yamashita R, Nishio M, Do RKG et al (2018) Convolutional neural networks: an overview and application in radiology. Insights Imag 9:611–629. https://doi.org/10.1007/s13244-018-0639-9

Zhang F (2021) Application of machine learning in CT images and X-rays of COVID-19 pneumonia. Medicine (Baltimore) 100(36):e26855. https://doi.org/10.1097/MD.0000000000026855

An Exploration of State-of-Art Approaches on Low-Light Image Enhancement Techniques

V. S. Anila, G. Nagarajan, and T. Perarasi

1 Introduction

Computer vision is an area of artificial intelligence (AI) that enables computers and systems to extract useful information from digital images. The quality of the image will depend on a number of factors, including illumination, contrast and brightness. Images that are captured in an environment having low illumination or low light are categorized as low-light images. In many real-time applications, this low-light condition may occur. So, to overcome this, many low-light image enhancement methods are used. This survey paper's main goal is to investigate the various image improvement techniques for low-light images. Image enhancement is a technique that helps to improve the quality of an image. The parameters that define the image quality are color, contrast, brightness, illumination, etc. During the image acquisition, sufficient light intensity is needed. If the light intensity is low, the captured image will give less information than the original image. In many applications, there is a possibility of low-light conditions. It is necessary to create an enhancement method that is more suited for low-light images in order to get around this. The popular low-light image enhancement methods are Gamma transformation, Histogram equalization, Retinex methods, machine learning and deep learning methods. In recent years, the availability of various learning models introduces a large exploration of low-light image enhancement methods. This survey paper divides the algorithm into two classes, traditional methods and learning-based methods. This learning-based algorithms are again classified into machine learning-based and deep learning-based

V. S. Anila (✉) · G. Nagarajan
Department of ECE, Puducherry Technological University, Pillaichavadi, India
e-mail: vsanila@pec.edu

G. Nagarajan
e-mail: nagarajanpec@ptuniv.edu.in

T. Perarasi
Department of ECE, Bannari Amman Institute of Technology, Sathyamangalam, India

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_15

methods. Section 2 describes a few existing low-light applications. Section 3 explains the classification of enhancement methods.

2 Low-Light Images

Medical image processing has been widely used in research in recent years to diagnose a variety of disorders. When considering various medical imaging techniques, this low-light environment could affect the accuracy of the diagnosis. One of the most important methods for identifying abnormalities of the larynx is laryngeal endoscopy. Due to the anatomical structure of the human body, it is difficult to get illuminated images of this region. As a result, low-light images are obtained.

This enhancement scheme is also applicable for the enhancement of chest x-ray for the detailed analysis of Covid-19 cases. Figures 1, 2 show the larynx endoscopy image and chest x-ray image. Night traffic monitoring is a major challenge in today's world. These types of enhancement algorithms are useful for improving the analysis of monitoring systems. The other important areas where this low-light condition may exist are underwater images, foggy images, satellite images, etc. (Figs. 3, 4).

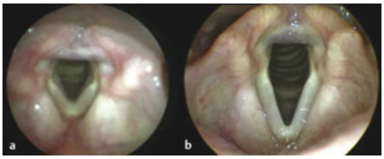


Fig. 1 Larynx endoscopy image

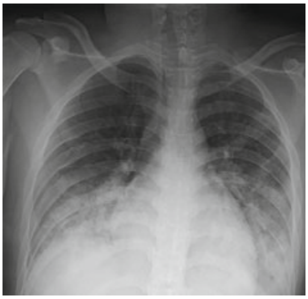


Fig. 2 Covid-19 chest x-ray image

Fig. 3 Underwater image

Fig. 4 Foggy image

3 Methodologies

This survey paper introduces a distinction between traditional and learning-based low-light image enhancement technique (Fig. 5).

The traditional methods are Gamma transformation, Histogram equalization and Retinex-based methods. The learning methods are machine learning (ML) and deep learning methods (DL). Methods based on machine learning have only recently become available. Machine learning is a subset of artificial intelligence. They are capable of learning by themselves without being explicitly programmed. The limitations of ML algorithms are, they require supervision for feature extraction and handle only thousands of data points. Commonly preferred ML algorithms are principal component analysis (PCA), regression, support vector machine (SVM), etc.

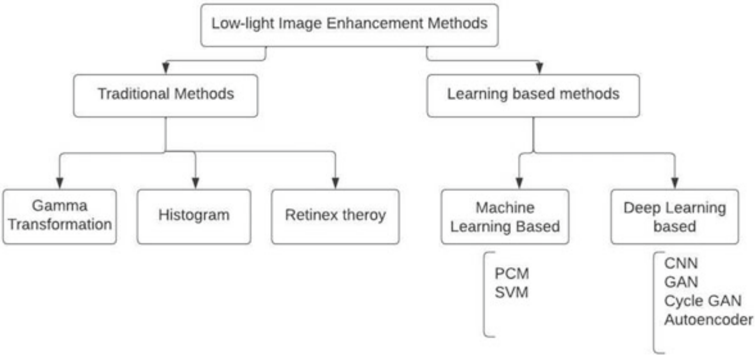


Fig. 5 Classification of low-light image enhancement techniques

Several deep learning-based image enhancement methods have also emerged since 2016. DL is a subset of the ML algorithm. Millions of data points are processed by DL algorithms. As a result, a large number of features are extracted without supervision. Convolution neural networks (CNNs) have been used as the foundation of deep learning frameworks in a variety of research papers. Deep learning-based methods can achieve excellent results in low-light image enhancement. Section 3.4 describes about deep learning algorithms.

3.1 Gamma Transformation

A Gamma function is a nonlinear transformation. Gamma correction is a technique used for image enhancement.

$$g(x, y) = f(x, y)^{\surd} \tag{1}$$

where '$\surd$' represents the gamma correction parameter. By varying the parameter, several different transformation curves can be obtained. When '$\surd$' > 1, the transformation will broaden the dynamic range of the low-gray value areas of the image and compress the range of the high-gray value areas. When '$\surd$' < 1 the transformation will have the low gray values and stretch the high gray values. When '$\surd$' = 1 output remains unchanged (Fig. 6).

A pair of complementary gamma functions by fusion is one of the methods used for low-light image enhancement (Li et al. 2020). The pair of complementary functions are as follows,

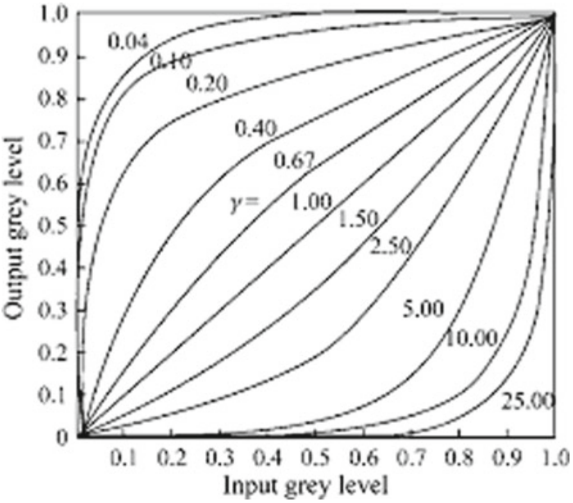


Fig. 6 Gamma transformation

$$y_1 = 1 - \left(1 - (x)\surd\right) \tag{2}$$

$$y_2 = (1 - (1 - (x)))^{1/\surd} \tag{3}$$

where x—input pixel value, y_1 and y_2—transformed output pixels.

The input red, green, blue (RGB) image is transformed into a hue, saturation, value (HSV) image. The brightness of the image is determined by the value component (V), which depends on the amount of light intensity present in the environment. The value component is enhanced by the above transformation equations. Then two enhanced 'V' components are combined by,

$$I_1 = c_1 y_1 + c_2 y_2 \tag{4}$$

where $c_1 = V_i / \sum V_i$.

"I_1" is the first input for the fusion process. The identical value component is subjected to sharpening and histogram equalization to produce the second input for the fusion. The second input for fusion is,

$$I_2 = (V + 2H(V) - G * H(V))/2 \tag{5}$$

The value components I_1 and I_2 are fused by the image fusion process. This overall process improves the brightness of the low-light images by adjusting the dark region and compressing the bright region. The advantage of using this gamma function is that it generates even brightness.

3.2 Histogram Equalization

Histogram equalization (Narendra and Fitch 1981; Abdullah-Al-Wadud et al. 2007) is one of the traditional methods for low-light image enhancement. The pixels are the basic building blocks of an image. Each pixel holds a specific intensity value. The histogram is a plot that shows the number of pixels versus their intensity values. The histogram equalization algorithm uses the cumulative distribution function (CDF) to adjust the output gray level to have a uniform distribution (Fig. 7).

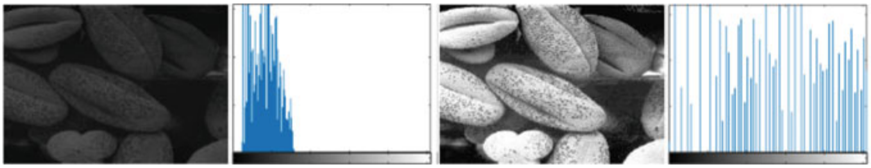


Fig. 7 Example of histogram equalization

'I_1' will serve as the input image, and 'L' will serve as the gray value. 'N' is for the overall number of pixels in a picture, '$I(i, j)$' stands for the gray value at the point with coordinates (i, j), and 'n_k' stands for the number of pixels at gray level k. The likelihood that a specific gray level 'k' will occur is,

$$P(k) = n_k/N; \quad \text{where } k = 0, 1, \ldots, L-1 \tag{6}$$

The cumulative distribution function (CDF) of the gray level of an image 'I' is given by,

$$C(k) = \sum_{0}^{k} p(r); \quad k = 0, 1, \ldots, L-1 \tag{7}$$

The histogram equalization algorithm maps the original image to an enhanced image with a uniform gray-level distribution based on CDF (Table 1). The enhanced output image is represented as follows:

$$f(k) = (L-1) * C(k) \tag{8}$$

3.3 Retinex Theory

Retinex theory (Land 1977) is one of the major strategies employed in low-light image enhancement. As per the Retinex theory, the observed image is represented as the product of reflectance and illumination component (Fig. 8).

As per Retinex theory,

$$S(X, Y) = R(X, Y) * L(X, Y) \tag{9}$$

where

$S(X, Y)$—Observed image,

$R(X, Y)$—Reflectance component,

$L(X, Y)$—Illumination component.

A low-light image is characterized as it is captured in a low illuminance region. Illuminance is the measure of how much incident light illuminates the surface. For images taken in dim lighting, illuminance is below the standard level. As per the Retinex theory, the reflectance component is considered as the enhanced image, $R = S/L$. By choosing the proper illumination map, the required enhanced image is obtained. Most of the research work is carried out based on this equation. By

Table 1 Different histogram methods

Histogram method	Findings
Equal area Dualistic Sub-image Histogram Equalization (DSIHE) (Wang et al. 1999)	The input images are decomposed into two equal area sub-images based on its original probability density functions
Minimum Mean Brightness Error Bi-histogram Equalization (MMBEBHE) (Chen and Ramli 2003)	This method preserves the maximum brightness of the image
Adaptive Histogram Equalization (AHE) (Megha et al. 2016)	It deals with contrast restoration for medical and other unclear images
Partially Overlapped Subblock Histogram Equalization (POSHE) (Ganesan and Rabbani 2019)	This method separates subblock images recursively into different sub-images with the cumulative density function (CDF)
Contrast Limited Adaptive Histogram Equalization (CLAHE) (Yadav et al. 2014)	This method partitions the images into contextual regions and applies the histogram equalization to each region. It is a specially developed algorithm for medical images
Recursive Mean Separate Histogram Equalization (RMSHE) (Chen and Ramli 2013)	This is a generalized model of Bi-histogram equalization. This method is used to provide better and scalable brightness preservation
A Recursive Sub-image Histogram Equalization (RSIHE) (Sim et al. 2007; Singh 2014)	It is developed to overcome the drawback of generic histogram equalization for grayscale images and it provides better image compensation
An Entropy-based Dynamic Sub-histogram Equalization (EDSHE) method (Parihar and Verma 2016)	Recursive division of the histogram based on the entropy of the sub histograms is performed. Each sub-histogram is divided recursively into two sub-histograms with equal entropy. A dynamic range is allocated to each sub-histogram based on entropy
A Dynamic Histogram Equalization (DHE) (Abdullah-Al-Wadud et al. 2007)	The image histogram is partitioned based on local minima and a unique gray value range is assigned to each partition
A Fuzzy-based Brightness Preserving Dynamic Histogram Equalization (BPDHE) (Sheet et al. 2010)	It is used to reduce computational complexity, in which execution time is dependent on image size and nature of the histogram
Bi-histogram Equalization with a Plateau Limit (BHEPL) (Ooi et al. 2009)	It is preferred for short processing time image enhancement. It divides the input histogram into two independent sub-histograms to maintain the mean brightness
A Median Mean-based Sub-image clipped Histogram Equalization (MMSICHE) (Singh and Kapoor 2014)	To improve the brightness level, information content (entropy) and better enhancement rate

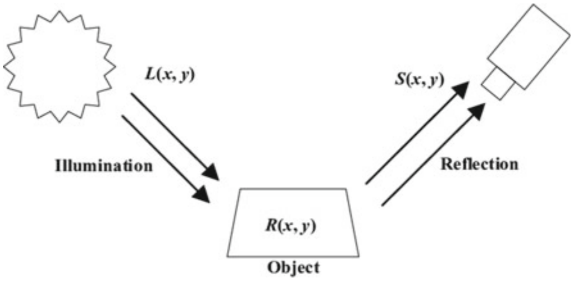


Fig. 8 Retinex model

using the illumination component, the enhanced outputs are obtained by performing division operations. To overcome the difficulty in this division operation an inverse term is used. The inverse term is expressed in the given equation,

$$R = S * L^{-1} \tag{10}$$

Using an inverse illumination map (L^{-1}), the enhanced image (R) is obtained. Many of the deep learning model uses this Retinex theory as the basic theory. As per the theory, illumination map is constructed by various CNN models. Current research works are carried out in deep learning without Retinex theory also. Deep learning models will play an important role in the enhancement of low-light images.

3.4 Deep Learning-Based Methods

Deep learning has been applied to computer vision tasks such as low-light image enhancement in recent years due to its excellent representation and generalization abilities. Many deep learning models use Retinex theory for their operation. A convolutional neural network (CNN) is a deep learning network architecture that learns directly from data. CNNs are especially useful for detecting patterns in images in order to recognize objects, classes and categories.

Figure 9 shows the basic architecture of convolutional neural network (CNN). The function of the convolution layer is to extract meaningful information by applying a sliding window on the input matrix. The pooling layer reduces the height and width while maintaining the depth information to conduct dimensionality reduction. Based on the application, different types of pooling are preferred. These are maximum pooling, average pooling and minimum pooling. Fully connected layer will perform the classification.

A generative adversarial network (GAN) (Goodfellow et al. 2014) is an unsupervised deep learning-based model. It uses unlabelled data for training. GAN contains two competing neural networks called generator and discriminator, which compete against one another and may evaluate, discover and follow variations within the

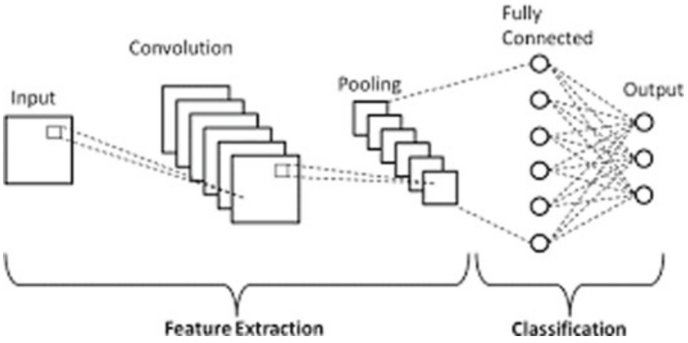


Fig. 9 Convolutional neural networks

dataset. The generator generates fake samples of images and tries to fool the discriminator. During the training phase, the generator and discriminator run in competition with each other. The model is trained to function more effectively during each epoch.

A Retinex-based attention network (Huang et al. 2020) uses Retinex as the basic theory for the learning of deep neural networks. This technique calculates an improved image from a reflectance map. Illumination extraction block is developed using an attention mechanism module, resulting in an illumination map prediction network. In order to gain more precise illumination information for the input image, this attention technique is inserted between the convolution layer and batch normalization. On both low illumination images with uniform light and uneven illumination, this model lessens the impact of noise and the augmented information that results.

A Multiscale Attention Retinex Network (MARN) (Zhang and Wang 2021) is designed to predict a detailed inverse illumination map of the input image. When compared with various CNN algorithms, the Multiscale Attention Retinex Network gives better feature extraction. This MARN improves the generalization capability of the network. Instead of using more image priors, an illumination attention map is used to learn the model. It improves the quality of the image in various lighting conditions. This utilizes reconstruction loss, structure similarity loss and detail loss. If the inverse illumination is predicted, the reflectance map is calculated by using Retinex theory and then this reflectance map is estimated as an enhanced image.

A simple generative adversarial network with a Retinex model (Ma et al. 2021), a decomposition network is used to decompose the low-light image into illuminance and reflection maps. For training the GAN structure unpaired datasets are used. This provides a better generalization to the model. By using this structure, reduced training complexity and reduced training time is achieved. This model is applied to mobile phones with small memory.

An enlighten GAN is a modified GAN structure (Jiang et al. 2021). It introduces Enlighten GAN structure that can be trained without image pairs. Even with unpaired datasets, this structure is generalized very well for various real-time images. This model introduces a global and local discriminator structure that handles spatially varying light conditions in the input image. The results of Enlighten GAN are compared with several state-of-art methods. All results show the superiority of Enlighten GAN.

Various approaches have been used to improve image segmentation (Long et al. 2015). Segmentation is the process of dividing an image into its various parts. These basic operations are performed in many computer vision tasks. Segmentation shows good performance during daytime or in bright light. In the case of low-light images, segmentation is not performed well because of the presence of noise, blurredness, etc. The process of segmentation can be divided into a single-class and multi-class segmentation. In single-class segmentation (Wang and Ren 2018), only one object or one feature is considered for segmentation. In multi-class segmentation (Dai and Gool 2018), multiple features are considered. In Cho et al. (2020), semantic segmentation of low-light images with modified Cycle GAN is introduced. The modified Cycle GAN is trained using paired dataset and the L_1 loss function is added to the existing Cycle GAN for improving the performance of the segmentation.

Table 2 summarizes the low-light image enhancement techniques.

4 Conclusion

Various state-of-art methods are discussed in this paper for low-light image enhancement. Many of the deep learning structures use Retinex as the basic theory of operation. The illumination map is modified by using various learning architectures, CNN, GAN, Cyclic GAN, etc., which are a few illustrations of deep learning models. This survey presents some works which are more suitable in a noisy environment also. In many real-time applications, low-light conditions may occur due to the unavailability of environmental light. Low-light image enhancement thus plays a crucial role in each of these scenarios. Low-light image enhancement can be extended to the enhancement of low-light video also.

Table 2 Low-light image enhancement techniques

References	Year	Design methodology	Inferences
Singh and Bhandari (2021)	2021	• It is a machine learning algorithm • The RGB input image is converted to hue, saturation, value (HSV) space • The reflection coefficient of the ‘V’ component is obtained • Image fusion is performed by principal component analysis (PCA) method • Contrast Limited Adaptive Histogram Equalization (CLAHE) is applied to fused image • The fused image is converted to RGB plane	The irregular brightness due to low illumination can be upgraded by this method
Liang et al. (2021)	2021	• It includes a two-stage CNN • Image Signal Processing pipeline (ISP) operations are grouped into the restoration group and the enhancement group • A CNN model called CAMERANET is designed with two sub-networks to perform the two groups of subtasks	It uses a framework for deep learning-based ISP pipeline design. The effect of noise is reduced
Zhu et al. (2021)	2021	• In the camera, a unit called an image signal processing pipeline (ISP) is used to generate JPEG format images • Denoising, demosaicing, detail enhancement and white balance of noises are all included in ISP • A two-stage network has modified the conventional ISP	The image-denoising module U net helps to reduce the effect of noises

(continued)

Table 2 (continued)

References	Year	Design methodology	Inferences
Cho et al. (2021)	2021	• Semantic segmentation is very difficult during the night-time due to the insufficient amount of external light • So, it is unable to detect the shape and color information of objects • The Cycle GAN structure is used for getting improved segmentation of low-light images	This architecture improves the segmentation performance by preserving the shape and color information of the images
Wang et al. (2020a)	2020	• It generates residual between low and normal images by using a deep neural network • A lightening backpropagation is used to lighten the dark images by predicting their illumination	By using the lightening backpropagation module better enhancement is achieved
Lu and Zhang (2021)	2021	• According to the Retinex theory, the illumination component is calculated • The two-branch exposure fusion network aimed to produce adaptable enhancement in various illuminations • SSIM, L_1 and L_2 losses are considered for better performance	It provides better enhancements in noisy environments
Lim and Kim (2021)	2021	• A Laplacian pyramid is a type of image representation in which the DSLR technique is used to adjust global illumination and restores the local details • These two tasks are performed with separate encoder-decoder architecture	This architecture produces better enhancement for local details without any color distortions

(continued)

Table 2 (continued)

References	Year	Design methodology	Inferences
Ravirathinam et al. (2021)	2021	• An encoder-decoder architecture is used for image enhancement • The function of the encoder is to extract the features from the images • The decoder enhances the input images with the help of extracted features	The multi-context feature extraction modules help to extract complex features Three loss functions are used here for better enhancement
Lu et al. (2020)	2020	• In the selfie image enhancement, background and foreground are separately enhanced and combined together • Modules, such as gain estimation and raw data processing are used • The gain estimation module separates the foreground and background of input images and predicts their gain	Effectiveness and robustness of the deep selfie images are obtained
Lamba et al. (2021)	2021	• The light fields are used for refocussing and depth estimation of images • For low-light images, it is difficult to get these light fields • L3FNet is a DNN architecture used for low-light field restoration	Better optimized results are achieved
Wang et al. (2022)	2022	• A mixed attention-guided generative adversarial network called MAGAN for low-light image enhancement is designed • It uses an unsupervised learning method • A mixed attention module called feature attention and pixel attention is used	Noise reduction is achieved

(continued)

Table 2 (continued)

References	Year	Design methodology	Inferences
Guo et al. (2017)	2017	• It is based on Retinex theory • The illumination map is calculated by finding the maximum intensity of each pixel in R, G and B components • From this an illumination map is constructed, where an augmented Lagrangian multiplier-based algorithm is used	A consistent illumination map is generated. Like traditional Retinex theory input image is not divided into reflection and illumination
Wang et al. (2020b)	2020	• It works well with low-light noisy images • It uses non-subsampled Shearlet Transform (NSST) to decompose the input image into low-pass subband and bandpass subband • For noise reduction, bandpass subband is suppressed	By this method, better noise reduction with good adaptability and good stability are achieved
Guo et al. (2020)	2020	• This model uses regularized illumination optimization and deep noise suppression • Gamma correction is used for obtaining an illumination map • A guided filter-based detail boosting is introduced to optimize the reflection map • These two images are combined to get enhanced images	It utilizes regularized illumination optimization and deep blind denoising. Maritime images are used in this method
Garg et al. (2022)	2022	• LiCENt—Light Channel Enhancement Network that uses a combination of an autoencoder and CNN • The RGB image is converted to HSL space and improves the 'L' value of the HSL image	The learnable parameter is reduced by a factor of 8.92. It avoids over-enhancement and color distortion. This is a lightweight and fast method

(continued)

Table 2 (continued)

References	Year	Design methodology	Inferences
Guo et al. (2019)	2019	• This architecture is based on Multiscale Retinex theory (MSR) which is considered as a CNN unit • Its output is given to perform a discrete wavelet transform to get better output • A pipeline neural network called LLIE-net is designed and it is trained using paired datasets	Better noise reduction. The pipeline network is designed such that in the prediction phase no artificial parameter is considered. Better results for real-time data
Park et al. (2018)	2018	• It is a Retinex-based model • Dual autoencoders are used • A low illumination component results in an enhanced reflectance component • The illumination component is blurred by using a stacked autoencoder unit • The improved reflectance component results an enhanced image • A convolutional autoencoder network is used to prevent noise amplification	This is applicable to robot vision and visual surveillance applications
Li et al. (2019)	2019	• An adaptive bright color channel-based enhancement method and a denoising method are formulated • A bright channel model and deep convolutional neural network are used for denoising and color correction	This scheme removes the electrical noise and modifies the color distortions of the radiometric compensation in water

(continued)

Table 2 (continued)

References	Year	Design methodology	Inferences
Ren et al. (2019)	2019	• A traditional Retinex model with a camera response model is combined • The exposure ratio for each pixel is estimated by the illumination estimation technique • As per the exposure ratio, the camera response model adjusts each pixel to the desired exposure • A less distorted enhanced image is obtained	Response characteristics of the camera is utilized to enhance the illumination. This model reduces color and lightness distortion
Ma et al. (2019)	2019	• This algorithm utilizes three stages, image reconstruction, image enhancement and color distortion • The RGB image is converted to an HSI image then low-light image enhancement via illumination map estimation (LIME) is used to enhance the reconstructed grayscale map. Retinex theory is used for illumination estimation	The effect of Halo and noise in the input image have been reduced. The modified color distortion method is designed. It is more applicable for edge detection, feature mapping, object recognition and tracking
Yang et al. (2021)	2021	• The sparse gradient minimization sub-network (SGM-Net) is designed, which extracts paired illumination maps. Two sub-networks namely enhance net and restore net are used for contrast enhancement and noise reduction	Reduces the effect of noise and preserves edge information
Wang et al. (2018)	2018	• The Global Illumination Aware and Detail Preserving Network (GLADNet) is formulated. An input image is rescaled to a certain size and using an encoder-decoder network a global illumination is generated	More vivid and natural results are obtained. GLADNet adjusts the whole image at the same time, so over exposure in brighter region and under exposure in darker regions can be avoided

References

Abdullah-Al-Wadud M, Kabir MH, Akber Dewan MA, Chae O (2007) A dynamic histogram equalization for image contrast enhancement. IEEE Trans Consum Electron 53(2):593–600. https://doi.org/10.1109/TCE.2007.381734

Chen SD, Ramli AR (2003) Contrast enhancement using recursive mean-separate histogram equalization for scalable brightness preservation. IEEE Trans Consum Electron 49(4):1301–1309. https://doi.org/10.1109/TCE.2003.1261233

Chen S-D, Ramli AR (2003) Minimum mean brightness error bi-histogram equalization in contrast enhancement. IEEE Trans Consum Electron 49(4):1310–1319. https://doi.org/10.1109/TCE.2003.1261234

Cho SW, Baek NR, Koo JH, Arsalan M, Park KR (2020) Semantic segmentation with low light images by modified CycleGAN-based image enhancement. IEEE Access 8:93561–93585. https://doi.org/10.1109/ACCESS.2020.2994969

Cho SW, Baek NR, Koo JH, Park KR (2021) Modified perceptual cycle generative adversarial network-based image enhancement for improving accuracy of low light image segmentation. IEEE Access 9:6296–6324. https://doi.org/10.1109/ACCESS.2020.3048366

Dai D, Gool LV (2018) Dark model adaptation: semantic image segmentation from daytime to night time. In: Proceedings 21st Interrnatinal Conference on Intelligent Transport System (ITSC), Maui, HI, pp 3819–3824

Ganesan SD, Rabbani M (2019) Contrast enhancement using completely overlapped uniformly decrementing sub-block histogram equalization for less controlled illumination variation. Int Arab J Inform Technol 16(3):389–396

Garg A, Pan X-W, Dung L-R (2022) LiCENt: low-light image enhancement using the light channel of HSL. IEEE Access 10:33547–33560. https://doi.org/10.1109/ACCESS.2022.3161527

Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative Adversarial Networks. http://arxiv.org/abs/1406.2661

Guo X, Li Y, Ling H (2017) LIME: low-light image enhancement via illumination map estimation. IEEE Trans Image Process 26(2):982–993. https://doi.org/10.1109/TIP.2016.2639450

Guo Y, Ke X, Ma J, Zhang J (2019) A pipeline neural network for low-light image enhancement. IEEE Access 7:13737–13744. https://doi.org/10.1109/ACCESS.2019.2891957

Guo Y, Lu Y, Liu RW, Yang M, Chui KT (2020) Low-light image enhancement with regularized illumination optimization and deep noise suppression. IEEE Access 8:145297–145315. https://doi.org/10.1109/ACCESS.2020.3015217

Huang W, Zhu Y, Huang R (2020) Low light image enhancement network with attention mechanism and retinex model. IEEE Access 8:74306–74314. https://doi.org/10.1109/ACCESS.2020.2988767

Jiang Y et al (2021) EnlightenGAN: deep light enhancement without paired supervision. IEEE Trans Image Process 30:2340–2349. https://doi.org/10.1109/TIP.2021.3051462

Lamba M, Rachavarapu KK, Mitra K (2021) Harnessing multi-view perspective of light fields for low-light imaging. IEEE Trans Image Process 30:1501–1513. https://doi.org/10.1109/TIP.2020.3045617

Land EH (1977) The retinex theory of color vision. Sci Amer 237(6):108128

Li Y, Li J, Li Y, Kim H, Serikawa S (2019) Low-light underwater image enhancement for deep-sea tripod. IEEE Access 7:44080–44086. https://doi.org/10.1109/ACCESS.2019.2897691

Li C, Tang S, Yan J, Zhou T (2020) Low-light image enhancement via pair of complementary gamma functions by fusion. IEEE Access 8:169887–169896. https://doi.org/10.1109/ACCESS.2020.3023485

Liang Z, Cai J, Cao Z, Zhang L (2021) CAMERANET: a two-stage framework for effective camera ISP learning. IEEE Trans Image Process 30:2248–2262. https://doi.org/10.1109/TIP.2021.3051486

Lim S, Kim W (2021) DSLR: deep stacked laplacian restorer for low-light image enhancement. IEEE Trans Multimedia 23:4272–4284. https://doi.org/10.1109/TMM.2020.3039361

Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Boston, MA, pp 3431–3440

Lu K, Zhang L (2021) TBEFN: a two-branch exposure-fusion network for low-light image enhancement. IEEE Trans Multimedia 23:4093–4105. https://doi.org/10.1109/TMM.2020.3037526

Lu Y, Kim D-W, Jung S-W (2020) DeepSelfie: single-shot low-light enhancement for selfies. IEEE Access 8:121424–121436. https://doi.org/10.1109/ACCESS.2020.3006525

Ma F, Chai J, Wang H (2019) Two-dimensional compact variational mode decomposition-based low-light image enhancement. IEEE Access 7:136299–136309. https://doi.org/10.1109/ACCESS.2019.2940531

Ma T et al (2021) Retinex GAN: unsupervised low-light enhancement with two-layer convolutional decomposition networks. IEEE Access 9:56539–56550. https://doi.org/10.1109/ACCESS.2021.3072331

Megha P, Swarna M, Sowmya V, Soman KP (2016) Low contrast satellite image restoration based on adaptive histogram equalization and discrete wavelet transform. In: International Conference on Communication and Signal Processing (ICCSP), pp 0402–0406. https://doi.org/10.1109/ICCSP.2016.7754166

Narendra PM, Fitch RC (1981) Real-time adaptive contrast enhancement. IEEE Trans Pattern Anal Mach Intell PAMI-3:655–661

Ooi CH, Kong NSP, Ibrahim H (2009) Bi-histogram equalization with a plateau limit for digital image enhancement. IEEE Trans Consum Electron 55(4):2072–2080. https://doi.org/10.1109/TCE.2009.5373771

Parihar AS, Verma OP (2016) Contrast enhancement using entropy-based dynamic sub-histogram equalisation. IET Image Process 10(11):799–808

Park S, Yu S, Kim M, Park K, Paik J (2018) Dual autoencoder network for retinex-based low-light image enhancement. IEEE Access 6:22084–22093. https://doi.org/10.1109/ACCESS.2018.2812809

Ravirathinam P, Goel D, Ranjani JJ (2021) C-LIENet: a multi-context low-light image enhancement network. IEEE Access 9:31053–31064. https://doi.org/10.1109/ACCESS.2021.3059498

Ren Y, Ying Z, Li TH, Li G (2019) LECARM: low-light image enhancement using the camera response model. IEEE Trans Circuits Syst Video Technol 29(4):968–981. https://doi.org/10.1109/TCSVT.2018.2828141

Sheet D, Garud H, Suveer A, Mahadevappa M, Chatterjee J (2010) Brightness preserving dynamic fuzzy histogram equalization. IEEE Trans Consum Electron 56(4):2475–2480. https://doi.org/10.1109/TCE.2010.5681130

Sim KS, Tso CP, Tan YY (2007) Recursive sub-image histogram equalization applied to gray scale images. Pattern Recogn Lett 28(10):1209–1221. https://doi.org/10.1016/j.patrec.2007.02.003

Singh N, Bhandari AK (2021) Principal component analysis-based low-light image enhancement using reflection model. IEEE Trans Instrum Measurem 70(1–10):5012710. https://doi.org/10.1109/TIM.2021.3096266

Singh K, Kapoor R (2014) Image enhancement using exposure based sub image histogram equalization. Pattern Recogn Lett 36:10–14. https://doi.org/10.1016/j.patrec.2013.08.024

Singh K, Kapoor R (2014) Image enhancement via median-mean based sub-image-clipped histogram equalization. Optik 125(17):4646–4651. https://doi.org/10.1016/j.ijleo.2014.04.093

Wang Y, Ren J (2018) Low-light forest frame image segmentation based on color features. J Phys Conf Ser 1069(1):012165

Wang Y, Chen Q, Zhang B (1999) Image enhancement based on equal area dualistic sub image histogram equalization method. IEEE Trans Consum Electron 45(1):68–75. https://doi.org/10.1109/30.75441

Wang L, Liu Z, Siu W, Lun DPK (2020a) Lightening network for low-light image enhancement. IEEE Trans Image Process 29:7984–7996. https://doi.org/10.1109/TIP.2020.3008396

Wang M, Tian Z, Gui W, Zhang X, Wang W (2020b) Low-Light image enhancement based on nonsubsampled shearlet transform. IEEE Access 8:63162–63174. https://doi.org/10.1109/ACCESS.2020.2983457

Wang R, Jiang B, Yang C, Li Q, Zhang B (2022) MAGAN: unsupervised low-light image enhancement guided by mixed-attention. Big Data Mining Anal 5(2):110–119. https://doi.org/10.26599/BDMA.2021.9020020

Wang W, Wei C, Yang W, Liu J (2018) GLADNet: low-light enhancement network with global awareness. In: Proceedings—13th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2018, pp 751–755. https://doi.org/10.1109/FG.2018.00118

Yadav G, Maheshwari S, Agarwal A (2014) Contrast limited adaptive histogram equalization-based enhancement for real time video system. In: International conference on advances in computing, communications and informatics (ICACCI), pp 2392–2397. https://doi.org/10.1109/ICACCI.2014.6968381

Yang W, Wang W, Huang H, Wang S, Liu J (2021) Sparse gradient regularized deep retinex network for robust low-light image enhancement. IEEE Trans Image Process 30:2072–2086. https://doi.org/10.1109/TIP.2021.3050850

Zhang X, Wang X (2021) MARN: multi-scale attention Retinex network for low-light image enhancement. IEEE Access 9:50939–50948. https://doi.org/10.1109/ACCESS.2021.3068534

Zhu H, Zhao Y, Wang R, Wang R, Chen W, Gao X (2021) LLISP: low-light image signal processing net via two-stage network. IEEE Access 9:16736–16745. https://doi.org/10.1109/ACCESS.2021.3053607

Skin Cancer Detection with Metadata Using Deep Learning Strategies

Nagaraju Devarakonda, Manda Venkata Ramana Murthy, Racham Reddy Chinmay Reddy, and Pabbathi B. L. Shree Harsha

1 Introduction

According to WHO percentage of people who got effected by skin cancer is increased to 33% in 2021. Back in 2021, Around 1.2 million people died due to skin cancer all around world (American Cancer Society 2018). Where 7.22 lakhs of people are men and 4.76 lakhs of people are women. Skin cancer is affecting a lot of people from Australia, New Zealand, and Denmark than compared with other countries. Where as in India, most skin cancer cases are affecting from North India. When human body is exposed to UV rays for longer time then Basal cells in our humans get affected which is the main cause for skin cancer and these cells damage the unaffected and active ones which increases the damage percentage (Alfi et al. 2022). Generally, Basal cells produce new skin cells when old skin cells are damaged. But when the skin is affected by skin cancer the Basal cells will not work properly and will not generate new skin cells. Sooner or later the cancer will spread throughout the body as there are no new cells produced, which is very dangerous and may lead to death.

Dermoscopy is an assistant diagnosis method which is done by taking some pictures with the help of computer systems (Demir et al. 2019). We used deep learning techniques instead of machine learning because in ML it takes just a small amount of dataset but when it comes for classification it requires more data. And even it takes more time for training the train set to perform as good as dl but still it cannot give us promising and accurate results. But here comes deep learning which will overcome all the problems in ml by using neural network (Bechelli and Delhommelle 2022). These neural networks are similar to the neurons in our brain. In dl we can extract the features from the given input and dl can even take large amount data for train and we can expect efficient accuracy and results. Now a days, deep learning became common approach for image detection. It is widely used for classification

N. Devarakonda (✉) · M. V. R. Murthy · R. R. C. Reddy · P. B. L. Shree Harsha
VIT AP University, Amaravati, Andhra Pradesh, India
e-mail: dnagaraj_dnr@yahoo.co.in

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_16

and prediction type of problem. Moreover, a greater accuracy rate of 97.78% was obtained in Mohamed et al. (2019) when the number of layers in CNN was increased to 14 layers on dermoscopic images from the ISIC dataset.

In previous research papers, they used CNN for processing the image and later to get better improvements in accuracy they used 5 models for better image classification and for improvement of accuracy in the models of one of the model is Efficient Net B4 which takes less parameters and provides less accuracy we also have other types of models like Efficient Net B5 which takes less parameters and give us better accuracy than Efficient Net B4 so in this paper we used Efficient Net B5 as one of the model in the 5 models which we used.

In our paper, we used CNN instead of ANN because ANN works as human brain by taking weights. If it anything gets wrong it goes back and gets updates its weights and again it will work as human brain. The updating of weights is based on cost function. Where as in CNN, it just uses layers for filtration and analyses the image input. The layers that are used in CNN are Convolution, ReLU, Pooling, Flattening. In our paper used different types of CNN models like ResNet, GoogleNet, VGGNet, EffectiveNet B5 but found better accuracy for EffectiveNet B5.

1.1 Use of Metadata

While there are many CNN models to detect skin cancer but the problem is the prediction of new image with skin cancer. In these models, we used metadata processing. When it comes to metadata, it has patient records like age, gender, location of infection, diagnosis of cancer, type of skin cancer to patient, etc. We used some standard datasets like HAM10000 and ISIC 2019 (Dildar et al. 2022).

In Fig. 1 it describes the Meta Block which is our model which contains image data which is preprocessed, reshaped into a NumPy array and is stored in variable '*x*' from the picture we can observe that it also contains metadata which contains records of patients along with target variable (skin cancer classifier) and we have taken target variable as '*y*' which we later categorized into different categories based on skin cancer types in their respective datasets. And used *x* and *y* to train and test the CNN model (Pacheco and Krohling 2019). The overall process will happen inside Meta Block (Wen et al. 2022).

- Initially imported the dataset from Kaggle and downloaded.
- And used label encoder to label the data in the target variable which is in metadata (CSV file).
- We resized the image data and converted them into NumPy arrays.
- Next we reshaped the NumPy array by dividing them with 255.
- Then we categorized the target variable into 2D array based on number of classes.
- Later we aggregated features of the image along with target variable.
- Then we applied CNN model, GoogleNet, VGGNet, RESNet, Efficient Net B5 to the aggregated features.

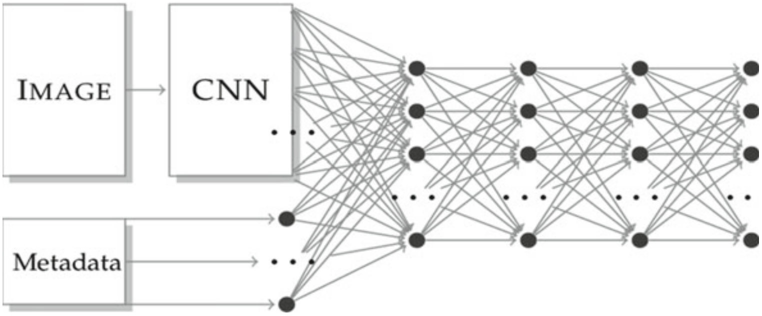


Fig. 1 Working of Meta Block in which we can see that it is the addition of CNN with metadata

- Finally compared the accuracies for all architectures.

2 Related Works

Basically in this paper we used a concept called Meta Block which is previously used by some of the researchers for better image classification they used this to enhance the features that are extracted from the images. The metadata is basically used as a guide for the features if the images so by using metadata the features extracted are correctly used so that the classification of images is done more accurately. The combination of metadata with image data prevents the overfitting occurring in CNN due to the usage of dermoscopy images in skin cancer classification. They used One Hot Encoder for binary variables and label encoder for categorical variables. These will help for labeling the string data into numerical data by assigning similar string into common numerical value. This will help for target variable. We take these label target variables convert into a 2D array. Based on number of classes and by using to categorical function. This is the metadata we use to process the image data which is used for better classification.

A large form of data that is available online is the major reason for developing a mechanism to create effective and much scalable algorithms in metadata-based image mining. To maximize the benefits of clustering, current approaches focus on data processing to generate side information. Advantages of clustering for generating side information. Researches proposed approaches to cluster image data using side information. There are ways to run the mining process in ways that maximize the benefit of one page of information. More over metadata gives more information by doing data augmentation and using reshaping. This data is does the model much easier to improve the prediction than normal way of doing it. This is achieved with an algorithm that combines traditional partitioning algorithms with probabilistic models for effective clustering. Douglas indicates that document clustering is underutilized for searching the metadata base. His rebuttals fall primarily into his two categories. First, it is too slow for large-scale clustering, and second, classification and clustering do not improve retrieval. Clustering is used to improve predictive search techniques. However, data processing as an information access tool itself circumvents these objections and offers an efficient new access paradigm.

3 Methodology

3.1 Convolutional Neural Networks (CNN)

CNN is a type of deep learning technique used for image classification and for pixel classification data. For identifying and recognition CNN is the best architecture. The another type of neural network that can uncover key information in both time series and image data. CNN is also used to identify pattern recognition in the image (Fuadah et al. 2020).

As we discussed earlier, from convolution to fully connected layer the complexity of CNN increases. Convolution is the core part of CNN architecture. In this we do a dot product of image pixels and filter which gives an output know as feature map or convolved map (Manne et al. 2020). Ultimately image is converted into numerical values in this stage which helps in extracting image patterns. After this ReLU is applied to it. ReLU helps in converting the negative values into zero and keeping the positive values as it is. In pooling layer, we reduce the parameters in the input and some information is lost. But leaving the disadvantage, this helps to reduce complexity of layer and improves efficiency of CNN (Saeed and Zeebaree 2021).

Fully connected network does the image classification based on features that are extracted from previous layers. In this our data is flattened into a single array of line and these are linked to each other. This linked data extracts each feature from one layer and finally give us the output, i.e., image classification.

In previous research papers, they used CNN for processing the image and later to get better improvements in accuracy they used 5 models for better image classification and for improvement of accuracy in the models of one of the model is Efficient Net B4 which takes less parameters and provides less accuracy we also have other types of models like Efficient Net B5 which takes less parameters and give us better accuracy than Efficient Net B4 so in this paper we used Efficient Net B5 as one of the model in the 5 models which we used.

As we know that in image classification features will be extracted from input image that is like patterns, based on the image patterns the model will be trained and tested accordingly. We use other models to extract shape, color, and texture of the input image. The main problem we face in this is to combine the features extracted from the image and the metadata. So, we use concatenation to link the images and metadata in single file and we add extracted images data features in the metadata file according to the respective images.

Usually, image data has a greater number of features than the features of metadata as image data is high dimensional which means it has high resolution. But the metadata contains only textual format of patient's records which are less complex than image data. So normal concatenation of data might not work in all the cases. So better aggregation data gives us more accurate results.

3.2 Meta Block

Now let's see the mathematical representation of Meta Block

$$T_{\text{metadata}} = \phi(\text{Patients records}) \tag{1}$$

$$T_{\text{images}} = \phi(\text{Dermoscopy images}) \tag{2}$$

When we combine both metadata and dermoscopy images we will get Meta Block

$$T_{\text{metablock}} = \phi(\text{Patients record} + \text{Dermoscopy images}) \tag{3}$$

Meta block will enhance the feature extraction from the dermoscopy images.

3.3 Google Net

Google Net was proposed by Google in 2014 to dig deeper into convolutional networks (Yilmaz and Trocan 2021). This architecture is very different from other architectures. It is a deep convolutional neural network with 22 layers (27 layers including pooling layers), some of these layers are total 9 Inception modules. 224 × 224 dimensions of image is taken as input layer for this architecture. This architecture consists of many things, like type, patch size, stride, output size, input size, depth, pools, params, ops (Sa'idah et al. 2022). In convolution's filters 1 × 1, 3 × 3, 5 × 5 are used in the Inception module. Whereas 1 × 1 is the filters dimensions.

3.4 ResNet

ResNet is also known as Residual Network. Residual Network architecture is also one of the techniques to increase models working efficiency. Used to skip shifts without affecting model performance. On comparing to other architectures, it has very deep network (152 layers). Also, this is subject to both vanishing and exploding gradients, impacting model performance when transitioning to the next epoch. This leaves a learning framework to facilitate much deeper training networks (Budhiman et al. 2019). With ResNet, you can train hundreds or even thousands of layers and still get convincing performance.

Here we see that ResNet consists of convolution and pooling steps followed by four layers with similar behavior (Jinnai et al. 2020). Each layer follows the same pattern. They perform 3 × 3 convolutions with fixed feature map dimensions (F) [64, 128, 256, 512], bypassing the input every two convolutions. Also, the width (W) and height (H) dimensions are constant throughout the shift.

3.5 VGG 16

VGG stands for Visual Geometry Group and is a standard multilayer deep convolutional neural network (CNN) architecture. This VGG network consists of 140 million parameters (Manasa and Murthy 2021). The 16 in VGG16 refers to 16 layers with weights. VGG16 has 13 convolution layers, 5 max-pooling layers, 3 dense layers, and 21 layers in total, but only 16 weight layers. The input to the network is an image of dimension (224, 224, 3). The first two layers have 64 channels with a filter size of 3*3 and the same padding. Then, after a maximum pool layer of stride (2, 2), there are two layers of convolution layers with filter size 128 and filter size (3, 3) (Tabrizchi et al. 2020).

3.6 Effective Net B-5

Effective Net is a powerful convolution neural network to increase model's performance. This model does Scaling, Depth balancing, and Resolution balancing which leads to efficient output (Saeed and Zeebaree 2021). It has 3*3 and 5*5 convolutional layers and refers to the various convolutional filters used in the Inception module. It optimizes both accuracy and efficiency, measured on a floating-point operations per second (FLOPS) basis (Manne et al. 2020). This developed architecture uses mobile reverse bottleneck convolution (MBConv).

4 Dataset

HAM10000

The HAM10000 dataset, a large collection of multi-sources dermoscopic images of common pigmented skin lesions (Wen et al. 2022). This dataset contains 10,014 dermoscopic images of 7 different classes namely melanoma, melanocytic nevus, basal cell carcinoma, actinic keratosis, benign keratosis, dermatofibroma, and

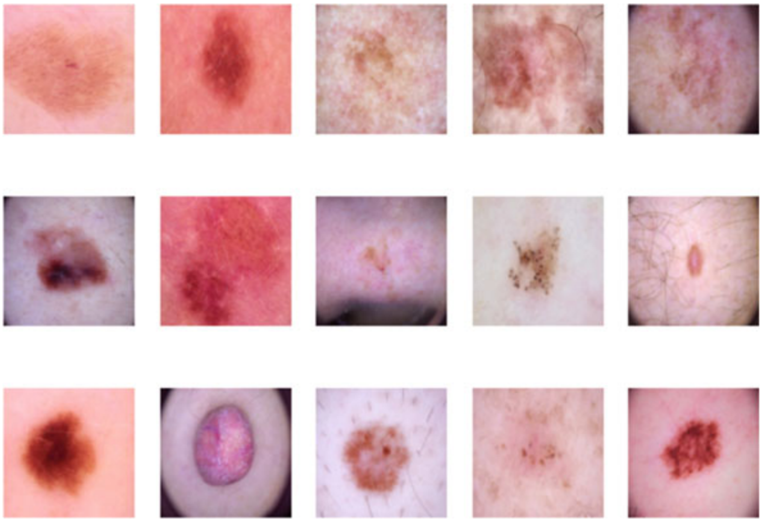


Fig. 2 Samples of skin cancer in HAM10000 dataset

vascular lesion. The target values and metadata are present in train .csv file. Size of images is 450*600 pixels (Fig. 2).

ISIC 2019

The ISIC 2019 dataset, a large collection of multi-sources dermoscopic images of common pigmented skin lesions (Wen et al. 2022). The dataset for ISIC 2019 contains 25,331 images available for the classification of dermoscopic images among nine different diagnostic categories: Melanoma, Melanocytic nevus, Basal cell carcinoma. The target values and metadata are present in train .csv file (Pacheco and Krohling 2019). The size of each image in the dataset is 512*512 pixels (Fig. 3).

5 Results

Let us check the results that found from our model for HAM10000 dataset with and without considering metadata as per Tables 1 and 2.

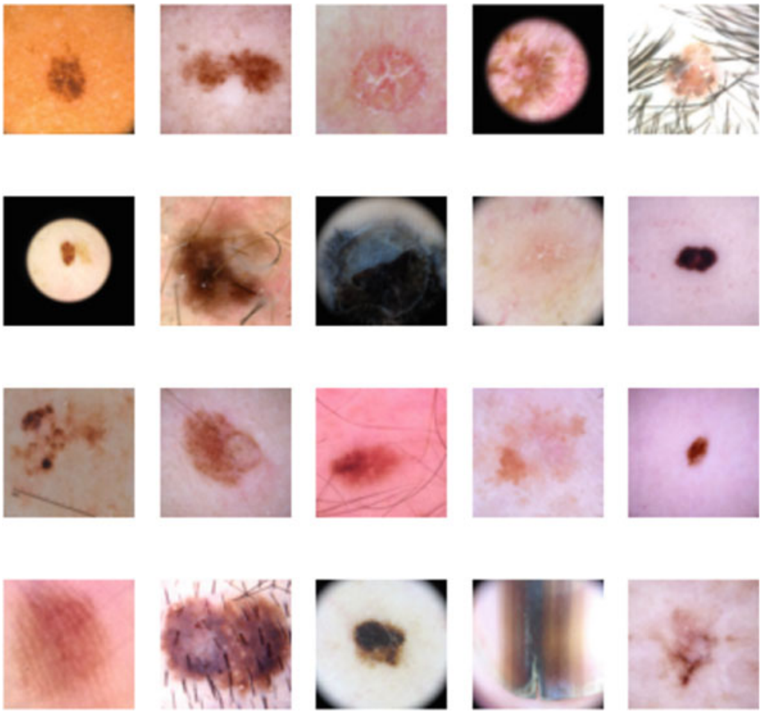


Fig. 3 Samples of skin cancer in ISIC 2019 dataset

Table 1 Accuracy and loss values for HAM10000 dataset using different architectures

Models	Training accuracy	Testing accuracy	Training loss	Testing loss
Basic CNN	98.56	73.48	0.0124	0.4282
Google NET	98.86	78.07	0.0102	0.3468
VGG Net 16	96.69	71.81	0.0262	0.3291
RESNET 50	99.30	75.04	0.0064	0.4138
Efficient Net B5	99.40	80.83	0.0065	0.2647

Table 2 Accuracy and loss values for ISIC2019 dataset using different architectures

Models	Training accuracy	Testing accuracy	Training loss	Testing loss
Basic CNN	98.89	61.96	0.0347	4.1967
Google NET	97.48	70.20	0.0183	0.3288
VGG Net 16	99.63	71.18	0.0030	0.4240
RESNET 50	99.30	67.47	0.0058	0.6234
Efficient Net B5	99.14	75.28	0.0069	0.7224

5.1 Graphical Representation of Accuracy and Loss for HAM10000 Dataset

Basic CNN

See Fig. 4.

Google Net

See Fig. 5.

VGG16

See Fig. 6.

ResNet 50

See Fig. 7.

Effective Net B5

See Fig. 8.

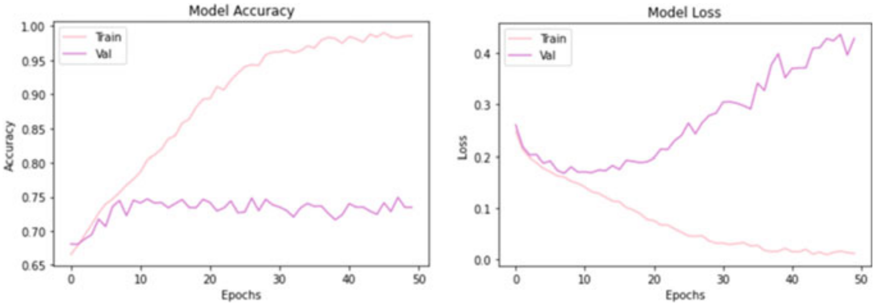


Fig. 4 Accuracy and loss using Basic CNN model for HAM10000

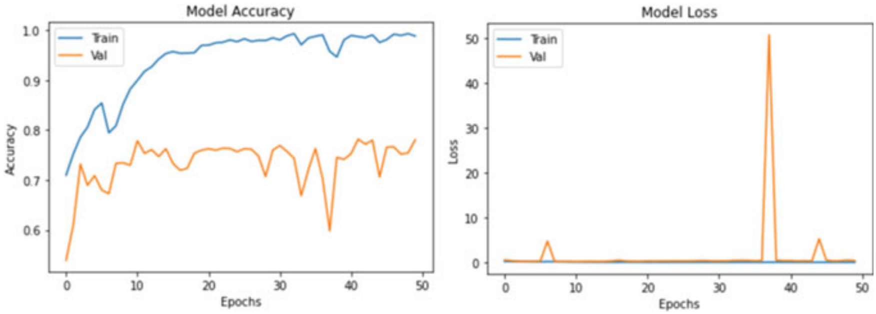


Fig. 5 Accuracy and loss using Google Net model for HAM10000

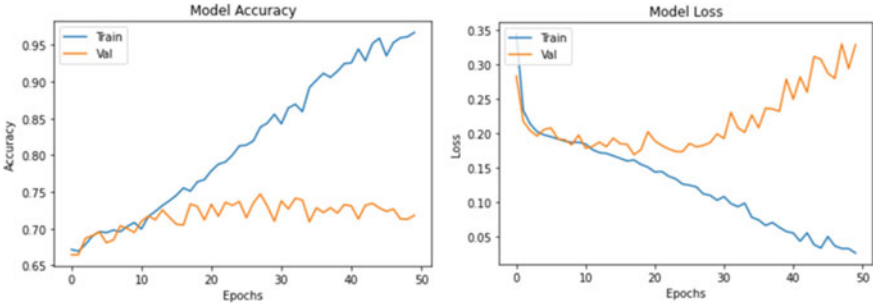


Fig. 6 Accuracy and loss using VGG16 model for HAM10000

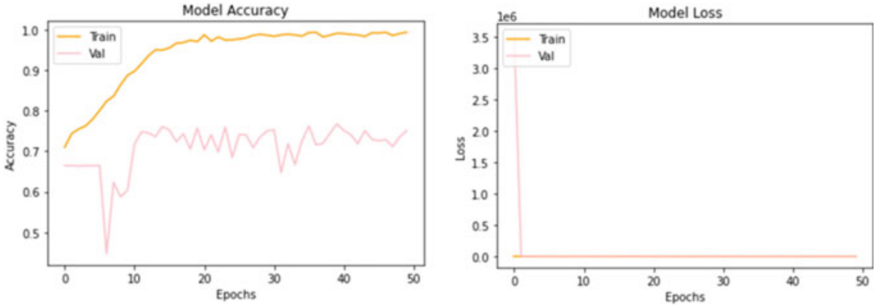


Fig. 7 Accuracy and loss using ResNet 50 model for HAM10000

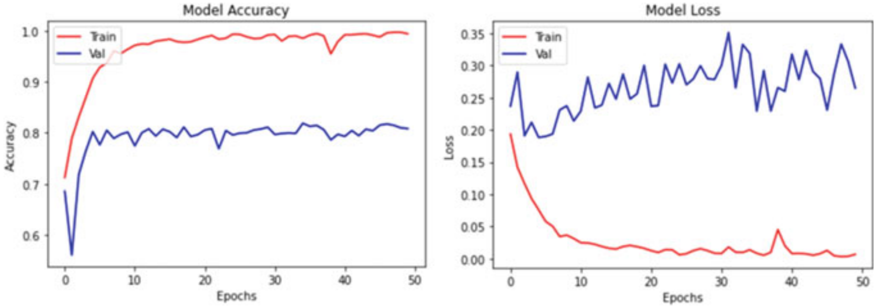


Fig. 8 Accuracy and loss using Efficient Net B5 model for HAM10000

5.2 Comparing Accuracies

See Figs. 9 and 10.

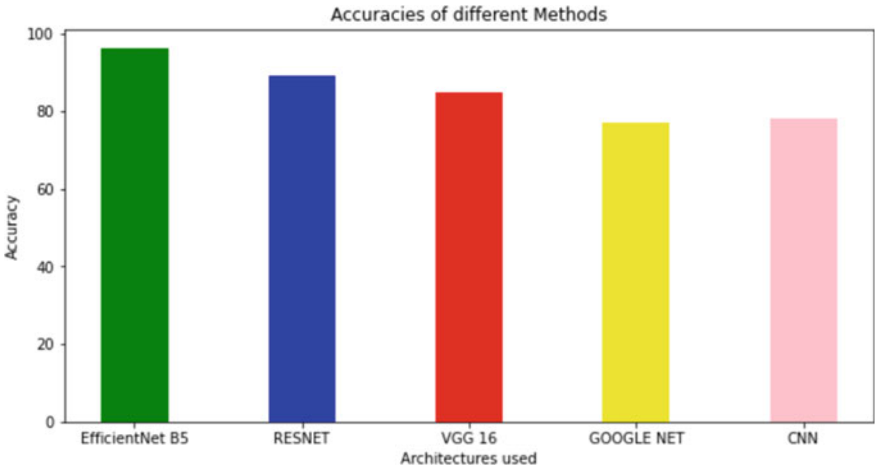


Fig. 9 Train accuracy for all architectures in HAM10000

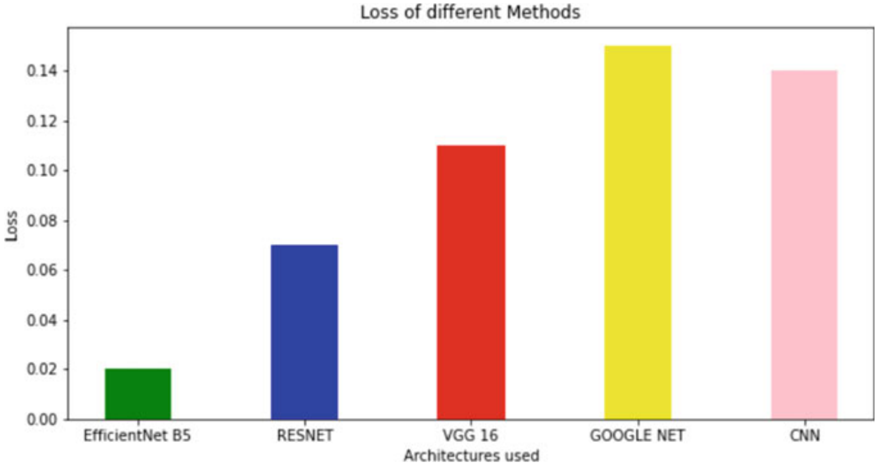


Fig. 10 Test accuracy for all architectures in HAM10000

5.3 Graphical Representation of Accuracy and Loss for ISIC 2019 Dataset

Basic CNN

See Fig. 11.

Google Net

See Fig. 12.

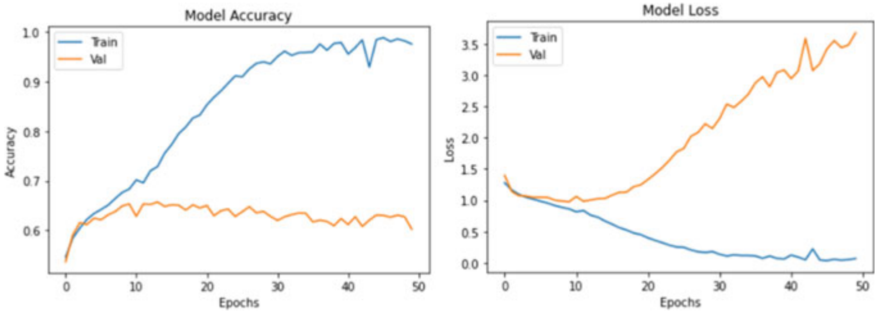


Fig. 11 Accuracy and loss using Basic CNN model for ISIC 2019

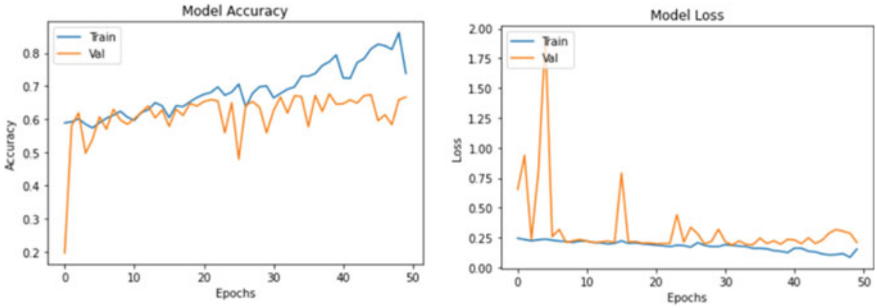


Fig. 12 Accuracy and loss using Google Net model for ISIC 2019

VGG16

See Fig. 13.

ResNet 50

See Fig. 14.

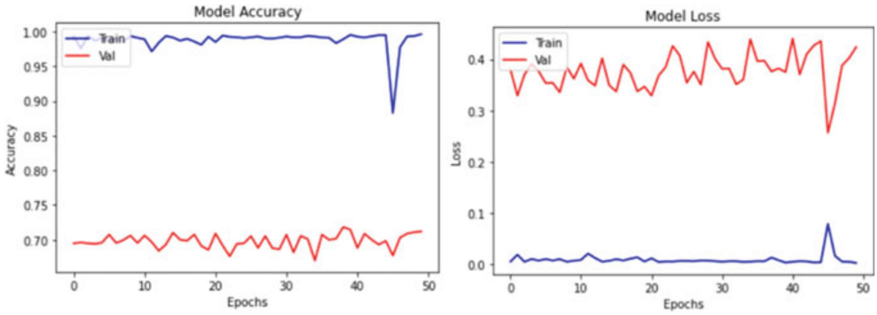


Fig. 13 Accuracy and loss using VGG16 model for ISIC 2019

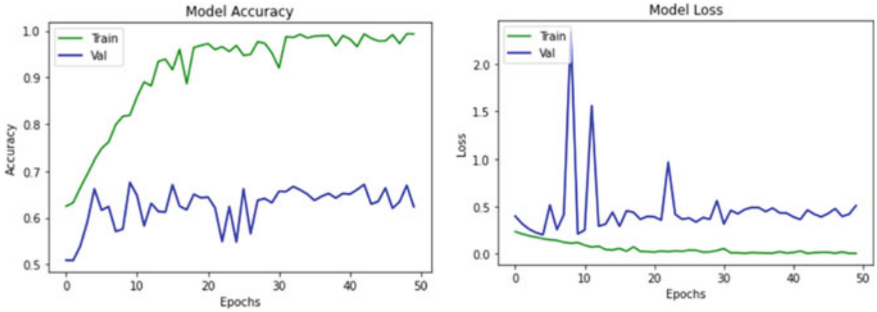


Fig. 14 Accuracy and loss using ResNet model for ISIC 2019

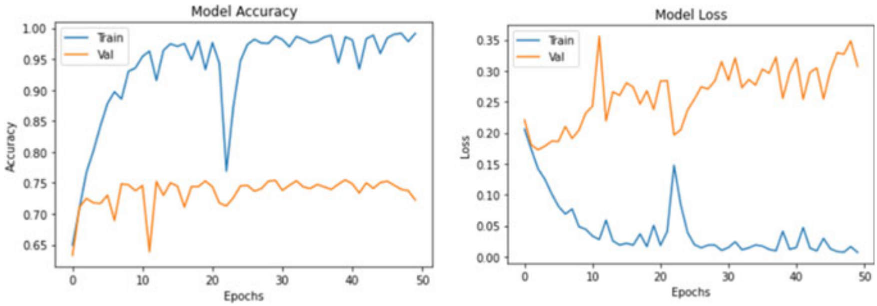


Fig. 15 Accuracy and loss using Efficient Net model for ISIC 2019

Effective Net B5

See Fig. 15.

5.4 *Comparing Accuracies*

See Figs. 16 and 17.

6 Result Analysis and Future Scope

As we can see in the previous section the results clearly state that for both HAM10000 and ISIC 2019 datasets, Effective Net gave more accuracy compared to remaining architectures. For HAM10000 dataset by using Basic CNN model we have good training accuracy compared to existing, i.e., around 90% and where as we got around 98.5%. We can see the clear view of accuracy and loss for all the models

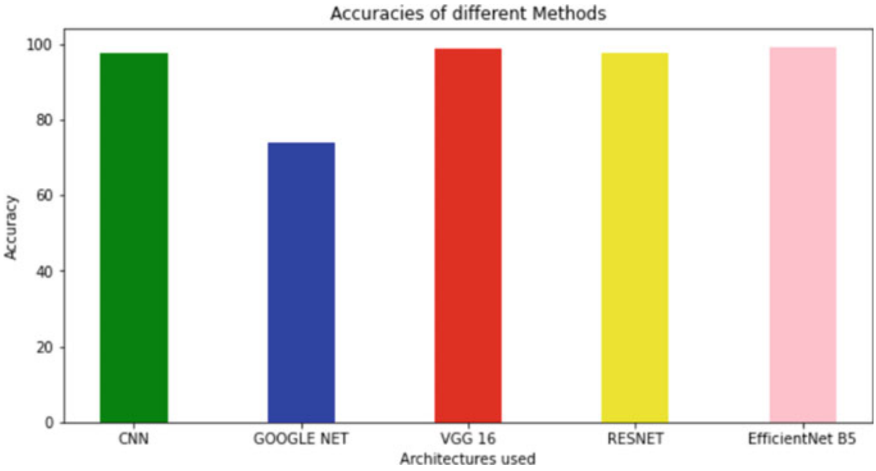


Fig. 16 Train accuracy for all architectures in ISIC 2019

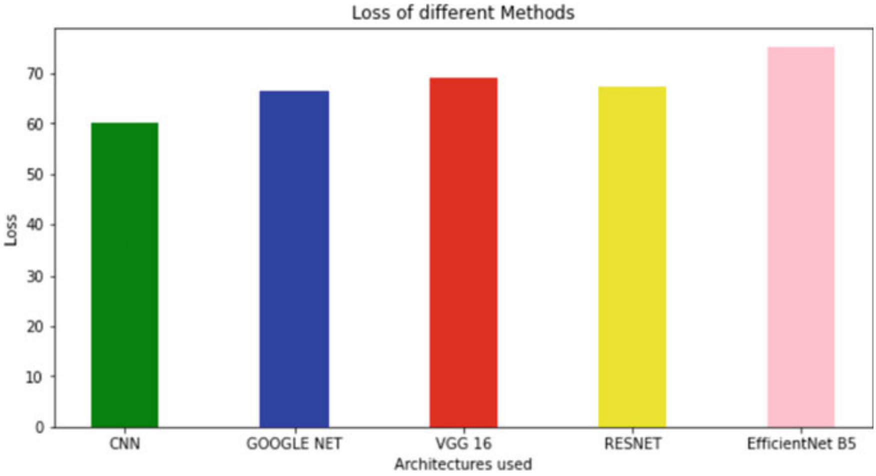


Fig. 17 Test accuracy for all architectures in ISIC 2019

for HAM10000 with in the graphical representation which shows its values for each epoch. In this Basic CNN model, according to graph test accuracy is very low compared to train and loss is high for test compared to train (Fig. 4). For the same dataset we used Google Net as secondary model where with combining metadata we got same training accuracy as Basic CNN but compared with testing we got much higher accuracy which increases the model's prediction efficiency. The testing accuracy increased to 78% in this model. According to Fig. 5, training and testing accuracies are comparatively better than previous model whereas training loss and testing loss is also low.

The next model here we used is VGG16 which got accuracy and loss less than previous models (Fig. 6). So, we can say clearly that VGG16 is not much suitable for this model. Coming to ResNet 50, training accuracy and loss are very good compared to previous models. Looking at the graph we can say that training loss is very low than any other model till now (Fig. 7). The final model we used in this is Effective Net B5. We got highest accuracy and highest testing than any other and loss is also very low (Fig. 8). Now looking at (Figs. 9 and 10), we clearly state that Effective Net B5 is having highest accuracy and least loss than any other models. This tells us that Effective Net B5 is best suited for this model and we got to know that Basic CNN has least prediction accuracy compared to remaining which is not best suited for this.

However we got a clarity on HAM10000 dataset, similarly now we will see about ISIC 2019 dataset for skin cancer. Well looking at Basic CNN model for this dataset, the accuracies for training and testing are less when compared to HAM10000 dataset (Fig. 11). But for this model it is average rated. Looking at the next model, i.e., Google Net, the training and testing accuracies are very close which is a good sign for this and decreases the problem for overfitting (Fig. 12). The loss for this model is also low compared to Basic CNN. The next model we used here is VGG16, on looking at (Fig. 13) we can clearly say that there is a huge difference between training and testing accuracies, on the other side the loss also represents the same thing. Looking around ResNet 50 we have both training and testing accuracies at a good level and loss is also less (Fig. 14). The final model of this dataset we used is Effective Net B5, here the accuracies for both training and testing are well obtained and train loss is also low (Fig. 15).

On observing (Figs. 16 and 17) as previous, the same got repeated here. Effective Net architecture shows best accuracy for both training and testing for both the datasets. The most suitable architecture for skin cancer prediction is effective net and least suitable for ISIC 2019 is Google Net.

This project helps doctors and many researches to find the cancer earlier and also helps in speed treatment and dialysis. This project helps people to find that they have cancer at the initial stage and can decrease the cause of infection. In future we will work upon different types of diseases including different types of cancers.

7 Conclusion

In this paper, we found a way for detection of skin cancer detection from dermoscopic images and metadata of patient. We applied this methodology in two different datasets of ISIC 2019 and HAM10000 which were RGB. The techniques that we used are CNN, VGGNet, Google Net, ResNet, and EffectiveNet. These techniques will help in finding the skin cancer in the earlier stage. Upon all the techniques we got the best accuracy using Effective Net B5. We also compared datasets with metadata and without metadata and found maximum accuracy with metadata. We also got balanced accuracy of 95 with training data. We used model for different dataset and got best accuracy and results for every dataset. We performed prediction in both datasets and found that our model predicted accurately. And also performed good.

References

Alfi IA, Rahman MdM, Shorfuzzaman M, Nazir A (2022) A non-invasive interpretable diagnosis of melanoma skin cancer using deep learning and ensemble stacking of machine learning models. Diagnostics 12:726. https://doi.org/10.3390/diagnostics12030726

American Cancer Society (2018) Cancer facts and figures. Available: https://www.cancer.org/content/dam/cancer-org/research/cancer-factsand-statistics/annual-cancer-facts-and-figures/2018/cancer-facts-andfigures-2018.pdf. Accessed 15 Aug 2018

Bechelli S, Delhommelle J (2022) Machine learning and deep learning algorithms for skin cancer classification from dermoscopic images. Bioengineering 9:97. https://doi.org/10.3390/bioengineering9030097

Budhiman A, Suyanto S, Arifianto DA (2019) Melanoma cancer classification using ResNet with data augmentation. In: International seminar on research of information technology and intelligent systems (ISRITI), vol 4(123).

Demir A, Yilmaz F, Kose O (2019) Early detection of skin cancer using deep learning architectures. In: 2nd international congress on engineering and architecture, pp 1364–1329

Dildar M, Akram S, Irfan M, Khan HU, Ramzan M, Mahmood AR, Alsaiari SA, Saeed AHM, Alraddadi MO, Mahnashi MH (2022) Skin cancer detection: a review using deep learning techniques. Diagnostics 12:726

Fuadah YN, Pratiwi NKC, Pramudito MA, Ibrahim N (2020) Convolutional neural network (CNN) for automatic skin cancer classification system. IOP Conf Ser Mater Sci Eng 982:012005. https://doi.org/10.1088/1757-899X/982/1/012005

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778

Hosny KM, Kassem MA, Foaud MM (2017) Skin cancer classification using deep learning and transfer learning. IBM J Res Dev 61:5:1–5:15

Jinnai S, Yamazaki N, Hirano Y, Sugawara Y, Ohe Y, Hamamoto R (2020) The development of a skin cancer classification system for pigmented skin lesions using deep learning. Biomolecules 10:1123. https://doi.org/10.3390/biom10081123

Jones T, Matin RN, van der Schaar M, Bhayankaram KP, Ranmuthu CKI, Islam MS, Behiyat D, Boscott R, Calanzani N, Emery J, Williams HC, Walte FM (2022) Artificial intelligence and machine learning algorithms for early detection of skin cancer in community and primary care settings. Lancet Digit Health 4. www.thelancet.com/digital-health

Manasa K, Murthy GV (2021) Skin cancer detection using VGG-16. Eur J Molecular Clin Med 08(01). ISSN 2515-8260

Manne R, Kantheti S, Kantheti S (2020) Classification of skin cancer using deep learning, convolutional neural networks. Int J Mod Trends Sci Technol 6(11):101–108

Mohamed A, Mohamed WA, Zekry AH (2019) Deep learning can improve early skin cancer detection. Int J Electron Telecommun 65

Montaha S, Azam S, Rafid AKMRH, Islam S, Ghosh P, Jonkman M (2022) A shallow deep learning approach to classify skin cancer using down-scaling method to minimize time and space complexity. https://doi.org/10.1371/journal.pone.0269826

Pacheco AGC, Krohling RA (2019) An attention-based mechanism to combine images and metadata in deep learning models applied to skin cancer classification. IEEE J Biomed Health Inform 25(9)

Saeed JN, Zeebaree SRM (2021) Skin lesion classification based on deep convolutional neural networks architectures. J Appl Sci Technol Trends 02(01):41–51. ISSN 2708-0757

Sa'idah S, Suparta IPYN, Suhartono E (2022) Modification of convolutional neural network GoogLeNet architecture with dull razor filtering for classifying skin cancer. Jurnal Nasional Teknik Elektro dan TeknologiInformasi 11(2)

Tabrizchi H, Parvizpour S, Razmara J (2020) An improved VGG model for skin cancer detection. Neural Process Lett (online). https://doi.org/10.1007/s11063-022-10927-1

Wen D, Khan SM, Xu AJ, Ibrahim H, Smith L, Caballero J, Zepeda L, de Blas Perez C, Denniston AK, Liu X, Matin RN (2022) Characteristics of publicly available skin cancer image datasets. Lancet Digit Health 4:e64–74

Yilmaz E, Trocan M (2021) A modified version of GoogLeNet for melanoma diagnosis. J Inform Telecommun 5(3):395–405. https://doi.org/10.1080/24751839.2021.1893495.

Detect and Alleviate DDoS Attacks in Cloud Environment

Bandi Kulwanth, V. Srinivasarengan, Peddinte Anish, and K. Abirami

1 Introduction

Distributed denial of service (DDoS) is a cyberattack in which the attacker sends an enormous amount of network traffic to the targeted server to make it inaccessible. Recently, similar attacks have been used to block access to business-related websites that depend on an ongoing Internet presence. In December 2022, VTB Bank, the second-largest bank in Russia, suffered the largest cyberattack in Soviet history. The attack became to known as DDoS. The targeted nature and sheer volume of attacks on targeted hosts set these attacks apart from conventional DDoS strikes. Even veteran Internet businesses like Microsoft and the SCO group have experienced attacks, and previously well-known companies like eBay have had their services interrupted. DoS attacks may be used by businesses to eliminate their rivals in the marketplace. DoS assaults used for extortion have increased in recent years.

The political and financial motivations of those behind the most recent attacks, particularly the participation of international organized crime in attempts at protection extortion, have increased the threat posed by them. The issue is that we are unable to identify the perpetrators of these attacks, which is why we are unable to mount an effective defense. There are a lot of reasons for the omission of these attacks: The attack is subtle, and understanding it fully requires a more thorough explanation of

B. Kulwanth · V. Srinivasarengan · P. Anish · K. Abirami (✉)
Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Coimbatore, India
e-mail: k_abirami@cb.amrita.edu

B. Kulwanth
e-mail: cb.en.u4cse19208@cb.students.amrita.edu

V. Srinivasarengan
e-mail: cb.en.u4cse19246@cb.students.amrita.edu

P. Anish
e-mail: cb.en.u4cse19264@cb.students.amrita.edu

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_17

how the Internet is built than the mass media are willing to provide their audiences. This understanding is necessary to understand why we cannot cope with it right now and what must change before we can. Although there is no simple way to stop these attacks, there are strategies that can help to mitigate their consequences.

1.1 Major Contribution

This project aims at developing a DDoS attack detection model using machine learning models and neural network models.

- The proposed model focuses on using machine learning models and neural network models on preprocessed dataset (CIC-DDoS2019 dataset).
- We have compared the DDoS detection scores of three machine learning models and two neural network models.
- The all of the five models will be given as an input to an ensemble classifier which will merge all of them and return one ensembled machine learning model.

1.2 Motivation

DDoS attacks are a common occurrence in the realm of networks and security, but they have changed substantially over the past 25 years, from human attacks launched by protestor groups to automated attacks that take advantage of server protocol flaws to botnets built of connected devices. Multi-user systems were the target of the attacks before they spread to servers. It is interesting to monitor and identify DDoS assaults and gain more knowledge about them. By researching on attacks of DDoS and its types, we continue to improve the ability to identify and prevent the incoming attacks in the future.

2 Literature Survey

This chapter includes the various DDoS detection techniques and how to detect them using machine learning algorithms. In this section covers about detecting DDoS using Deep Neural Network, K-Nearest neighbors and Naive Bayes algorithm. DNN method detects DDoS attack through repetitive learning from a small sample. K-NN predicts the attack by assuming the similarity of new entry packet and data presented to it and puts the new packet to one of the categories that is most similar to the available categories. Naive Bayes predicts the attack based using Bayes theorem on the new entry packet. The dataset used here has nearly 23 new features in total and other data were extracted.

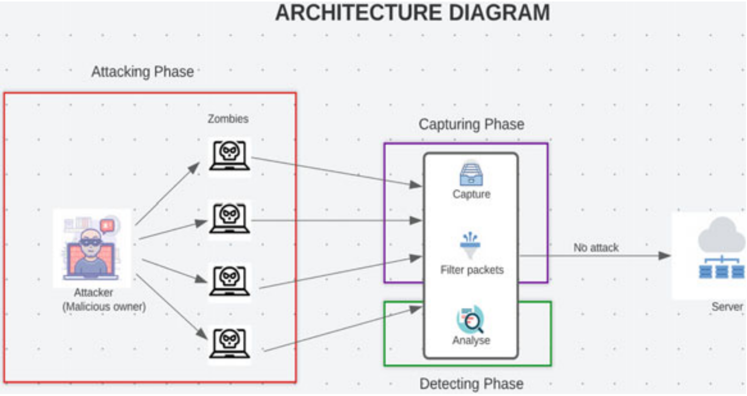
There are plethora of ways where denial-of-service attacks impact the SDN-based cloud computing. The majority of three methods consisted are UDP flood, SYN flood, and ICMP flood (Wani et al. 2019). Slowloris attack is one of the ways to attack the server and making it unstable for users. TCP link will be created which will be the first step of Slowloris attack. The new TCP link gets more frequent requests incoming to the server at customary timespan, all of the incoming requests are slowly being used up. Until some of the engaged connections are released, other servers would not be able to connect. This paper's major objective was to lower the attack detection rate and quickness (Zekri et al. 2017). The paper's main objective is to utilize SDN to predict and prevent DDoS attacks in cloud environment.

The three steps of the SDTMA architecture's function are the filtering and evaluation phase, the attack detection phase, and the mitigation phase. The phase of filtering and evaluation analyzes the incoming packets according to each category. Machine learning algorithms are applied during the attack detection phase to identify incoming attacks on the cloud environment. Mitigation phase will try to reduce speed rate (Kautish et al. 2022). The DDoS attack can be divided into three categories, including Stateful Protocol Analysis, Anomaly-based Detection, and Signature-based Detection. By observing events and seeing patterns that resemble the signatures of recognized attacks, Signature-based Detection can find intrusions. Attacks must be carried out in coordination with certain key events, which are specified in an attack signature. Furthermore, it only retrieves assaults whose signatures have already been saved in a database. The network behavior is defined in anomaly-based detection in order to detect the DDoS attack. The network behaves as it is expected to and according to specified rules. In Stateful Protocol Analysis, the ability to identify a DDoS attack depends on vendor-developed generic profiles for protocols that outline when and how to employ each protocol. Stateful Protocol Analysis offers essential resources for analyzing attacks and defending against them. This study provides a synopsis of previous studies that put out several DDoS detection methods.

Random Forest Algorithm has been shown to be superior to SVM algorithm (Pei et al. 2019; Kumar and Somani 2022). They employ scale-inside-out for the user's uninterrupted attack request and separation at the container level (Amjad et al. 2019). Naive Bayes is more effective than Random Forest (Gupta et al. 2021). An unforeseen peak in website traffic that triggers an abrupt rise in server load is known as a flash crowd or flash event. It is challenging to differentiate between acceptable flows like flash mobs and illegal ones like distributed denial-of-service attacks. The author utilizes the help of deep learning techniques to solve the flash crowd problem (Mishra et al. 2021). The paper proposes a supervised machine learning technique to identify and stop DDOS at servers in cloud environments, with a 97% accuracy rate (Lunkad and Singh 2020). DDOS attack detection has been proved using machine learning tools including SVM, K-NN, and Naive Bayes, and accuracy for the network performance has been achieved with the construction of classifiers among the suggested models (Saini et al. 2020). J48 is a machine learning decision tree classification technique which helps the data to be examined in the categorical and in the regular fashion. The author takes five distinct classes and 27 features were used in the dataset and proved that J48 is superior to Naive Bayes, Random Forest, and MLP.

3 Proposed System

Model overview: For model implementation, we have used the preprocessed datasets (CIC dataset 2019) where it consists datasets for different packets such as SYN, TCP, ICMP. Each of the dataset contains 84 parameters for each transmitting packets, out of which we selected 20 features for our analysis. We will feed the 20 parameters as input into machine learning algorithm and analyzed the detection accuracy of the each algorithm. We compared and ensembled five machine learning models into one model for detecting DDoS packet. Now in order to use our own dataset, we have to create a DDoS attack on a local server and capture all the incoming packets to the server. Captured data can be preprocessed and used as a dataset that can be fed into ensembled machine learning model which we will be doing in the next phase of our project.



3.1 *Attacking Phase*

The next step of the project is to create a local server and try to attack via one of the DDoS methods. As there are plethora of methods available, we have chosen Ufonet tool to attack a local server created by XAMPP. Ufonet is a DoS and DDoS tool which can send larger amounts of packet to targeted server and makes the server inaccessible. As an input, we have provided the local IP address of the server created by Xampp.

3.2 Capturing Phase

In this phase, sniffing of the packets is done. Sniffing here means to monitor and capture all the incoming packets to the local server. It can be done using Wireshark tool. When the DDoS attack is initiated using Ufonet, simultaneously start capturing the packets using Wireshark. At a certain time, targeted local server will be down, by then stop capturing the packets. Now we got the captured data which can be converted into csv file format, so that it can be fed into machine learning model for analysis.

3.3 Modeling Phase

The necessity of this phase is to create a machine learning model which can be used to detect DDoS attack. We have used preprocessed datasets of Canadian institute called as CICDDOS2019, which has 84 parameters for each different packet. We loaded the dataset and removed the unnecessary parameters such as expected time, arrival time. Then, we normalized the dataset and split the dataset for test and train. We compared five machine learning models and used ensemble method to build a single model.

3.4 Detecting Phase

The last step of the project is to deploy the ensembled machine learning model which is created in the first step. The model is deployed to the server so that the captured data can be fed into the model. Now model analyses the dataset and detects whether it is a DDoS attack or not. If DDoS is detected, send an alert.

3.4.1 Random Forest Classifier

The primary component of Random Forest classifiers is the decision tree. The hierarchical structure of the decision tree is constructed using the features of a data collection. The measure associated to a subset of the characteristics is used to segment the decision tree's nodes. A collection of decision trees and a set of bootstrap samples from the original dataset make up the Random Forest. The nodes are divided according to the entropy of a selected subset of the features. The size of the original dataset is maintained in the subsets produced by rebuilding from it. We trained the model using Random Forest Classifier with training sets as parameters and then predicted using testing data. We got a percentage accuracy of 78.47444 and F_1-Score of 0.71387.

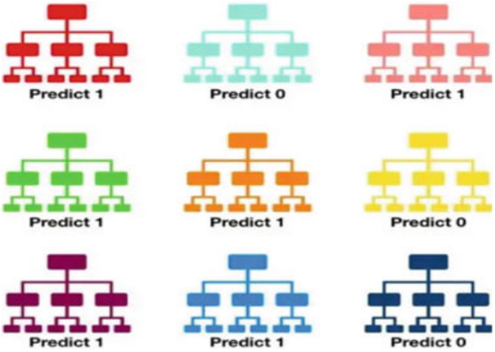


Image courtesy https://towardsdatascience.com/understanding-random-forest-58381e0602d2

3.4.2 Multi-layer Perceptron

Multi-layer perception is referred to as MLP. It is composed of layers that are dense and flawlessly connected, and it can transform any input dimension into any required dimension. A multi-layer perception is a neural network that has many layers. We combine neurons so that some of their outputs are also their inputs to create a neural network. We trained the model using Multi-layer Perceptron with training sets as parameters and then predicted using testing data. We got a percentage accuracy of 83.54524 and F_1-Score of 0.72593.

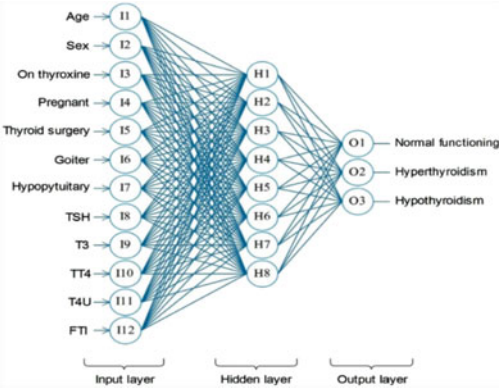


Image courtesy https://link.springer.com/article/10.1007/s11227-020-03404-w

3.4.3 Gradient Descent Classifier

To determine the parameters or coefficients of functions that minimize a cost function, one can use the straightforward yet effective optimization process known as stochastic Gradient Descent (SGD). To put it another way, it is employed in the discriminative learning of linear classifiers under convex loss functions, including SVM and logistic regression.

The Gradient Descent classifier should be fitted into two arrays:

(1) Array A holding training samples.
(2) Array B holding class labels for the training values.

We trained the model using Gradient Descent Classifier with training sets as parameters andthen predicted using testing data. We got a percentage accuracy of 83.51453 and F_1-score of 0.70151.

3.4.4 Decision Tree Classifier

A supervised machine learning algorithm called decision tree uses a set of principles to make judgments, much like how people do. A machine learning classification algorithm can be thought of as being created to make decisions. The model is typically said to forecast the class of the novel, previously unseen input, but the algorithm must choose the class to be assigned. The idea behind decision trees is to repeatedly partition the dataset until all the data points that belong to each class are isolated by using the dataset features to produce yes/no questions. We trained the model using decision tree classifier with training sets as parameters and then predicted using testing data. We got a percentage accuracy of 83.53213 and F_1-score of 0.64467.

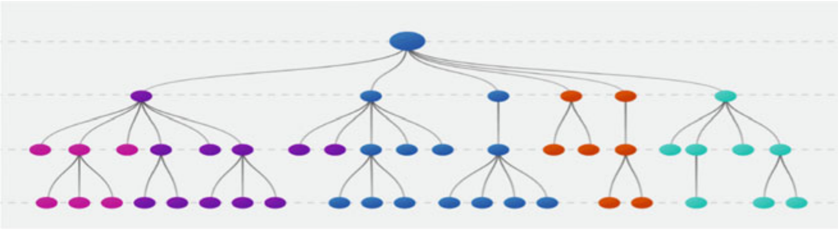


Image courtesy https://ai-pool.com/a/s/decision-trees

3.4.5 Radial Basis Function

In comparison to most neural network topologies, the architecture of radial basis function (RBF) networks is fundamentally different. In most neural network architectures, nonlinearity is introduced by repeatedly using nonlinear activation functions, which results in many layers and complexity. On the other hand, an RBF network only has an input layer, one hidden layer, and an output layer. The input layer does not do any computations; rather, it merely collects input data and sends it to the RBF network's unique hidden layer. The computation that takes place inside the hidden layer of the RBF network is extremely dissimilar to that of most other neural networks, and this is what gives it its power. The prediction methods, such as classification or regression, are carried out by the output layer. We trained the model using radial basis function with training sets as parameters and then predicted using testing data. We got a percentage accuracy of 85.45345 and F_1-Score of 0.7.5234.

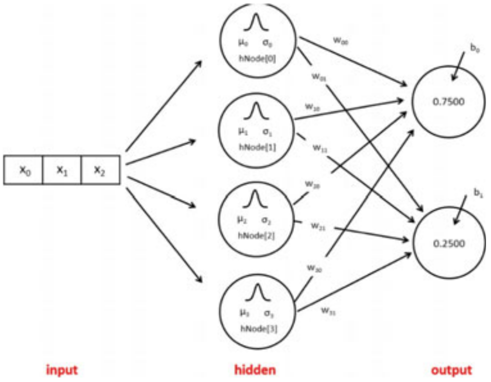


Image courtesy https://iq.opengenus.org/radial-basis-neural-network/

4 Results and Analysis

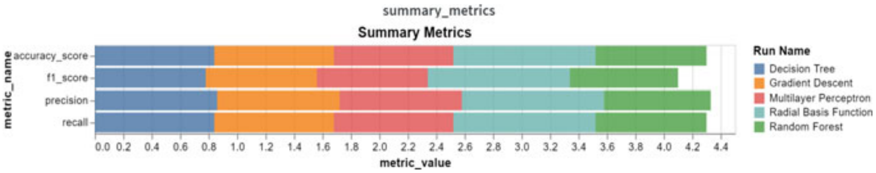
S. No.	Machine learning model	Parameters	Results	
			F_1-Score	Accuracy
1	Random Forest	Benign	1.00	78.47444
		Syn	0.87	
		UDP-lag	0.28	
		WebDDoS	0.70	
2	Multi-layer perceptron	Benign	1.00	83.54524

(continued)

(continued)

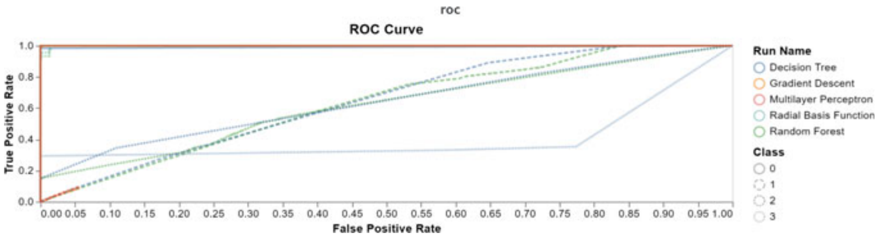
S. No.	Machine learning model	Parameters	Results	
			F_1-Score	Accuracy
		Syn	0.91	
		UDP-lag	0.26	
		WebDDoS	0.74	
3	Gradient descent	Benign	0.98	83.51453
		Syn	0.91	
		UDP-lag	0.26	
		WebDDoS	0.66	
4	Decision tree	Benign	0.98	83.53213
		Syn	0.91	
		UDP-lag	0.26	
		WebDDoS	0.44	
5	Radial basis function	Benign	1.00	85.45345
		Syn	0.96	
		UDP-lag	0.26	
		WebDDoS	0.82	

- Classification report



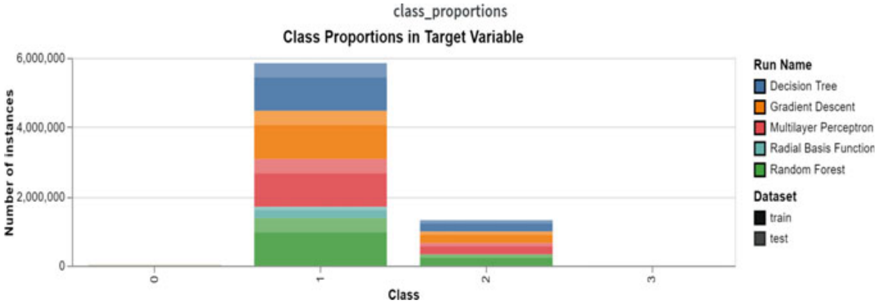
Above classification report shows the accuracy score, f_1-score, precision, and recall of the machine learning algorithms. According to the report, the radial basis function has the highest precision and accuracy of all others. Next to radial basis algorithm, decision tree is next second highest of them all.

- ROC curve



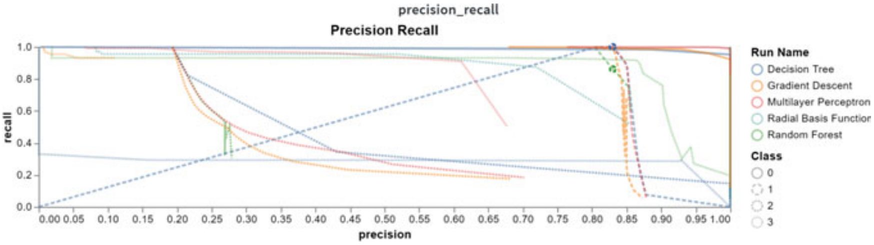
The Receiver Operating Characteristic curve, or ROC curve, is a graph that displays how well a classification model performs across all categorization levels. False Positive Rate and True Positive Rate are the two parameters that the aforementioned curve plots. The Efficient ROC curve can be seen as the closer an ROC curve is to the upper left corner, the more efficient is the model. From the above, the Curve Multi-layer Perceptron and Gradient Descent are closest to upper left corner. The efficient models are Multi-layer Perceptron and Gradient Descent.

- Class propositions



In the above graph, class proportions are depicted. In each color, there are two types of shades: dark and light which represent the training and testing of the machine learning model. The numbers in the *X*-axis represent the types of packets: 0-BENIGN, 1-SYN, 2-UDP-lag, 3-WebDDOS. From the above graph, test and train for radial basis graph are low at Syn dataset and high for the Random Forest model.

- Precision–Recall



Precision–recall graph depicts the precision for every model for different packets. The precision for decision tree for BENIGN packets near close to 1 whereas precision of decision tree is lower for WebDDOS packet. The precision for other ML models is nearly to 0.8, and only the Random Forest precision is low for BENIGN packets

5 Conclusion and Future Work

DDoS attacks are a real danger that harm many Internet users severely. Losses have progressed from being merely bothersome to truly being disastrous and crippling for certain users. There is every reason to think that DDoS attacks will become more frequent and severe. The current low level of losses brought on by DDoS attacks is presumably not the result of effective defense against them, challenges in carrying them out, or a dearth of desirable targets to attack. Application-based DoS attacks are far less resource-intensive for the attacker in terms of processing power and bandwidth, and as a result, they pose a higher danger to companies because there are a lot more potential threat agents. In our paper, we have briefly discussed the how to detect the DDoS attack via machine learning models and with the help of Neural networks. The future work of the project is to deploy the ensembled machine learning model and merge both the attack and detection phases so that the whole cycle of DDoS attack and detection will be completed. Finally, DDoS attacks cannot be stopped completely, but there are certain strategies that can be implemented, which can reduce the impact of the incoming attack in the future.

References

Amjad A, Alyas T, Farooq U, Tariq MA (2019) Detection and mitigation of DDoS attack in cloud computing using machine learning algorithm

Gupta BB, Gaurav A, Perakovic D (2021) A big data and deep learning based approach for DDoS detection in cloud computing environment

Jose S, Priyadarshini K, Abirami K (2016) An analysis of black-box web application vulnerability scanners in SQLi detection. In: Advances in intelligent systems and computing, vol 398, pp 177–1851, Jan 2016

Kannan Mani SM, Balaji Bharatwaj M, Harini N (2021) A scheme to enhance the security and efficiency of MQTT protocol. Smart Innov Syst Technol 194:79–93, cited 1 times. https://doi.org/10.1007/978-981-15-5971-6_9

Kautish S, Reyana A, Vidyarthi A (2022) Attack detection for DDoS AND IT's vulnerabilities in hybrid cloud environment

Kumar A, Somani G (2022) DDoS attack mitigation in cloud targets using scale-inside out assisted container separation

Lunkad D, Singh G (2020) DDOS attack detection using machine learning for network performance improvement

Mishra A, Gupta BB, Perakovic D, Penalvo FJG, Hsu C-H (2021) Classification based machine learning for detection of DDoS attack in cloud computing

Neethu MR, Harini N (2018a) Safe sonet: a framework for building trustworthy relationships. Int J Eng Technol (UAE) 7(2.26 special issue 26):57–62, cited 3 times. https://doi.org/10.14419/ijet.v7i1.8822

Neethu MR, Harini N (2018b) Securing image posts in social networking sites. In: Lecture notes in computational vision and biomechanics, vol 28, pp 79–91, cited 3 times. https://doi.org/10.1007/978-3-319-71767-8_7

Pei J, Chen Y, Ji W (2019) A DDoS attack detection method based on machine learning

Saini PS, Behal S, Bhatia S (2020) Detection of DDoS attacks using machine learning algorithms

Wani AR, Rana QP, Saxena U, Pandey N (2019) Analysis and detection of DDoS attacks on cloud computing environment using machine learning techniques

Zekri M, El Kafhali S, Aboutabit N, Saadi Y (2017) DDoS attack detection using machine learning techniques in cloud computing environments

Smart Grading System Using Bi LSTM with Attention Mechanism

Dnyanada Mahajan, Prachi Channe, Shruti Diwate, Sneha Kharate, and Rudragouda Patil

1 Introduction

In the rapidly evolving digital era, educational institutions are continually seeking innovative approaches to enhance their teaching and assessment methodologies. Grading, being an integral part of the educational process, plays a pivotal role in evaluating students' understanding and academic performance. However, traditional grading methods often suffer from inherent limitations, such as subjectivity, time-consuming nature, and potential biases. To address these challenges, this research paper proposes a Smart Grading System utilizing Long Short-Term Memory (LSTM) with Bi-LSTM and attention mechanism, which promises to revolutionize the conventional grading practices.

The traditional grading methods, primarily reliant on manual assessment, are prone to several shortcomings. Firstly, subjectivity arises due to the variance in interpretations and grading standards among different evaluators, leading to inconsistent and unfair evaluations. This subjectivity can negatively impact students' academic outcomes and create disparities within the educational system. Secondly, the extensive time required for manual grading often results in delayed feedback, hindering students' timely understanding of their performance and inhibiting their growth. Lastly, the labour-intensive nature of traditional grading limits scalability, particularly in large classrooms or institutions, making it challenging to provide prompt, and constructive feedback to students.

The proposed Smart Grading System leverages cutting-edge techniques such as LSTM with Bi-LSTM and attention mechanism, offering several significant advantages over traditional grading methods. Firstly, by employing advanced deep learning

D. Mahajan (✉) · P. Channe · S. Diwate · S. Kharate · R. Patil
School of Computer Engineering and Technology, MIT Academy of Engineering, Pune, India
e-mail: dnyanada02@gmail.com

R. Patil
e-mail: rgpatil@comp.mitaoe.ac.in

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_18

models, the system introduces an automated and objective approach to grading, minimizing the influence of subjectivity and ensuring consistent evaluations. This not only promotes fairness but also fosters a more transparent educational environment, benefiting students, teachers, and educational institutions alike.

Secondly, the integration of LSTM with Bi-LSTM allows the system to capture sequential dependencies and contextual information within students' answers. By understanding the sequential nature of student responses, the system can effectively analyse and evaluate their thought process, providing more insightful and accurate feedback. Moreover, the attention mechanism enables the system to identify salient features and key areas in students' answers, enhancing the granularity of assessment. Additionally, the Smart Grading System significantly reduces the grading time by automating the assessment process. By swiftly processing a large volume of student responses, it enables prompt feedback, allowing students to address their weaknesses and improve their performance in a timely manner. Moreover, the system's scalability ensures its applicability across various educational settings, accommodating the grading needs of both small-scale classrooms and large-scale educational institutions.

This research paper presents a pioneering Smart Grading System that employs LSTM with Bi-LSTM and attention mechanism to overcome the limitations of traditional grading methods. By addressing issues related to subjectivity, time-consuming evaluations, and scalability, this system revolutionizes the grading process, offering objective assessments, prompt feedback, and enhanced granularity. The subsequent sections of this paper will delve into the technical aspects, implementation details, and experimental evaluations of this grading system.

2 Related Work

Li and Yan (2012) developed a system that takes the part-of-speech tag and the words into consideration. Regression was employed in this system instead of classification using the Support Vector Machine (SVM) toolbox. It uses the basic model and Content Vector Analysis (CVA) (Li and Yan 2012) model for the topic detection component, and can efficiently determine whether an essay is off-topic, especially for huge numbers of essays. In addition, the system is improved by the similarity to full-score essays. Given the two score deviations in comparison to a human rater, it has an accuracy of 86%.

Ben and Hongbo Chenproposed a listwise learning to rank approach to automated essay scoring that incorporates human–machine agreement directly into the loss function. As baselines, this used classical machine learning algorithms, SVM for classification, regression, and preference ranking, respectively. Experiments on the public English dataset Automated Student Assessment Prize (ASAP) (Chen and He 2013) yielded Quadratic Weighted Kappa's in the 0.80 for prompt-specific rating and 0.78 for generic rating.

Weighted Score was used for the Automated Essay Grading System in the Matrix Laboratory (Oduntan 2018). Using coefficient of determination, it was evaluated how well the K-nearest neighbour classifier performed in comparison to the Cosine Similarity Measure in establishing the Weighted Score for the Automated Essay Grading System.

An Empirical Analysis of Machine Learning Models for Automated Essay (Goyal 2018) proposed a work that used support vector machines, K-nearest neighbour, and a classifier based on linear regression to develop multiple classification algorithms. According to this analysis, set 1's KNN classifier has a 92.23% accuracy rate with respect to class "B." The most effective classifier for Set 1 is KNN. SVM, which has an accuracy of 78.68%, is the top-performing classifier for Set 7.

The NN model for Automated Essay Grading scored 0.94 during the Automatic Essay Grading Kaggle Competition (Nguyen 2016) (Neural Networks for Automated Essay Grading, 2016) on the Quadratic Weighted Kappa metric. This neural network model employed a 300-dimensional Glove (Nguyen 2016) as the embedding layer initialization.

The study on essay scoring demonstrates how a NN with cross-sentence dependencies and a discourse-based training objective can outperform both feature-based state-of-the-art models and hierarchical LSTMs in terms of automatic essay scoring for the Test of English as a Foreign Language (TOEFL) (Farah Nadeem 2019) essay data. The best results are achieved with a model that learns the combination of hand-crafted features and the neural document representation with Quadratic Weighted Kappa of 0.852 and 0.736 for set 1 and set 2, respectively.

Research paper on essay scoring presents novel LSTM dependency tree transfer learning scoring method for short essays in Indonesia (Wiratmo and Fatichah 2020). The LSTM architecture for essay grading can take both sequence and dependency into account. There proposed technique offers QWK and accuracy results of 53.68% and 16.23%, respectively.

The Intelligent Grading System model (Zhu and Sun 2020) developed a straight forward grading scheme that achieves a Quadratic Weighted Kappa of 0.7 using machine learning and natural language processing. The tokenized sequences are evaluated using an LSTM neural network, while the vector representation is evaluated using a 2-layer neural network.

The Automatic Short Answer Grading model (Mahana 2012) employed a model to train and test across each collection of essays. Within each essay set, a fivefold cross validation was performed. The average Kappa value for this was 0.73.

The Short Answer Grading (SAG) (Wang et al. 2019) uses attention semi-supervised method to train model. It incorporates rule-based model top of regular expression matching and span matching methods. Model outperforms with QWK of 0.61 for training size of 200.

The Automatic Short Answer Grading (ASAG) (Galhardi et al. 2020) was applied to 9885 answers. Six different approaches were selected to model answers and combined as soft-6. Both linear and Quadratic Kappa Scores were within the range of 0.4 and 0.6.

The Automatic Essay Scoring Transformer Models (Ludwig et al. 2021) compares traditional algorithms along with RNN. It implements Bidirectional Encoder Representations from Transformers (BERT) and regression models with evaluation metrics as accuracy, Cohen's Kappa of 0.94 and 0.59, respectively, for large datasets.

3 Smart Grading System

3.1 Task Description

The objective of this research is to develop a machine learning-based, intelligent system that can grade essays on its own. A dataset with a significant number of essays on a certain topic should be carefully picked in order to ensure consistency among the human raters. Our dataset has enough essays on various topics that have been graded. Pre-processing of the dataset is the next step. Cleaning the data is the first stage in the pre-processing process. The process of cleaning data involves removing any inaccurate, incomplete, duplicate, or other wrong data from the dataset. The removal of all characters from the dataset that aren't alphabets is the second step in the data cleaning process. The stop words are then all eliminated from the text. To get rid of stop words, the text is broken up into words, and those words are eliminated if they appear on the NLTK (Bird 2006) list of stop words.

Word tokenization is then applied to the words. The phrases, sentences, and paragraphs in this passage are divided into many units. Tokens are the name for these more compact objects. Then, these tokens are further examined in order to categorize or count them according to a specific sentiment. Then, using word embedding, we create a Word2Vec model in which the words or phrases are translated into real-number vectors. When words are embedded, those that share a semantically similar meaning are closer together than those that do not. These are then passed to the 3 layered LSTM model. There are a total three models with different layer combinations but common activation function and evaluation metrics. The activation metrics used is ReLU which is explained in the further section of this paper. We have used Quadratic Weighted Kappa as the evaluation metric for the models (Fig. 1).

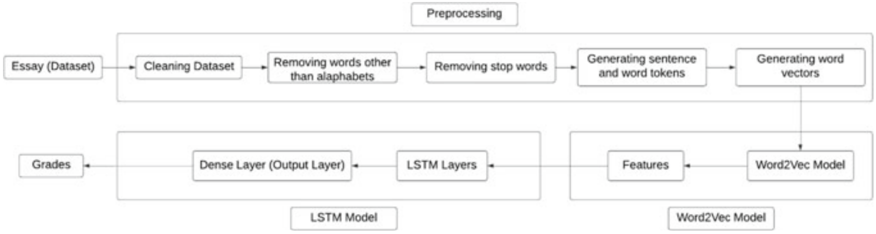


Fig. 1 Data flow in the model

4 System Design

This section introduces the proposed methodology. We have implemented Recurrent Neural Network (RNN) (Rani 2021) which are the most commonly used neural network for problem solving by researchers. RNN is a type of neural network which uses the output from the previous layer to feed as input to the next layer. Paper consists of implementation and observation of the comparative working of Long Short-Term Memory (LSTM), Bidirectional-LSTM (Bi-LSTM), and Bi-LSTM with attention layer (Wang et al. 2018). LSTM is one of the types of RNN which is capable of solving complex problems. This section provides a brief description about the architecture and working of mentioned LSTM models and the training phase.

4.1 Word2Vec Model (Word Embedding Model)

In the Word2Vec model, each given essay consists of a number of words, and each word is represented by a word embedding according to Word2vec. The embedding representations are expected to catch the semantic information carried by each word, i.e. the words with similar meanings will be near to each other in the vector space.

For example, from our corpus, we can find that "computer" is similar to "laptop."

The neural network can learn to identify the vector of the input word by using the surrounding word of this input. That means if two different words have the same context, the network tends to give them similar word vectors. Hence, every word in our dataset has a unique vector containing the latent semantic and the vectors of the words in one essay can combine to an essay matrix which is the input unit of our scoring machine.

4.2 3-Layer LSTM Model

The paper first introduces the 4-layer LSTM model. It is a sequential model and works better for chosen dataset which is a corpus of essays in text format. The first layer of the model is the word embedding layer. The layer takes 300 as the first argument which is the number of features (output generated from word embedding layer, i.e. Word2Vec model), dropout and recurrent dropout as 0.4, respectively, and input size from 1 to 300 that is length of each sentence sequence. The next layer takes 64 features as the first argument with recurrent dropout of 0.4. Third layer is also a dropout layer with a dropout value of 0.5. The final layer is a dense layer, it reduces the dimensionality to 1 which is the predicted score. Model have uses ReLU activation function in the dense layer so that the score can be predicted correctly since the values of ReLU function ranges form $-\infty$ to $+\infty$. For fitting of training data, models have been passed through batch size of 64 and 100 epochs. These epochs are

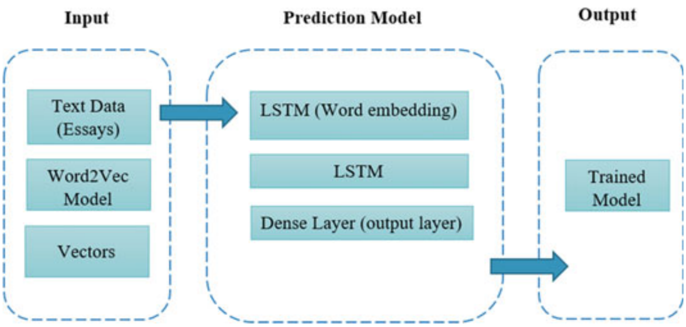


Fig. 2 LSTM model

varied according to the size of the test data. The model produces effective results. Diagrammatic representation is given in Fig. 2. Equations of LSTM are as follows:

$$f_t = \sigma_g\left(W_f \times x_t + U_f \times h_{t-1} + b_f\right)$$

$$i_t = \sigma_g(W_i \times x_t + U_i \times h_{t-1} + b_i)$$

$$o_t = \sigma_g(W_o \times x_t + U_o \times h_{t-1} + b_o)$$

$$c'_t = \sigma_c(W_c \times x_t + U_c \times h_{t-1} + b_c)$$

$$c_t = o_t.c_{t-1} + i_t.c'_t$$

$$h_t = o_t.\sigma_c(c_t)$$

where x_t and h_{t-1} are the input and output vectors at time *t*, respectively. W_i, W_f, W_c, W_o, U_i, U_f, U_c, and U_o are weight matrices. b_i, b_f, b_c, and b_o are bias vectors. f_t is the forget gate, i_t is the input gate, o_t is the output gate, c_t is the cell state, h_t is the hidden state, σ_g is sigmoid activation, σ_c is tanh activation function, and '.' is element wise multiplication.

4.3 LSTM with Bi-LSTM

To improve the performance/results obtained in Unidirectional LSTM, next we implemented the Bi-LSTM model (Alikaniotis et al. 2016). Unlike standard LSTM, here, the input is allowed to flow in both directions and it is capable of utilizing information from both sides. Again, it is powerful while modelling the sequential

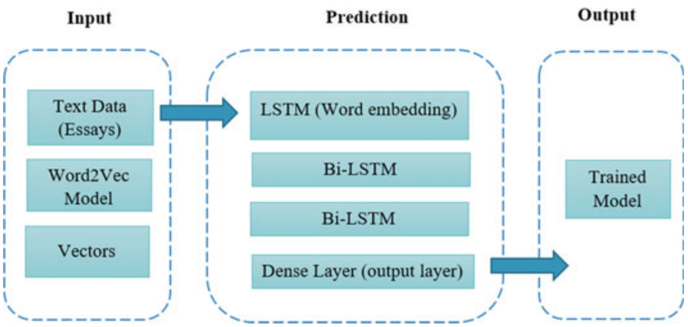


Fig. 3 LSTM with Bi LSTM model

data, improving dependencies between words and sentences in both directions. It adds one more LSTM layer to the previous model, the direction of information flow is revered. Then the model combines the output of both LSTM layers to get the final output. Bi-LSTM is considered to produce more meaningful output when both LSTMs are combined. All other parameters are kept the same. Experiments prove that Bi-LSTM model performs better than the standard LSTM model. Diagrammatic representation is given in Fig. 3. Equations of Bi-LSTM are as follows:

$$\hat{y}^{\langle t\rangle} = g\left(W_y\left[\vec{a}^{\langle t\rangle}, \overleftarrow{a}^{\langle t\rangle}\right] + b_y\right)$$

where $\hat{y}$ output at time t.

4.4 LSTM and Bi-LSTM with Attention Layer

Though, using Bi-LSTM gave better results, in order to improve model performance, attention layer (Wang et al. 2018) is introduced to the model. Sometimes, basic LSTM gets confused between the words and can predict the wrong word. So, in order for the encoder to search for the most relevant information, models have been introduced with an additional layer of attention mechanism. By applying the attention mechanism, the model will be able to effectively extract the information between essays through inter-sentence alignment and gain better performance. Diagrammatic representation is given in Fig. 4.

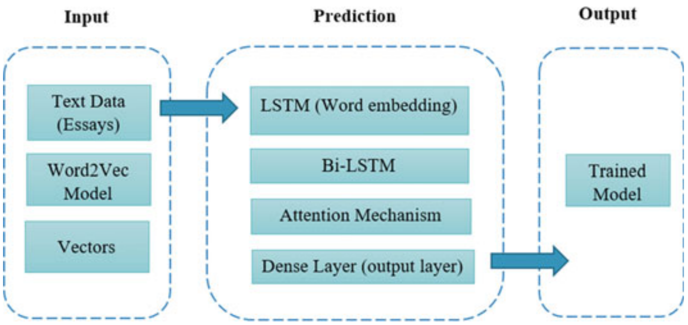


Fig. 4 LSTM and Bi LSTM with attention

4.5 Activation Function

Layers of nodes make up a neural network, which may be trained to map instances of inputs to outputs. The inputs are multiplied by the weights of a node and added together for a certain node. The node's total activation is known as this value. The activation total is then modified using an activation function, which determines the node's specific output or "activation." The simplest activation function, where no transform is used at all, is known as linear activation. A network that exclusively uses linear activation functions can be trained relatively quickly, but it is unable to learn complex mapping functions. It is preferable to use nonlinear activation functions since they enable the nodes to understand more intricate data structures. The sigmoid and hyperbolic tangent activation functions are two widely utilized nonlinear activation functions. The sigmoid and tanh functions both saturate, which is a general issue. This indicates that for tanh and sigmoid, high values snap to 1.0 and small values snap to -1 or 0. Additionally, sigmoid and tanh are only really sensitive to input changes around their midpoints, or 0.5 and 0.0, respectively. That's why we decided to use the ReLU activation function in our model.

ReLU is the most commonly used activation function in machine learning models. Any negative input causes the function to return 0, but any positive value x causes it to return that value. Thus, it may be expressed as:

$$f(x) = \max(0, x)$$

5 Experiment Setup and Evaluation Metrics

5.1 Setup

Dataset used in this paper is "The Hewlett Foundation: Automated Essay Scoring" on Kaggle. The dataset includes 8 essay sets. Each of the essay sets was generated from an individual prompt. Average length of essays is in the range of 150–550 words per response. All essays were hand graded and were double-scored. The training data is in the format of tab-separated value (TSV) file. There are total 3 scores, i.e. rater 1 score, rater 2 score, and domain score. The domain score is an addition of rater 1 and 2 scores. There are some unwanted empty columns also present in the dataset; these columns were dropped using pandas' libraries. Hence the final dataset has essay_id, essay_set, essay, domain1_score columns.

After generation of the dataset the data needed to be pre-processed before passing on to the model. The essays were first cleaned by removing the stop words, punctuation marks, and converting all characters to lowercase. Once the data is cleaned the next step is to generate a feature vector that is to be passed to the Word2Vec model. In order to generate the feature vectors first the cleaned essay was converted into sentence tokens and finally to word tokens. The output of the Word2Vec model was passed as an input to the LSTM layers. For models other than neural network word embeddings were generated.

5.2 Evaluation Metrics

Quadratic Weighted Kappa

A set of predictions and a set of multiclass labels are measured by the Quadratic Weighted Kappa index. It attempts to take into account the similarity between the classes, beyond only the class, rather than just focusing on the precision of the match between predictions and labels. As a gauge of agreement between observed raters in cross-classification, Cohen's Weighted Kappa is frequently used. When ratings are given on nominal scales without an order structure, an appropriate index of agreement is used. Calculation of Cohen's Kappa may be performed according to the following formula:

$$K = \frac{(\Pr(a) - \Pr(e))}{1 - \Pr(e)}$$

where $\Pr(a)$ represents the actual observed agreement, and $\Pr(e)$ represents chance agreement. Generally, a Kappa of less than 0.4 is considered poor (a Kappa of 0 means there is no difference between the observers and chance alone). Kappa values of 0.4–0.75 are considered moderate to good and a Kappa of > 0.75 represents excellent agreement. A Kappa of 1.0 means that there is perfect agreement between all raters.

Mean Squared Error

Mean squared error also known as MSE. It defines the square of the absolute difference between actual and predicted value. Here, squares avoid the cancellation of negative terms.

$$\text{MSE} = \frac{1}{n(y - y')^2}$$

6 Results and Discussion

In this section the results of comparison of different machine learning models are explained and further the comparison between different LSTM layers is also proposed. Previously research has been done for which machine learning model to be used. A bunch of different ML models were selected and QWK was calculated for each model, Table 1 shows the results of different machine learning models out of which LSTM gives best performance in terms of QWK as well as MSE. Hence the LSTM model was selected for further experiment purposes and predictions of grades.

Once the model was finalized, various combinations of layers in the LSTM were used for better results. The models were modified with different LSTM, Bi LSTM, and attention layers. The combination of layers used are as follows:

1. LSTM + LSTM
2. LSTM + Bi LSTM + Bi LSTM
3. LSTM + Bi LSTM + Attention.

Hyperparameter for all of the models were set as constant with batch size of 64, 50 epochs, and activation function as ReLU. Fivefold cross validation was used in each model. The results of QWK and MSE were noted. The results of this comparative study of layers are given in Table 2. It is observed that the model that contains the attention layer performs best amongst all the three models.

Further different activation functions were used and the results were noted. The results are shown in Table 3. For sigmoid and tanh, the QWK is zero and the MSE is

Table 1 Comparative study of different machine learning models

Models	Kappa score	Mean squared error
Linear regression	0.85	20.94
SVR	0.64	38.19
LSTM	**0.94**	7.47
K-nearest neighbours	0.92	14.7
Decision tree classifier	0.86	22.56

Table 2 Comparative study of different LSTM layers

Model	Activation function	Epochs	Batch size	QWK	MSE
LSTM (2)	ReLu	50	64	0.94	7.86
LSTM + Bi LSTM (2)	ReLu	50	64	0.95	6.66
LSTM + Bi LSTM + Attention	ReLu	50	64	0.96	6.2

very large as compared to the other models. The noted results were not satisfactory and ReLU was decided to be used as activation function for building the final model.

Results were also noted for usage of different batch sizes in all models. The results are shown in Table 4. The combinations of batch sizes used were 32, 64, and 128. As the standard batch size used is 64 and from our comparison, we found that model with batch size 64 is stable and performs well so we have selected batch size as 64.

Table 3 Using different activation function

Model	Activation function	Epochs	Batch size	QWK	MSE
LSTM (2)	Sigmoid	50	64	0	110.45
	tanh	50	64	0	109.33
	ReLu	50	64	0.94	7.86
LSTM + Bi LSTM (2)	Sigmoid	50	64	0	124.36
	tanh	50	64	0	114.17
	ReLu	50	64	0.95	6.66
LSTM + Bi LSTM + Attention	Sigmoid	50	64	0	114.68
	tanh	50	64	0.95	112.23
	ReLu	50	64	0.96	6.2

Table 4 Using different batch size

Model	Activation function	Epochs	Batch size	QWK	MSE
LSTM (2)	Sigmoid	50	32	0.96	6.35
	tanh	50	64	0.94	7.86
	ReLu	50	128	0.95	7.82
LSTM + Bi LSTM (2)	Sigmoid	50	32	0.93	6.35
	tanh	50	64	0.95	6.66
	ReLu	50	128	0.92	6.56
LSTM + Bi LSTM + Attention	Sigmoid	50	32	0.95	6
	tanh	50	64	0.96	6.2
	ReLu	50	128	0.96	6.5

Table 5 Prediction model QWK

Model	QWK
LSTM (2)	0.924215844
LSTM + Bi LSTM (2)	0.956398147
LSTM + Bi LSTM + Attention mechanism	0.962121614

Table 6 Predictions

Essay Id	Model	Actual score	Predicted score
457	LSTM (2)	10	9 (9.16)
	LSTM + Bi LSTM (2)	10	9(8.558)
	LSTM + Bi LSTM + Attention	10	10(10.113)

Hyperparameters were set to constant after the comparison of various combinations of parameters. Moving forward with the experiment, prediction models were made and prediction QWK of all the selected models were calculated. It is shown in Table 5.

Predictions were made using all the three models generated with selected random essays from the dataset. The results are shown in Table 6. It was observed that the model with LSTM, Bi LSTM, and attention layer performs well as compared to the other two models.

7 Conclusion

In this paper, we introduced a neural network model with a combination of different layers for essay scoring. For selection of the neural network model various machine learning algorithms were compared and the model with best Kappa Score was selected. After finalizing the model different combinations of LSTM layers were used to train the model. It was observed that the model with combination of all three layers, i.e. LSTM, Bi LSTM, and attention gave good QWK and variance. Also, the predictions made by this model were best as compared to the predictions made by the other models. For the selected best model for predictions amongst the three, we calculated the prediction QWK and it was observed that the model with LSTM, Bi LSTM, and attention layer has the best QWK, i.e. 0.96.

8 Future Scope

Though our model gives good results with a Kappa Score of 0.96 there is always a scope to improve your model. The proposed model works well with the typed essays, further it can be modified for handwritten essays as well as essays in different languages. Not only essays there are also other modes of question and answers for which this model can be modified for grading. Short answers, long answers, answers specific to questions can also be graded.

References

Aldea AI, Haller SM, Luttikhuis MG (2020) Towards grading automation of open questions using machine learning. In: van der Veen J, van Hattum-Janssen N, Järvinen H-M, de Laet T, ten Dam I (eds) Engaging, engineering, education: book of abstracts, SEFI 48th annual conference University of Twente (online), 20–24 Sept 2020. University of Twente, pp 584–593

Alikaniotis D, Yannakoudakis H, Rei M (2016) Automatic text scoring using neural networks. arXiv preprint arXiv:1606.04289

Bernius JP, Krusche S, Bruegge B (2021) A machine learning approach for suggesting feedback in textual exercises in large courses. In: Proceedings of the eighth ACM conference on learning at scale. Association for Computing Machinery, New York, NY, USA, pp 173–182. https://doi.org/10.1145/3430895.3460135

Bird S (2006) NLTK: the natural language toolkit. In: Proceedings of the COLING/ACL on interactive presentation sessions (COLING-ACL'06). Association for Computational Linguistics, USA, pp 69–72. https://doi.org/10.3115/1225403.1225421

Chauhan RK, Saharan R, Singh S, Sharma P (2020) Automated content grading using machine learning. arXiv preprint arXiv:2004.04300

Chen H, He B (2013) Automated essay scoring by maximizing human-machine agreement. In: EMNLP

Rani AM (2021) Automated explanatory answer evaluation using machine learning approach. Des Eng 1181–1190

Farah Nadeem HN (2019) Automated essay scoring with discourse-aware neural models. Florence, Italy

Galhardi L, de Souza RCT, Brancher J (2020, November) Automatic grading of Portuguese short answers using a machine learning approach. In: Anais Estendidos do XVI SimpósioBrasileiro de Sistemas de Informação. SBC, pp 109–124

Goyal A (2018) An empirical analysis of machine learning models for automated essay grading. https://doi.org/10.7287/peerj.preprints.3518v1

Krithika R, Narayanan J (2015) Learning to grade short answers using machine learning techniques. In: Proceedings of the third international symposium on women in computing and informatics (WCI'15). Association for Computing Machinery, New York, NY, USA, pp 262–271

Li Y, Yan Y (2012) An effective automated essay scoring system using support vector regression. In: 2012 fifth international conference on intelligent computation technology and automation, pp 65–68. https://doi.org/10.1109/ICICTA.2012.23

Liu T, Ding W, Wang Z, Tang J, Huang GY, Liu Z (2019) Automatic short answer grading via multiway attention networks. In: Isotani S, Millán E, Ogan A, Hastings P, McLaren B, Luckin R (eds) Artificial intelligence in education. AIED 2019. Lecture notes in computer science, vol 11626. Springer, Cham. https://doi.org/10.1007/978-3-030-23207-8_32

Ludwig S, Mayer C, Hansen C, Eilers K, Brandt S (2021) Automated essay scoring using transformer models. Psych 3(4):897–915

Mahana MJM (2012) Automated essay grading using machine

Matt UV (1994) Kassandra: the automatic grading system. Institute for Scientific Computing, ETH Zurich, Switzerland

Süzen N, Gorban AN, Levesley J, Mirkes EM (2020) Automatic short answer grading and feedback using text mining methods. Procedia Comput Sci 169:726–743. ISSN 1877-0509

Nguyen H (2016) Neural networks for automated essay grading

Oduntan OE (2018) Performance analysis of K-nearest neighbour classifier and cosine similarity measure in generating weighted scores for an automated essay grading system

Ormerod CM, Malhotra A, Jafari A (2021) Automated essay scoring using efficient transformer-based language models. arXiv preprint arXiv:2102.13136

Sanchez V, Nerbonne J, Verspoor M (2013) Identifying important factors in essay grading using machine learning

Sharma A, Urkude V, Sonawane A, Sharma N (2021) Smart essay grader system. 3:2582–5208

Wang Z, Liu J, Dong R (2018) Intelligent auto-grading system. In: 2018 5th IEEE international conference on cloud computing and intelligence systems (CCIS), pp 430–435. https://doi.org/10.1109/CCIS.2018.8691244

Wang T, Funayama H, Ouchi H, Inui K (2019) Incorporating rubrics to short answer grading

Wiratmo A, Fatichah C (2020) Assessment of Indonesian short essay using transfer learning Siamese dependency tree-LSTM. In: 4th international conference on informatics and computational sciences (ICICoS)

Zhu W, Sun Y (2020) Automated essay scoring system using multi-model machine learning, pp 109–117. https://doi.org/10.5121/csit.2020.101211

Analysis and Prediction of Polycystic Ovarian Syndrome Using ML Classifiers

Lavanya Sanjay, Neha Nayak, Disha Sriram, Ashwini Kodipalli, Trupthi Rao, and Shoaib Kamal

1 Introduction

"Having PCOS has zero contingency on your worth and value as a woman. Know that you didn't do this to yourself, your body is not broken, and there is nothing wrong with you." As one might have already guessed, this research paper mainly deals with The Analysis and Prediction of Polycystic Ovarian Syndrome [PCOS].

Polycystic Ovarian Syndrome is a genetic, hormonal, metabolic and reproductive disorder, one of the main factors for infertility in women. If one has PCOS, they might not have periods intermittently or have periods that last longer than usual. This also increases the possibility of being prone to other serious conditions such as anxiety and depression, endometrial cancer, obesity and cardiovascular disease. This usually affects one in ten women. 20% women of childbearing age are estimated to have PCOS. 50% of the women with PCOS go undiagnosed, and 50% women with PCOS are at the risk of developing type 2 diabetes or prediabetes before the age of

L. Sanjay (✉) · N. Nayak · D. Sriram · A. Kodipalli (✉) · T. Rao · S. Kamal
Department of Artificial Intelligence and Data Science, Global Academy of Technology, Bangalore, Karnataka, India
e-mail: 1ga20ad020@gat.ac.in

A. Kodipalli
e-mail: dr.ashwini.k@gat.ac.in

N. Nayak
e-mail: 1ga20ad028@gat.ac.in

D. Sriram
e-mail: 1ga20ad013@gat.ac.in

T. Rao
e-mail: trupthirao@gat.ac.in

S. Kamal
e-mail: shoaib.kamal@gat.ac.in

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_19

40. The main symptoms of PCOS include irregular periods, excess production of androgen hormone, mood swings, depression and obesity among other factors. The precise cause of PCOS is uncharted, but an early diagnosis and regulated hormonal treatment accompanied with weight loss will lower the risk of the above-mentioned long-term obstacles.

Our approach mainly focuses on diagnosing PCOS with the help of certain easily available and simple factors. For doing so, we selected a dataset that had about 44 attributes and finalised top 11 attributes for further study and analysis. Following are the attributes along with their descriptions.

Ovarian follicles are small fluid filled sacs found in the ovary. These sacs secrete hormones that tend to influence the stages of the menstrual cycle and have the capability to release an egg during fertilisation. Follicles and their size play a vital role when it comes to fertility.

- Follicle No. (R): The number of follicles on the right ovary
- Follicle No. (L): The number of follicles on the left ovary
- Skin darkening (Y/N)
- Hair growth (Y/N)
- Weight gain (Y/N)
- Cycle (R/I)
- Fast food (Y/N)
- Pimples (Y/N)
- Weight (kg)
- BMI (Body Mass Index)
- PCOS (Y/N): This is an output label which basically points out if a person has PCOS (Y) or does not have PCOS (N).

2 Literature Survey

According to the literature survey by Danaei et al. (Mehr and Polat 2022; Subha et al. 2022; Denny et al. 2019) on the diagnosis of PCOS, the steps to predict include reducing overfitting by filtering and performing feature selection on the dataset. After reducing the dimensionality of the dataset, the classification (diagnosis) is done using ensemblers. They have obtained an accuracy of 99.08% using AdaBoost, MLP and Random Forest algorithms.

This approach can also be used to detect other medical illnesses like auto-immune disorders, diabetes, autism spectrum disorder (ASD), etc., thus linking machine learning and human health.

In the diagnosis done by Subha et al. (Xie et al. 2020; Zhang et al. 2021; Bharati et al. 2020; Suha and Islam 2022; Tiwari et al. 2022), the concepts of Swarm Intelligence and Flashing Firefly algorithm are used to intelligently select the features using statistical ML methods like Chi square and correlation. On this reduced dataset with optimal number of features, the classification is done using Random Forest after the train-test validation. The method used to calculate the performance is using the classification metrics, namely recall, precision, specificity, accuracy and also using the F1 score.

In the prediction system developed by Denny et al. (Nandipati et al. 2020; Thomas and Kavitha 2020; Hassan and Mirza 2020; Aggarwal and Pandey 2021), the feature selection is done using an advanced analytics tool SPSS V 22.0 developed by IBM for multivariate analysis, thus reducing the features from 23 to 8. On this dataset multiple classification algorithms are applied to test which algorithm provides the best accuracy. Out of multiple algorithms like CART, SVM, KNN, RF, the best accuracy was obtained using Random Forest (89.02%) and is therefore used in prediction of PCOS.

In the work done by Ning-Ning et al. (Silva et al. 2022; Chauhan et al. 2021; Krishnaveni 2019; Mehrotra et al. 2011), they have linked PCOS detection with the concept of genetics. Gene samples are collected from samples, and it is made to undergo a sequence of tests and computations to analyse and understand the difference in the genetic makeup of women with and without PCOS. Most of the analysis is done graphically using area under curve (AUC) and artificial neural networks (ANN). As a product, they have developed a model that uses the RNA genotype as an input and makes the required predictions.

The study by Zhang et al. (Gupta et al. 2022; Soni and Vashisht 2018; Deepika 2019; Vikas et al. 2018) involves tracking the menstrual cycles of different women and studying the change in the follicular stages via images. Raman spectroscopy is used here to analyse the fluid in the follicles. PCA was done and various gradient boost algorithms were applied for easy distinguishing between PCOS versus non-PCOS women. This kind of diagnosis is proved to be very accurate and effective.

In the work done by Bharati et al. (Rachana et al. 2022a; Guha et al. 2022a; Rao 2022; Sagarnal et al. 2021), FH and LSH are considered to be the primary hormones that cause PCOS among the other 8 primary factors. 40 fold cross-validation is done on the dataset. A hybrid of two algorithms is employed for classification: Random Forest and Logistic Regression.

Using the RFLR a high testing accuracy is obtained (91%) with a recall of 90%.

The research by Suha et al. (Bhoomika et al. 2022; Kodipalli et al. 2022a; Ruchitha et al. 2021) involves the usage of computer-based systems and convolutional neural networks to detect ovarian cysts instead of physical-manual methods. Feature selection from images is done using transfer learning method. Here, the usage of various ensembling techniques is involved along with bagging and boosting techniques. The model is trained, saved, and results are finalised using VGGNet16, a 16 layer object detection technique of CNN model. As it is a very deep convolutional network, the accuracy obtained is almost 99.79%

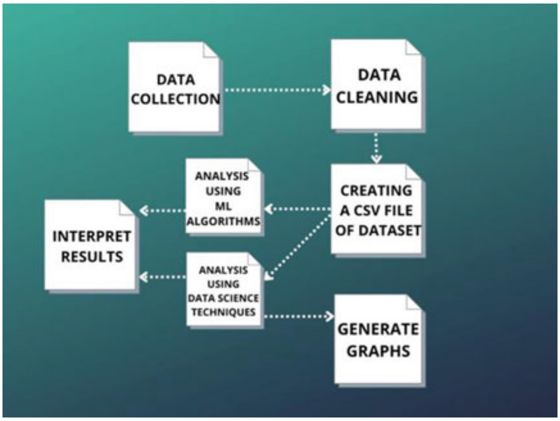
Tiwari et al. (Kodipalli et al. 2022b; Ruchitha et al. 2022; Gururaj et al. 2022) have utilised out of bag error to analyse the predictions made by Random Forest algorithm after data cleaning and filtering. They have used multiple classification algorithms alongside boosting techniques and meta learners to provide visualisations for factors affecting PCOS.

The dataset used here is scanned ultrasound images. Also, the assessment of performance of multiple algorithms done in this work helps analyse the accuracy of the various ML classification algorithms.

In the analysis done by Nandipati et al. (Guha et al. 2022b; Rachana et al. 2022b; Zacharia and Kodipalli 2022; Kodipalli and Devi 2021) on RapidMiner using KNN-SVM and Random Forest, two important observations were made, one being the accuracy of both algorithms were very high and very similar (94%) and the other observation being that the optimal factors could be chosen based on the performance by the two models.

In the PCOS study done by Thomas et al. (Bhagwani et al. 2021; Dhanush et al. 2021), data was collected from hospital forms, processed, filtered and classified using MATLAB. Neural network simulation was done effectively using feed forward propagation. Later in the study hybridisation of NN was done with Naive Bayes classifier to obtain a stronger accuracy of classification, thus concluding that the accuracy produced by hybrid classifiers is higher than a single high-performance classifier.

3 Methodology


DATA COLLECTION
DATA CLEANING
ANALYSIS USING ML ALGORITHMS
CREATING A CSV FILE OF DATASET
INTERPRET RESULTS
ANALYSIS USING DATA SCIENCE TECHNIQUES
GENERATE GRAPHS

1. **Gathering the data**

This dataset contains all parameters, both physical and clinical to determine PCOS and infertility related issues. There are a total of 44 parameters. This data is collected from various hospitals located across Kerala in India.

2. **Exploratory data analysis (EDA)**

We have analysed the dataset with respect to its size, shape, data types, information and headings.

3. **Splitting the dataset**

The dataset is split into training and testing sets where the model is trained and is later tested. This has been split in 70:30 ratios.

4. **Applying the models**

We have applied Logistic Regression and Random Forest algorithms.

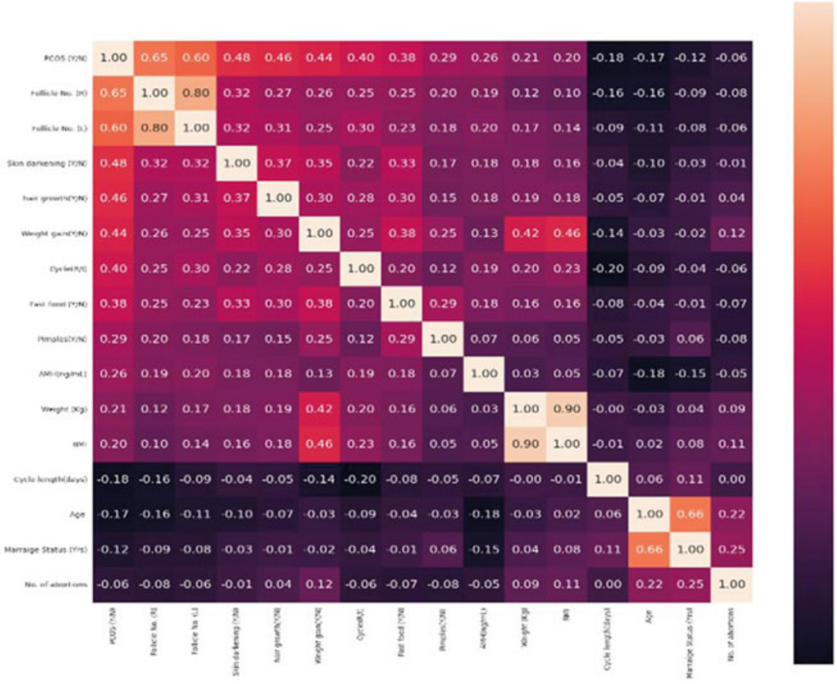
- **Logistic Regression**: A Logistic Regression model is a classification algorithm that analyses the link between one or more independent variables and predicts the dependent variable.
- **Random Forest**: A classification algorithm which is a collection of multiple decision trees.
- **Prediction**

When an array of inputs are given to a model, the model is able to predict if a female has PCOS or not.

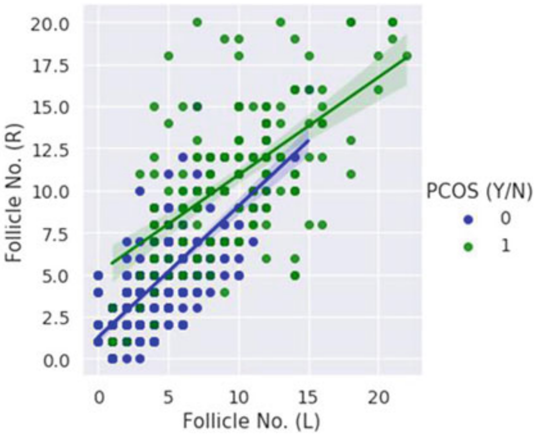
6. **Result**

Displays the output saying if the female is diagnosed with PCOS.

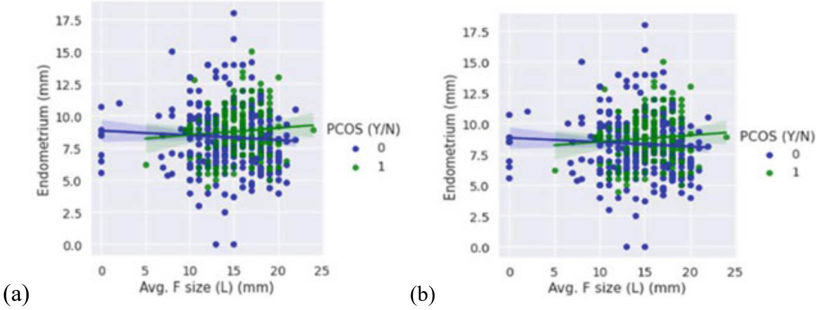
4 Results



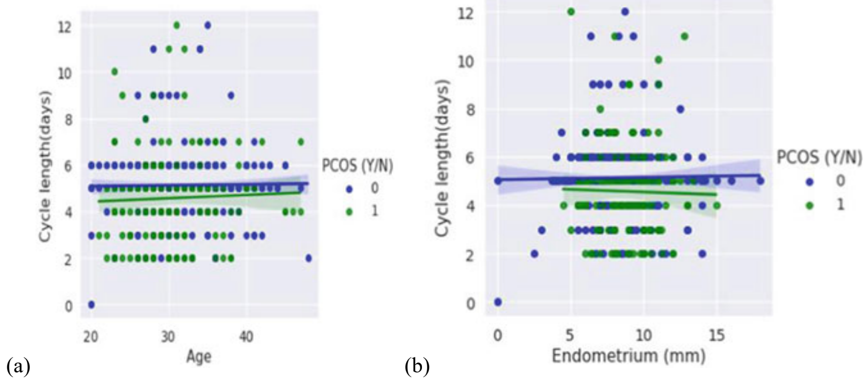
Heat map is plotted where we have selected a few features based on the importance of it medically as well as programmatically using a correlation matrix as initially the dataset consisted of about 44 attributes.



In the graph above, comparison between the follicle count in the left and right ovary is visualised, and we observe that the range varies for women with PCOS, it is slightly higher while compared to those who are tested negative for PCOS. The distribution of number of follicles in the ovaries is not equal for normal women and women with PCOS.

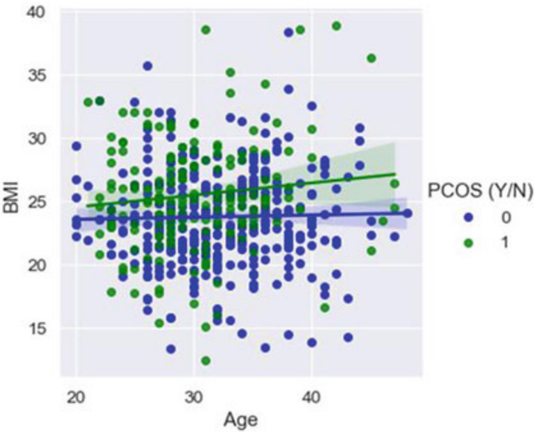


The graph depicts the follicular study and hence assesses both the size of the follicle and the thickness of the Endometrium lining. (a) For the women who are likely to have PCOS, we observe an increasing trend and decreasing for those who do not have PCOS. (b) Here the trends are almost similar with slight variation in the range. All in all, though graph (a) shows a drastic change, we cannot infer a lot from the above as medically they are not closely related.

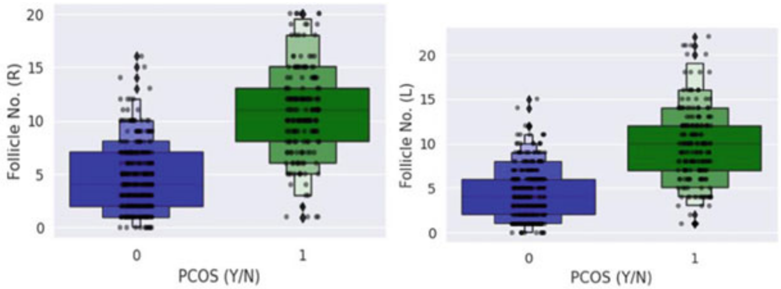


(a) When the comparison between Age and Cycle length is performed, we observe that as the woman ages, her cycle length remains almost constant for those who are tested negative for PCOS, but for women who have PCOS, the cycle length has shown slight increase as their age increases.

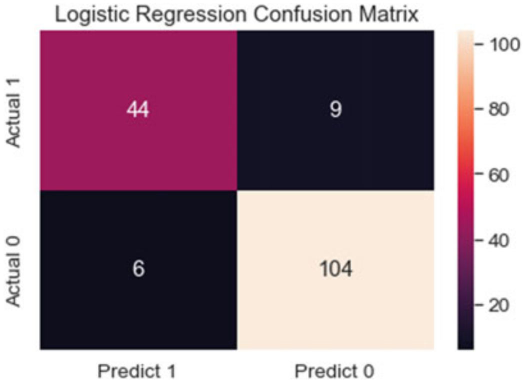
(b) The comparison between Endometrium and Cycle length gives us the following insights. For women who have PCOS, the cycle length has slightly decreased with increase in Endometrium thickness but for those who do not have PCOS the cycle length slightly increases with increase in Endometrium thickness.



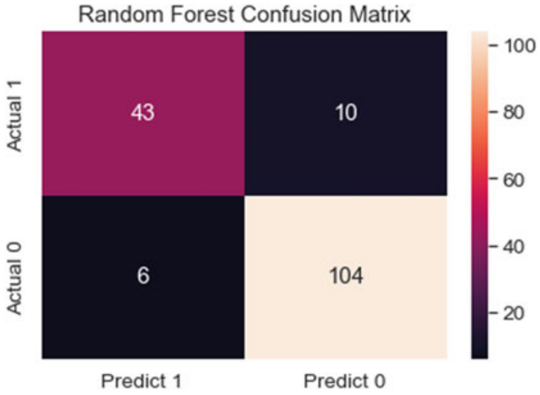
As observed in the above graph Age and BMI are compared, women with PCOS show an increase in the Body mass index with increase in their age, whereas for women without PCOS BMI remains almost consistent throughout. Body mass index (BMI) is consistent for non-PCOS women, whereas the BMI increases with age for women with PCOS.



In the above box plots, we have utilised only the follicle size to conclude [PCOS(Y/N)]. The quantiles for women without PCOS differ while compared with women with PCOS; the latter shows a wider range of distribution.



On applying the Logistic Regression Classifier, it resulted in an accuracy of 79.26%.



On applying Random Forest Classifier, we received an accuracy of 90.78%.

5 Conclusion

Polycystic Ovarian Syndrome is a hormonal, genetic, metabolic, and reproductive disorder which affects women and is one of the main reasons for infertility. Since there are no exact reasons as to why PCOS occurs and considering the fact that running tests and procedures are time consuming and energy draining, we have used 11 features including the output label which are easily available and can help us predict.

Both Logistic Regression and Random Forest are applied and the accuracy of each of the models is calculated.

Logistic Regression Accuracy before correlation: 79.26%

Logistic Regression Accuracy after correlation: 90.79%

Random Forest Accuracy before correlation: 90.78%

Random Forest Accuracy after correlation: 89.57%. Clearly, before removing a few features there was a huge difference in the accuracies of the classifiers, and when the new dataset is trained, it is giving almost equal accuracy.

By giving an array of inputs (values of the features), our model is now trained to predict whether a female is undergoing PCOS or not.

References

Aggarwal S, Pandey K (2021) An analysis of PCOS disease prediction model using machine learning classification algorithms. Rec Patents Eng 15(6):53–63

Bhagwani H, Agarwal S, Kodipalli A, Martis RJ (2021, December) Targeting class imbalance problem using GAN. In: 2021 5th international conference on electrical, electronics, communication, computer technologies and optimization techniques (ICEECCOT). IEEE, pp 318–322

Bharati S, Podder P, Mondal MRH (2020, June) Diagnosis of polycystic ovary syndrome using machine learning algorithms. In: 2020 IEEE region 10 symposium (TENSYMP). IEEE, pp 1486–1489

Bhoomika R, Shahane S, Siri TC, Rao T, Ashwini K, Chodon PK (2022) Ensemble learning approaches for detecting Parkinson's disease. In: UPCON

Chauhan P, Patil P, Rane N, Raundale P, Kanakia H (2021, June) Comparative analysis of machine learning algorithms for prediction of pcos. In: 2021 international conference on communication information and computing technology (ICCICT). IEEE, pp 1–7

Deepika V (2019) Applications of artificial intelligence techniques in polycystic ovarian syndrome diagnosis. J Adv Res Technol Manag Sci 1(3):59–63

Denny A, Raj A, Ashok A, Ram CM, George R (2019, October) i-hope: detection and prediction system for polycystic ovary syndrome (PCOS) using machine learning techniques. In: TENCON 2019–2019 IEEE region 10 conference (TENCON). IEEE, pp 673–678

Dhanush N, Prajapati PR, Revanth M, Ramesh R, Kodipalli A, Martis RJ (2021, September) Prediction of gold price using deep learning. In: 2021 IEEE 9th region 10 humanitarian technology conference (R10-HTC). IEEE, pp 1–5

Guha S, Kodipalli A, Rao T (2022) Computational deep learning models for detection of COVID-19 using Chest X-ray images. In: ERCICA

Guha S, Kodipalli A, Rao T (2022) Computational deep learning models for detection of COVID-19 using Chest X-ray images. In: Emerging research in computing, information, communication and applications: proceedings of ERCICA 2022. Springer Nature Singapore, Singapore, pp 291–306

Gupta A, Soni H, Joshi R, Laban RM (2022) Discriminant analysis in contrasting dimensions for polycystic ovary syndrome prognostication. arXiv preprint arXiv:2201.03029

Gururaj V, Ramesh SV, Satheesh S, Kodipalli A, Thimmaraju K (2022) Analysis of deep learning frameworks for object detection in motion. Int J Knowl-Based Intell Eng Syst 26(1):7–16

Hassan MM, Mirza T (2020) Comparative analysis of machine learning algorithms in diagnosis of polycystic ovarian syndrome. Int J Comput Appl 975:8887

Kodipalli A, Devi S (2021) Prediction of PCOS and mental health using fuzzy inference and SVM. Front Public Health 1804

Kodipalli A, Guha S, Dasar S, Ismail T (2022) An inception-ResNet deep learning approach to classify tumours in the ovary as benign and malignant. Exp Syst e13215

Kodipalli A, Devi S, Dasar S, Ismail T (2022) Segmentation and classification of ovarian cancer based on conditional adversarial image to image translation approach. Exp Syst e13193

Krishnaveni V (2019) A roadmap to a clinical prediction model with computational intelligence for pcos. Int J Manage Technol Eng 9(2):177–185

Mehr HD, Polat H (2022) Diagnosis of polycystic ovary syndrome through different machine learning and feature selection techniques. Health Technol 12(1):137–150

Mehrotra P, Chatterjee J, Chakraborty C, Ghoshdastidar B, Ghoshdastidar S (2011, December) Automated screening of polycystic ovary syndrome using machine learning techniques. In: 2011 annual IEEE India conference. IEEE, pp 1–5

Nandipati SC, Ying CX, Wah KK (2020) Polycystic Ovarian Syndrome (PCOS) classification and feature selection by machine learning techniques. Appl Math Comput Intell 9:65–74

Rachana PJ, Kodipalli A, Rao T (2022) Comparison between ResNet 16 and Inception V4 network for covid-19 prediction. In: ERCICA

Rachana PJ, Kodipalli A, Rao T (2022) Comparison between ResNet 16 and Inception V4 network for COVID-19 prediction. In: Emerging research in computing, information, communication and applications: proceedings of ERCICA 2022. Springer Nature Singapore, Singapore, pp 283–290

Rao T, Devamane S, Moumen A (2022) Machine learning approaches for stratification of Parkinson's disease. In: MAASI scientific week (SW22)

Ruchitha PJ, Richitha YS, Kodipalli A, Martis RJ (2021, December) Segmentation of ovarian cancer using active contour and random walker algorithm. In: 2021 5th international conference on electrical, electronics, communication, computer technologies and optimization techniques (ICEECCOT). IEEE, pp 238–241

Ruchitha PJ, Sai RY, Kodipalli A, Martis RJ, Dasar S, Ismail T (2022, October) Comparative analysis of active contour random walker and watershed algorithms in segmentation of ovarian cancer. In: 2022 international conference on distributed computing, VLSI, electrical circuits and robotics (DISCOVER). IEEE, pp 234–238

Sagarnal C, Devamane SB, Hosamani R, Rao T (2021) Deep learning approaches for COVID-19 diagnosis. In: IDDM

Silva IS, Ferreira CN, Costa LBX, Sóter MO, Carvalho LML, de C Albuquerque J, Gomes KB et al (2022) Polycystic ovary syndrome: clinical and laboratory variables related to new phenotypes using machine-learning models. J Endocrinol Invest 45(3):497–505

Soni P, Vashisht S (2018, October) Exploration on polycystic ovarian syndrome and data mining techniques. In: 2018 3rd international conference on communication and electronics systems (ICCES). IEEE, pp 816–820

Subha R, Nayana BR, Radhakrishnan R, Sumalatha P (2022) Computerized diagnosis of polycystic ovary syndrome using machine learning and swarm intelligence techniques

Suha SA, Islam MN (2022) An extended machine learning technique for polycystic ovary syndrome detection using ovary ultrasound image. Sci Rep 12(1):1–16

Thomas N, Kavitha A (2020) Prediction of polycystic ovarian syndrome with clinical dataset using a novel hybrid data mining classification technique. Int J Adv Res Eng Technol (IJARET) 11(11):1872–1881

Tiwari S, Kane L, Koundal D, Jain A, Alhudhaif A, Polat K, Althubiti SA et al (2022) SPOSDS: a smart polycystic ovary syndrome diagnostic system using machine learning. Exp Syst Appl 117592

Vikas B, Anuhya BS, Chilla M, Sarangi S (2018) A critical study of Polycystic Ovarian Syndrome (PCOS) classification techniques. IJCEM Int J Comput Eng Manage 21(4)

Xie NN, Wang FF, Zhou J, Liu C, Qu F (2020) Establishment and analysis of a combined diagnostic model of polycystic ovary syndrome with random forest and artificial neural network. BioMed Res Int

Zacharia S, Kodipalli A (2022) Covid vaccine adverse side-effects prediction with sequence-to-sequence model. In: Emerging research in computing, information, communication and applications: proceedings of ERCICA 2022. Springer Nature Singapore, Singapore, pp 275–281

Zhang X, Liang B, Zhang J, Hao X, Xu X, Chang HM, Tan J et al (2021) Raman spectroscopy of follicular fluid and plasma with machine-learning algorithms for polycystic ovary syndrome screening. Molecular Cellular Endocrinol 523:111139

Classification of Breast Cancer Using Computational Machine Learning Algorithms

P. Gagana, Darshan Aladakatti, Ashwini Kodipalli, Trupthi Rao, and Shoaib Kamal

1 Literature Survey

In the conference paper "Breast Cancer Detection and Diagnosis Using AI" by Khalid Shaikh, we get to understand the depth of AI and its powerful techniques in medicine and healthcare. This paper emphasizes the early detection and diagnosis of breast cancer with its benefits and risks, respectively. The steps taken by Khalid Shaikh are image acquisition, and image processing where the quality of the image is improved for gaining clarity in understanding and interpretation by making use of various techniques. Processing is followed by image segmentation where the entire image is divided into segments to observe and study in detail, then the features of the segmented images are extracted, and based on these features the images of the patients are classified as either normal or abnormal (Shaikh et al. 2021; Feng et al. 2019; Das et al. 2022; Vergis et al. 2021; Vandana and Radhika 2021; Sakib et al. 2022; https://www.webmd.com/cancer/features/top-cancer-killers).

In the paper "Accurate Prediction of Neoadjuvant Chemotherapy Pathological Complete Remission (PCR) for the Four Sub-Types of Breast Cancer", Xin Feng along with his co-ordinates expresses his thoughts on different stages of breast cancer, and its subtype which is detected based on the size of real nodes. Xin Feng talks about the treatment as well which is considered to be the best

P. Gagana · S. Kamal
Electronics and Communication Engineering, Global Academy of Technology, Bangalore, India

D. Aladakatti · S. Kamal
Computer Science and Engineering, Global Academy of Technology, Bangalore, India

A. Kodipalli (✉) · T. Rao · S. Kamal
Artificial Intelligence and Data Science Engineering, Global Academy of Technology, Bangalore, India
e-mail: dr.ashwini.k@gat.ac.in

T. Rao
e-mail: trupthirao@gat.ac.in

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_20

treatment option that is Neoadjuvant chemotherapy also famous known in the medical field as NAC. The results after the prediction were evaluated using performance metrics called Avc. A minimum of 0.7594 Avc was achieved for all four subtypes of breast cancer discussed in (https://analyticsindiamag.com/primer-ensemble-learning-bagging-boosting/; https://www.kaggle.com/code/jagannathrk/predicting-breast-cancer-logistic-regression/data; https://www.geeksforgeeks.org/python-seaborn-pairplot-method/; Rachana et al. 2022a; Guha et al. 2022a; Rao et al. 2022).

In the paper, "Comparison-Based Analysis and Prediction for Earlier Detection of Breast Cancer Using Different Supervised ML Approach" a comparative study was made by Soumen Das and his fellow authors on one of the deadliest diseases, breast cancer. Soumen Das opted to make use of binary classification and chose nine classification models out of which only three of those gave positive variances. Naive Bayes gave a favorable variable as 0.01433, followed by SVM-polynomial that gave 0.01931 and SVM-sigmoid that gave 0.00034 (Sagarnal et al. 2021; Bhoomika et al. 2022; Kodipalli et al. 2022a, 2022b; Ruchitha et al. 2021).

In the paper titled "Decision Support System for Breast Cancer Detection Using Biomarker Indicators" the author Spiridon Vergis develops a mobile application where the patients can enter the required prompted inputs which get tested as new data to the classification algorithms like Logistic Regression popularly known as LR, Naive Bayes, Support Vector Machine familiarly called as SVM, followed by gradient boosting classification. Among these gradient boosting classification provides the best results (Ruchitha et al. 2022; Gururaj et al. 2022; Guha et al. 2022b).

Modern-day treatments are based on the concept of molecular heterogeneity, which is why they can be influenced by various features. Early breast cancer, which is usually confined to the breast, is considered to be a curable disease. This stage is commonly known as benign, which in our dataset is represented as B.

However, when the symptoms are not detected and treated at the early stage the treatment becomes more complicated and cannot to treated using the present therapeutic options and might have an associated risk of spreading to other parts of the body. This condition is called malignant and this is represented as M in our dataset (Rachana et al. 2022b; Zacharia and Kodipalli 2022; Kodipalli and Devi 1804; Bhagwani et al. 2021; Dhanush et al. 2021; Raj et al. 2021).

2 Methodology

We have worked with two algorithms one is the simplest one and another is the most accurate one.

Machine learning has seven main steps such as

(i) Gathering the data
(ii) Preparing data
(iii) Training the model

(iv) Improving the performance

2.1 Gathering the Data

This step includes acquiring the data in the form of a dataset from either an open source or from a medical institute. We have collected our dataset from an open-source website called kaggle (https://www.kaggle.com/code/jagannathrk/predicting-breast-cancer-logistic-regression/data). We selected one such dataset which will classify the given data as either benign or malignant based on 32 features and train on 569 records.

2.2 Preparing Data

The data which we acquire may or maybe be in an ideal form so we need to process it according to our requirements. With our dataset, we first do Exploratory Data Analysis (EDA) look for the presence of null values, and remove them as they will not contribute in any way to classify the tumor as benign or malignant. We then checked for outliers which renders the accuracy, following which we count the number of values for both benign and malignant to make sure we have nearly the same values. If there is a huge difference in the values of these classes then our classification might get biased toward the class with the higher count. We have value counts of benign and malignant as 357 and 212, respectively. Noting that we can say that the difference is minimal and in an acceptable range. Now, we identified the categorical data and replaced them with numerical values for easy and uniform processing by making use of Label encoder. For more reliable and easy understanding we need to scale the values within the range of 0 to 1 which was done using normalization.

To represent our dataset pictorially we plotted a bivariate distribution using a pairplot present in the seaborn library (https://www.geeksforgeeks.org/python-seaborn-pairplot-method/). Pairplot gives us multiple plots of two classes, in which the diagonal plots are univariate. The pairplot was constructed for five columns seen in Fig. 1.

2.3 Training the Model

We have worked on one of the simplest algorithms which are Logistic Regression algorithm, which is tuned using a suitable technique called SVC and the most accurate one Random Forest algorithm which is one bagging technique algorithm and we have also worked on Boosting algorithms like Adaptive boosting algorithm popularly known as Adaboost, Gradient boosting algorithm, and eXtreme gradient boosting algorithm popularly known as XG-boost algorithm. We shall see each one in detail.

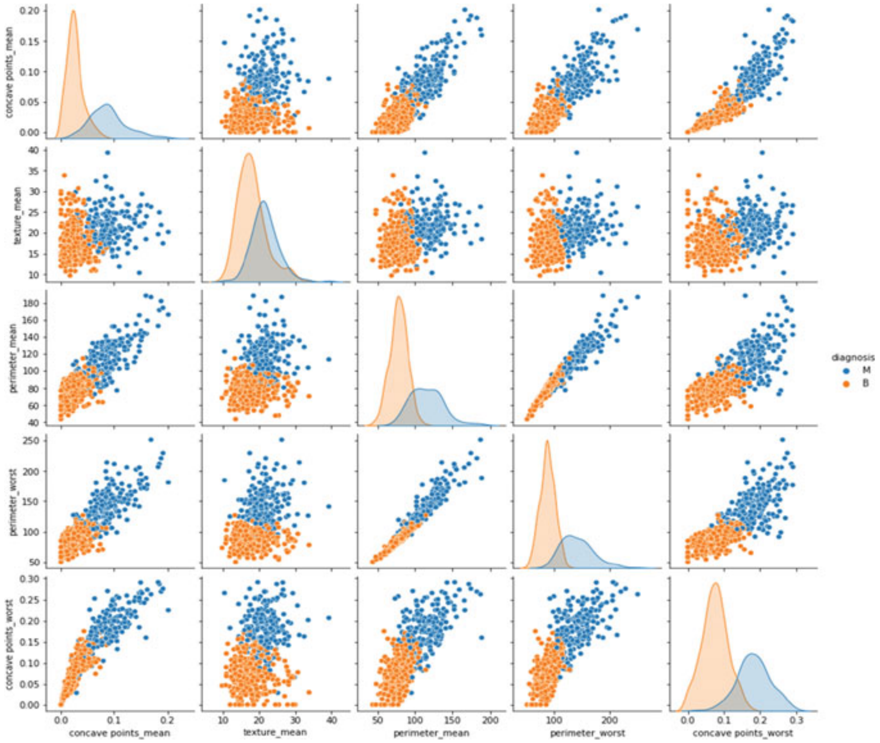


Fig. 1 Pairplot of concave points_mean, texture_mean, perimeter_mean, perimeter_worst, concave points_worst

2.3.1 Logistic Regression

Logistic Regression is a supervised machine learning technique, it is mainly used for classification problems. The value of logistics is a regression between 0 and 1 so it forms the S-like curve, this s-form curve is known as a sigmoid function or the logistic function.

$$\log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

We had two classes to bifurcate, to do the same we used Support Vector Classification popularly known as SVC which is present in Sci-kit learn library of python. Our dataset 1 represents malignant which states that breast cancer can be dangerous and spreads to other parts of the patient's body. Whereas 0 represents benign which states that the cancer is at its early stage and can be treated completely.

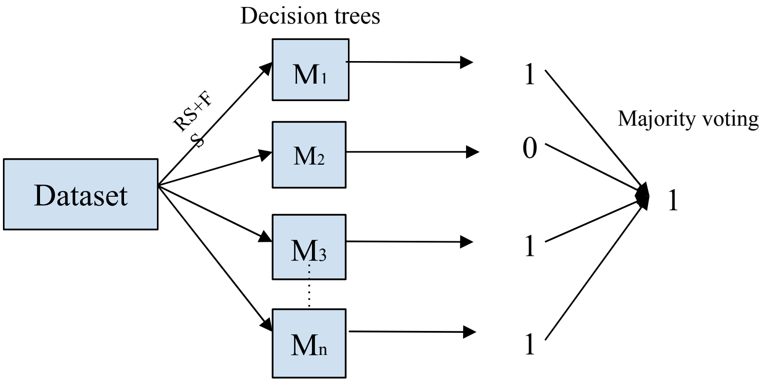


Fig. 2 Representation of bagging technique in Random Forest algorithm

The accuracy obtained from Logistic Regression was 93.57%. We needed to prove the accuracy, for which we used a tuning technique after which our accuracy improved by 4.09% and gave us 97.66%.

2.3.2 Random Forest

Random Forest algorithm is the most widely used algorithm, it is a combination of many decision tree models each of the models will be provided with a sample dataset.

In Fig. 2, we can observe that every decision tree model is provided with a sample dataset, and every new sample dataset is obtained by Row sampling and feature sampling methods. Every decision tree model gives an output and then a voting classifier is applied, the majority of the votes given by the models will be considered.

2.3.3 Adaptive Boosting Algorithm

Ada boosting algorithm combines many weak learners and makes it a strong learner as shown in Fig. 3. Here the base learners are decision trees which are called stomps, for every feature in the dataset sample one stomp will be created. In the first step, every record will be assigned a sample weight, sample weight is calculated using the formula

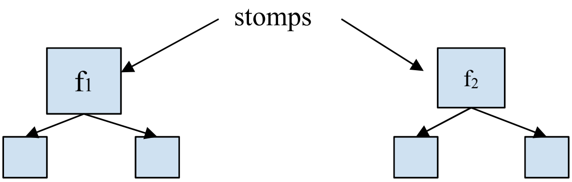
$$w = \frac{1}{n}$$

where n is the number of records

Initially all the records are assigned the same weights.

In step 2 the model with less entropy or Gini value will be selected as the first base learning model, the total error of the misclassified record in stomp is calculated

Fig. 3 Block diagram of weights distribution in adaptive boosting algorithm



as

$$\text{Total error} = \frac{1}{\text{sum of all the weights}}$$

In step 3 the performance of the stomp that is how the model has been classified. To calculate that we use

$$\text{performance of stomp} = \frac{1}{2}\log_e\left(\frac{1 - \text{total error}}{\text{total error}}\right)$$

The wrong classified data is only sent to the next sequential model, the weights are updated to every stomp, the weight is increased for the wrong classified record, and the weight is increased in case of the correct classified record.

To update the weight of an incorrectly classified record:

$$\text{new sample weight} = \text{weight} * e^{\text{performance value}}$$

To update the weight of correctly classified record

$$\text{new sample weight} = \text{weight} * e^{(-\text{performance value})}$$

For the second base learning model the records from the updated normalized weight are selected as the new dataset and again the same process will be done. Whenever the new data is given it is classified based on the majority voting technique.

2.3.4 Gradient Boosting Algorithm

In gradient boosting algorithm ensembling happens by optimizing the loss, hence it is classification problem for building the first tree it considers the log of odds ratio, log of odds ratio can be calculated as:

$$\text{odds ratio} = \frac{\text{probability of success}}{\text{probability of failure}}$$

Then probability will be calculated using:

$$\text{probability} = \frac{e^{\text{odds ratio}}}{1 + e^{\text{odds ratio}}}$$

Extra column will be added to the dataset as predicted value with one of either classes based on the threshold value. Then residuals are calculated for the first base tree with respect to the predicted value column. The second tree will be built on independent values and residuals as the target column. Considering the learning rate specified, the model will again update the odds ratio and the whole will be repeated again.

2.3.5 Extreme Gradient Boosting Algorithm [XG-Boost]

XG-boost is a decision tree-based machine learning algorithm that uses a gradient boosting framebook. As there are two classes namely malignant and benign in the dataset the average probability will be 0.5 and this probability is used to calculate the residuals. The decision tree is constructed as a binary classifier, leaf nodes will always be two. Similarity score will be calculated using:

$$\text{similarity score} = \frac{\sum (\text{residuals})^2}{\sum \text{probability} * (1 - \text{probability}) + \lambda}$$

gain is calculated by:

$$\text{gain} = \text{left similarity} + \text{right similarity} - \text{root similarity}$$

Post pruning method is to build the final decision tree, the post pruning is done based on cover value:

$$\text{cover value} = \sum [p(1 - p)]$$

New probability will be calculated based on a new dataset using the sigmoid function on the base model. This process is continued until the value of residual becomes a very small or maximum number.

(iv) Improving the performance:

To improve the performance of the models we have applied hyperparameter tuning for each and every algorithm based on the observations performance of every model was increased. We have used the GridSearchCV approach for hyperparameter tuning, this approach searches for the best hyperparameters from the grid of given parameter values.

3 Results

Results before tuning the model as shown in Fig. 4 and in Table 1.

Results after tuning the models as shown in Fig. 5 and in Table 2.

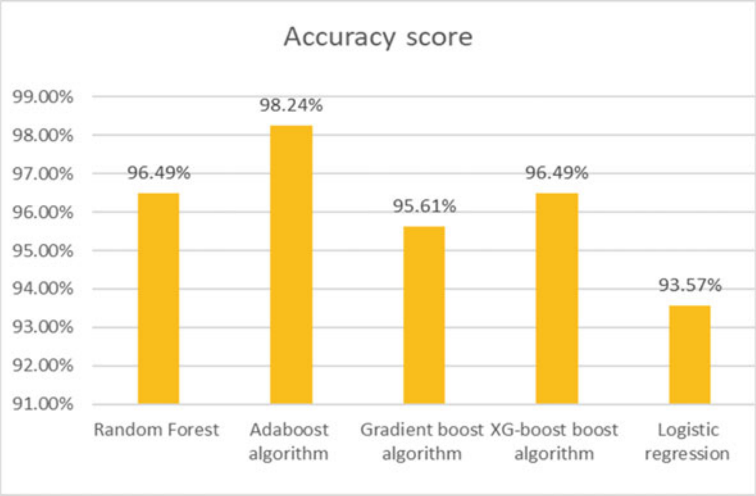


Fig. 4 The plot of the accuracy score of all the algorithms used before tuning

Table 1 The accuracy score of all the algorithms used before tuning

Algorithms	Accuracy scores (%)
Random Forest	96.49
Adaboost algorithm	98.24
Gradient boost algorithm	95.61, 96.49 (with exponential function)
XG-boost boost algorithm	96.49
Logistic regression	93.57

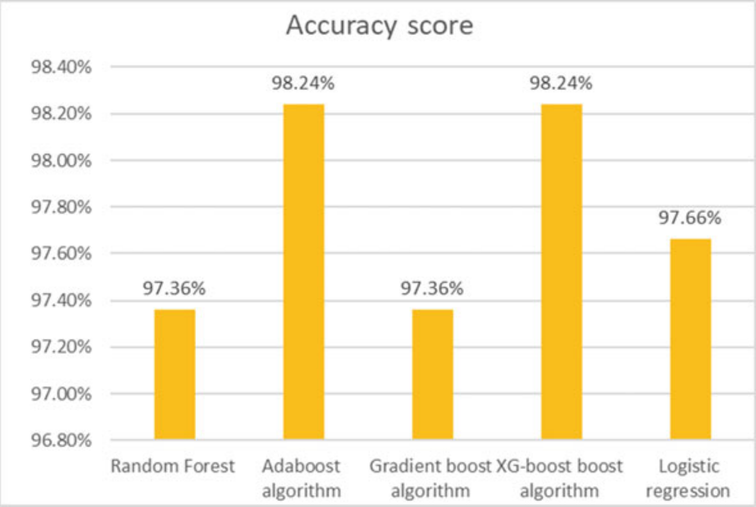


Fig. 5 The plot of accuracy score of all the algorithms used after tuning

Table 2 Accuracy score of all the algorithms used after tuning

Algorithms	Accuracy scores (%)
Random Forest	97.36
Adaboost algorithm	98.24
Gradient boost algorithm	97.36
XG-boost boost algorithm	98.24
Logistic regression	97.66

4 Conclusion

We hereby conclude that the prediction of breast cancer and classification of cancer as either benign or malignant was successful and the highest accuracy was obtained by Adaboost. Other algorithms namely Random forest, Gradient boost, XG-boost, and Logistic Regression improved their accuracy after tuning and gave us comparatively greater results, where Random Forest gave 97.36% which was 0.87% better than the earlier similarly Gradient boost gave 97.36% which was 1.75% better, XG-boost gave 98.24% which had improved by 1.75%, followed by Logistic Regression that gave 97.66% that improved 4.09%. The future work of this paper shall be to build a mobile application and make the correct and efficient medication available to every woman when the required data matches the symptoms of breast cancer. Along with this, the application should have a legend should have a list of hospitals specialized in the field of treating breast cancer.

References

Bhagwani H, Agarwal S, Kodipalli A, Martis RJ (2021, December) Targeting class imbalance problem using GAN. In: 2021 5th international conference on electrical, electronics, communication, computer technologies and optimization techniques (ICEECCOT). IEEE, pp 318–322

Bhoomika R, Shreya Shahane, Siri T C, Trupthi Rao, Dr. Ashwini K, Pradeep Kumar Chodon, "Ensemble Learning Approaches for Detecting Parkinson's Disease", 2022.

Das S, Chatterjee S, Sarkar D, Dutta S (2022) Comparison-based analysis and prediction for earlier detection of breast cancer using different supervised ML approach

Dhanush N, Prajapati PR, Revanth M, Ramesh R, Kodipalli A, Martis RJ (2021, September) Prediction of gold price using deep learning. In: 2021 IEEE 9th region 10 humanitarian technology conference (R10-HTC). IEEE, pp 1–5

Feng X, Song L, Wang S (2019) Accurate prediction of neoadjuvant chemotherapy pathological complete remission (PCR) for the four subtypes of breast cancer

Guha S, Kodipalli A, Rao T (2022) Computational deep learning models for detection of COVID-19 using chest X-ray images

Guha S, Kodipalli A, Rao T (2022) Computational deep learning models for detection of COVID-19 using chest X-ray images. In: Emerging research in computing, information, communication and applications: proceedings of ERCICA 2022. Springer Nature Singapore, Singapore, pp 291–306

Gururaj V, Shriya VR, Ashwini K (2019) Stock market prediction using linear regression and support vector machines. Int J Appl Eng Res 14(8):1931–1934

Gururaj V, Ramesh SV, Satheesh S, Kodipalli A, Thimmaraju K (2022) Analysis of deep learning frameworks for object detection in motion. Int J Knowl-Based Intell Eng Syst 26(1):7–16

https://analyticsindiamag.com/primer-ensemble-learning-bagging-boosting/

https://www.geeksforgeeks.org/python-seaborn-pairplot-method/

https://www.kaggle.com/code/jagannathrk/predicting-breast-cancer-logistic-regression/data

https://www.webmd.com/cancer/features/top-cancer-killers

Kodipalli A, Devi S (2021) Prediction of PCOS and mental health using fuzzy inference and SVM. Front Public Health 1804

Kodipalli A, Guha S, Dasar S, Ismail T (2022) An inception-ResNet deep learning approach to classify tumours in the ovary as benign and malignant. Exp Syst e13215

Kodipalli A, Devi S, Dasar S, Ismail T (2022) Segmentation and classification of ovarian cancer based on conditional adversarial image to image translation approach. Exp Syst e13193

Rachana PJ, Kodipalli A, Rao T (2022) Comparison between ResNet 16 and Inception V4 network for Covid-19 prediction

Rachana PJ, Kodipalli A, Rao T (2022) Comparison between ResNet 16 and Inception V4 network for COVID-19 prediction. In: Emerging research in computing, information, communication and applications: proceedings of ERCICA 2022. Springer Nature Singapore, Singapore, pp 283–290

Raj A, Umrani NR, Shilpashree GR, Audichya S, Kodipalli A, Martis RJ (2021, July) Forecast of covid-19 using deep learning. In: 2021 IEEE international conference on electronics, computing and communication technologies (CONECCT). IEEE, pp 1–5

Rao T, Devamane S, Moumen A (2022) Machine learning approaches for stratification of Parkinson's disease

Ruchitha PJ, Richitha YS, Kodipalli A, Martis RJ (2021, December) Segmentation of ovarian cancer using active contour and random walker algorithm. In: 2021 5th international conference on electrical, electronics, communication, computer technologies and optimization techniques (ICEECCOT). IEEE, pp 238–241

Ruchitha PJ, Sai RY, Kodipalli A, Martis RJ, Dasar S, Ismail T (2022, October) Comparative analysis of active contour random walker and watershed algorithms in segmentation of ovarian cancer. In: 2022 international conference on distributed computing, VLSI, electrical circuits and robotics (DISCOVER). IEEE, pp 234–238

Sagarnal C, Devamane SB, Hosamani R, Rao T (2021) Deep learning approaches for COVID-19 diagnosis

Sakib S, Yasmin N, Tanzeem AK, Shorna F, Hasib KMd, Alam SB (2022) Breast cancer detection and classification: a comparative analysis using machine learning algorithms

Sanjana S, Sanjana S, Shriya VR, Vaishnavi G, Ashwini K (2021) A review on various methodologies used for vehicle classification, helmet detection and number plate recognition. Evol Intel 14(2):979–987

Shaikh K, Krishnan S, Thanki R (2021) Artificial intelligence in breast cancer early detection and diagnosis. Exclusive license to Springer Nature Switzerland AG

Vandana L, Radhika K (2021) Detailed review on breast cancer diagnosis using different ML algorithms

Vergis S, Bezas K, Exarchos TP (2021) Decision support system for breast cancer detection using biomarker indicators

Zacharia S, Kodipalli A (2022) Covid vaccine adverse side-effects prediction with sequence-to-sequence model. In: Emerging research in computing, information, communication and applications: proceedings of ERCICA 2022. Springer Nature Singapore, Singapore, pp 275–281

Credit Card Fraud Analysis Using Machine Learning

Sree Charitha, Shivani Chowdary, Trupthi Rao, Ashwini Kodipalli, Shoaib Kamal, and B. R. Rohini

1 Introduction

Online payments and digital currencies have emerged as a result of rapid technological advancement and widespread use (Dornadula and Geetha 2019; Maniraj 2019; Kodipalli et al. 2022a). The most popular form of credit card used in everyday activities is a digital credit card offered by banks with a pre-determined credit limit. The credit limit is set by the card issuer relying on the recipient's credit score and income. In spite of the benefits of cashless transfer, credit card firms have seen a rise in card fraud as a result of the emergence of numerous new technologies. Scammers must be prevented from using your card information for immoral purposes and initiating transactions from your account without your consent. Scams can occur in a variety of methods, including phishing, dumpster diving, hacking, and skimming (Ruchitha et al. 2021; Kodipalli et al. 2022b). When a person is at an automated teller machine (ATM) or point of sale (POS), the PIN numbers are stolen in the common fraud known as "skimming". Card skimmers are successful as there is a lack of public awareness of the problem; thus, we must be attentive and vigilant to protect oneself from these incidences. We make sure that transactions are carried out through reputable and approved portals to avoid falling victim to another fraud called phishing, which is individually requesting information and details from the user via insurance agents

S. Charitha · S. Chowdary · T. Rao · A. Kodipalli (✉) · S. Kamal · B. R. Rohini
Department of Artificial Intelligence and Data Science, Global Academy of Technology, Bangalore, India
e-mail: dr.ashwini.k@gat.ac.in

T. Rao
e-mail: trupthirao@gat.ac.in

S. Kamal
e-mail: shoaib.kamal@gat.ac.in

B. R. Rohini
e-mail: rohini.br@gat.ac.in

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_21

or other marketers. Dumpster diving is a peculiar situation where fraudsters search through trash records for valuable information that people do not take carelessly and throw away, making a way to these divers and making them a jackpot. Accordingly, we must guarantee that all junk information has been destroyed. According to a new analysis, credit card fraud will result in significant global losses over the next 5 years and will greatly increase consumer anxiety. It is not that we cannot stop these frauds; we can only try to prevent them by tracking down the transactions using a variety of methods. Machine learning, fuzzy logic, and deep learning approaches can be used. In this study, machine learning methods have been used to examine all the approved transactions and flag any suspect ones. These techniques have emerged as a savior to safeguard all scams taking place around the globe and help deter perpetrators from these actions. Professionals that work with cardholders examine these reports to determine whether the transactions are legitimate or fraudulent and monitor them using the users' data. In order to prevent criminals from modifying their fraudulent tactics, fraud detection techniques are constantly being developed. Account takeover, social engineering fraud, application fraud, and counterfeit cards are just a few of the crimes that often arise (Ruchitha et al. 2022; Gururaj et al. 2022). To tackle this situation, we have several machine learning algorithms, including logistic regression (LR), random forest (RF), decision tree (DT), support vector machine (SVM), Naive Bayes (NB), and K-nearest neighbor (KNN) to analyze the transactions. The automated system receives a report from the investigators which is then used to program the working and make improvements, thereby enhancing this scam detection.

2 Literature Review

Fraud is defined as intentional breach of the law or illegal deception for financial or personal gain by unauthorized users. The Reserve Bank of India, which is in charge at the national level, consistently issues instructions to protect banks' and consumers' interests. Numerous pieces of literature relating to anomaly or fraud detection in this field have previously been published and are available to the general audience (Guha et al. 2022; Kodipalli and Devi 2021; Rachana et al. 2022; Zacharia and Kodipalli 2022).

Throughout many areas of the country, there were numerous attacks from the 2000s to the present that resulted in losses of enormous amounts, such billions of dollars. To deal with these skimmers, numerous rules and sanctions have been put into place. These frauds have increased in frequency in India since 2010. Section 66C, Section 468, and Section 471 are a few of the laws that are in effect in India and state that anyone discovered to have engaged in these frauds will be held accountable under these laws (Bhagwani et al. 2021; Dhanush et al. 2021; Sanjana et al. 2021). As per a survey done in 2013, 40% of all financial crime in 2013.

Even if there are numerous rules and regulations in place, there are still many examples occurring in various locations. To address this challenge, technology has advanced, which we will explain in the next sections of this article (Gururaj et al. 2019).

Genetic algorithms, clustering techniques, and neural networks are a few further ways that identify suspicious transactions through prediction methods based on a set of rules and logic. But in this study, we attempted to use a few machine learning techniques to effectively attack credit cards. Improvements to the feedback system have been made, and in the event of a fraudulent transaction, feedback would be sent to alert the authorized system and help stop fraud. In this way, we may make every effort to ensure that society is flourishing and free of any criminal acts, which can be stopped by applying the abovementioned methods to actual-world situation.

3 Problem Statement

The challenge is defined as tracing past transactions made by credit card users during regular usage to recognize the type of transaction and classify it as legitimate or fraudulent. The next step is to eliminate improper fraud categorization and attempting to identify every fraudulent transaction possible. Despite being complicated, it can be handled appropriately. The ultimate objective is to identify the transaction and prevent daily scams as much as feasible.

4 Proposed Work

The principal goal of this paper is to raise people's understanding of how to spot credit card fraud and real-world online credit card scam. The overall purpose of this detection system is to ensure the privacy and security of our transactions so that nothing goes wrong. With this system, fraudsters are prohibited from using a stolen or fake card to make a number of purchases before the cardholder becomes aware of the deception. Then, a new transaction is examined using this method to assess whether it is fraudulent or not. We can accomplish this using a variety of ways, but we primarily use ML algorithms. In this scenario, finding all fraudulent transactions and handling this undesirable fraud categories is our objective.

The system-level architecture is presented in the diagram below (Kodipalli et al. 2022a) (Fig. 1).

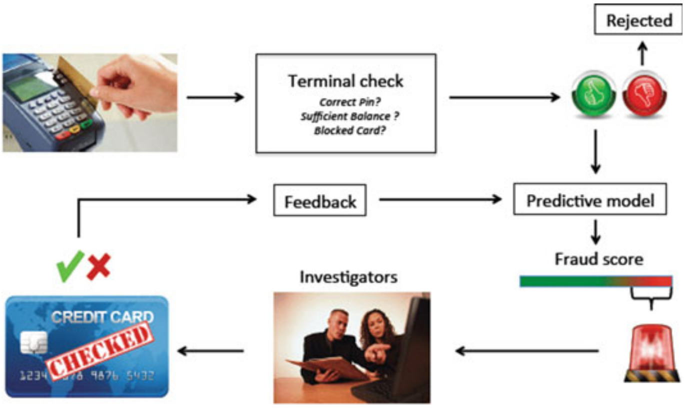


Fig. 1 System-level architecture (Kodipalli et al. 2022a)

5 Methodology

Although many algorithms and procedures are available, we have only implemented and applied a small number of them, namely logistic regression, decision tree, random forest, and sampling methods. These can be used to see which one fits better and can be used by credit card merchants to detect any of the fraudulent transactions.

Algorithm steps include:

Step 1: Read and load the dataset.
Step 2: Data exploration is performed.
Step 3: Partition the dataset into train dataset and test (validation) dataset.
Step 4: Feature selection is put forward for the recommended models.
Step 5: The effectiveness of several algorithms has been assessed using accuracy measures.
Step6: The best algorithm for the provided dataset will then be presented here.

The dataset before applying any algorithm is shown here (Fig. 2).

5.1 Dataset Description

When compared to the customer’s prior transactions, card transactions are always peculiar, and these need to be dealt with utmost care. When these issues arise in the real world, they are considered to be conceptual drift concerns. These variables factor a high discrepancy in data. The major purpose of this paper is to address the challenge of concept drift in order to apply it to practical concept (Dornadula and Geetha 2019) (Table 1).

Fig. 2 Count plot of labels

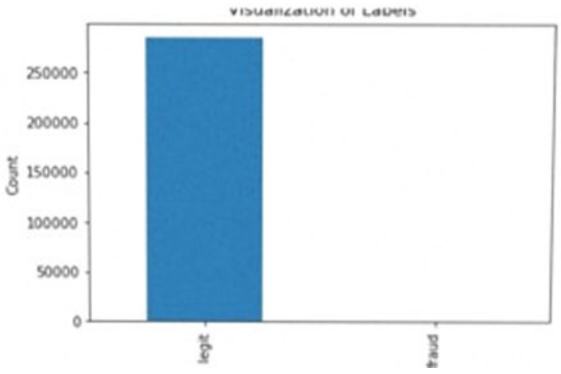


Table 1 Feature description (Maniraj 2019)

Attribute name	Description
Transaction id	Identification number of a transaction
Cardholder id	Unique identification number given to the cardholder
Amount	Amount transferred or credited in a particular transaction by the customer
Time	Details like time and date, to identify when the transaction was made
Label	To specify whether the transaction is genuine or fraudulent 0. Not fraud 1. Fraud

The table above displays core attributes that are collected when a transaction is made. There are a total of 284,807 transactions in the dataset of which 492 or 0.172% are fraudulent and the remaining transactions are legit (Dornadula and Geetha 2019). The dataset is highly unbalanced because of this circumstance. Therefore, principal component analysis (PCA) is used in order to modify the majority of the dataset features because disclosing customers' transactional information could result in a breach of their confidentiality (Zacharia and Kodipalli 2022). Time, quantity, and class are non-PCA applied aspects, while V1, V2, V3, …, and V28 are PCA applied aspects.

5.2 *Classifiers*

Logistic Regression

It is of the type supervised ML techniques. It can be applied for both regression and classification tasks, but most typically employed for classification tasks. Logistic regression is intended to forecast categorical variables by using the dependent variables. Two classes will be assessed, and then a new data point will be examined to

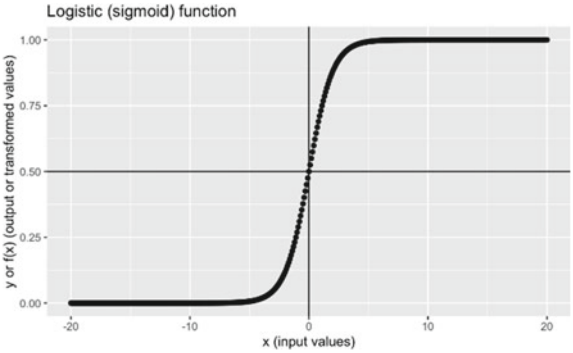


Fig. 3 Sigmoid curve

determine which class it belongs to and that will be regarded as predicted output. The algorithms give the probabilistic values that ranges from (0) to (1) (Fig. 3).

The sigmoid function or the logistic function, which is employed by logistic regression, is a more composite cost function. LR is a less rigid technique for statistical analysis compared to others. Logistic regression was ultimately deployed to anticipate the probability of theft of credit cards.

Decision Tree Algorithm

Decision tree belongs to the class of supervised learning technique that is extensively applied to classification tasks and regression tasks. It performs with continuous and categorical input and output variables. It is tree-structured classifier which starts with the root node and expands on further branches. In this methodology, the most relevant separator or differentiator among the input variables is utilized to divide the population or sample into two or more equivalent categories. Leaf nodes represent the outcome of decisions and do not have any extra branches, whereas decision nodes are used to make appropriate decisions and include numerous branches. To separate a node into two or more sub-nodes, they adopt a range of approaches. To look at it another way, we can state that the node's intelligibility improves in relation to the target variable (Kodipalli and Devi 2021). The nodes are divided using all relevant variables by a decision tree, which then determines the division that generates the most homogeneous sub-nodes. We can determine whether a transaction is fraudulent or not by identifying the most appropriate node. Gini index and information gain techniques are commonly utilized in the use of decision tree.

Random Forest

It is supervised ML algorithm which employs the idea of ensemble learning and used to solve both classification and regression tasks. It involves tree-based method that entails building numerous trees and modifying the results to increase the applicability of the model. The above approach is referred as ensemble approach. To develop a strong learner, only weak learners (individual trees) are integrated. The output variable is fixed in regression situations, whereas it is categorical in classification. To create random samples, the bagging Algorithm is employed. Bagging aggregates predictions from all models after fitting several models on various subsets of the training dataset. An implementation of the bagging technique called random forest chooses a selection of features at random from the dataset. In contrast to a decision tree, individual tree in a random forest is allowed to grow to its true capacity. In the final stage, the classifier algorithm component allows us to determine the transaction type of credit card.

Resampling Methods:

Resampling approach refers to a group of statistical analytical techniques which include generating a sample from training data and then refitting it to a model. It is the most effective way for handling an uneven dataset. The major types of resampling are as follows:

1. ***Under-sampling***: The majority class eliminates instances in the under-sampling method until the minority class is balanced. The drawback is that a model loses knowledge that would otherwise be valuable. The under-sampling method performed here is random under-sampling. Random under-sampling entails adding replacement and randomly choosing samples from the minority class to the training dataset until a more balanced proportion is attained (Gururaj et al. 2022). The illustration of random under-sampling is shown above.
2. ***Over-sampling***: It works by updating and enhancing the training data with duplicate examples from minority classes, and it leads to an overfitting of the model. There are a number techniques it uses, some of which include SMOTE and random over-sampling (Fig. 4).

SMOTE is abbreviated as Synthetic Minority Over-sampling TEchnique used for synthesizing new samples. The procedure involves identifying samples with similar feature spaces, drawing a line between them, and then establishing a new sample

Fig. 4 Under-sampling

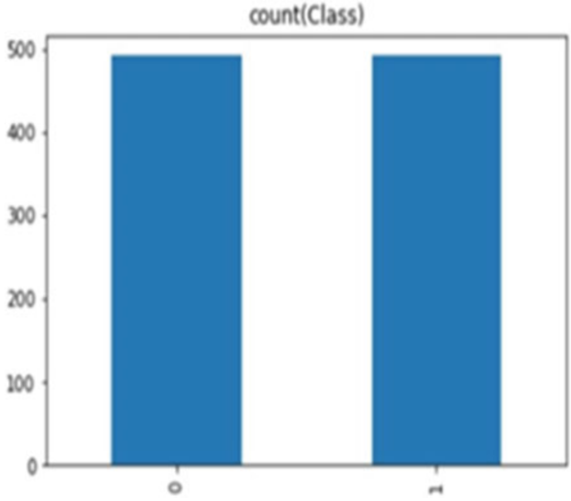
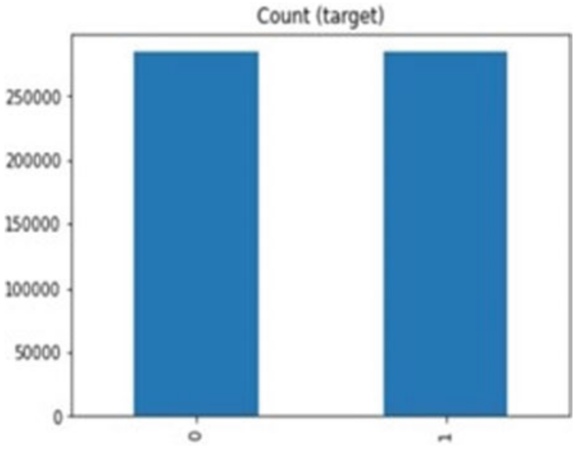


Fig. 5 Over-sampling



point along that line. This approach works well because new samples are reasonably closer to minority class in feature space.

Random over-sampling: It is a sampling algorithm which enables us to randomly choose a subset of samples from a population without including all possible population samples and then forecasting the outcome variable (Gururaj et al. 2022). The illustration of over-sampling method is given above (Fig. 5).

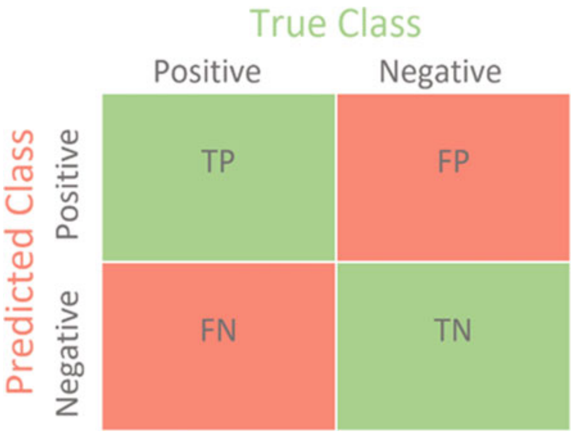
5.3 Related Work

Although we have several ML approaches at our disposal to address this issue, we tried to implement decision tree, random forest, logistic regression, and resampling algorithms in order to create a system for detecting credit card fraud based on this European dataset, which has a high variation between legitimate and fraudulent transactions, yielding a very unstable and unbalanced environment. Evaluating using each ML approach is performed using the classification accuracy.

The outcomes of the study revealed that the for logistic regression with over-sampling and under-sampling, decision tree, and random forest, the accuracy scores are as follows 99.9, 93.93, 99.10, and 99.40%, correspondingly. Despite the promising results, the experts advise using more extensive preprocessing strategies as they could boost the performance of the classifiers.

Problems with class imbalance are now increasingly prevalent, and employing sampling techniques to solve them is a huge relief. Here, we employed SMOTE, random over-sampling, and random under-sampling as sampling techniques.

Fig. 6 Confusion matrix



6 Performance Metrics

6.1 Confusion Matrix

Confusion matrix is categorized as the performance indicator for a machine learning (ML) classification issue where the output comprises two or more classes. It is a two-dimensional table with four possible predicted and actual value combinations which include the following cases (Fig. 6):

(a) **True Positive (TP)**: This illustrates the scenario in which the actual and expected classes of data items are both 1 (positive).
(b) **True Negative (TN)**: In this case, both the actual and expected class of data points are 0 (negative).
(c) **False Positive (FP)**: This is the case when a data point’s expected class is 1, but its actual class is 0.
(d) **False Negative (FN)**: This is the case when a data point’s expected class is 0, but its actual class is 1.

The following are performance measurements:

Accuracy: It is described as the proportion of correctly predicted outcomes to all forecasts.
Precision: It is described as the number of accurate samples the ML model returned.
Recall: It is represented as the quantity of positive results the ML model returned.
F1 Score: This score is characterized as being the harmonic mean of recall and precision.

7 Results

The outcomes of the specified algorithm are listed below using logistic regression, decision tree, random forest; the experimental outcomes showed following accuracy scores 97.3% (without sampling) and 99.10%, 99.40%, respectively. Despite the promising outcomes, the experts suggested that using sophisticated preprocessing techniques may guarantee the effectiveness of the classifiers. In order to address the issue of class imbalance problem in the dataset, we used resampling techniques like SMOTE, random over-sampling, and random under-sampling. As a result, we were able to achieve accuracy levels of 99.9%, 93.9%, while SMOTE provides the highest accuracy of all.

8 Conclusion

In this paper, we tried to initiate a revolutionary procedure for fraud detection, where customers are assembled based on their transactions and obtain observable patterns to evolve a profile of every customer. Next, we employed various classifiers to achieve the highest level of accuracy. A different approach was implemented, which enabled the system to adjust to new cardholders' transactions in a timely manner for each client, resolving the problem of concept drift. We explored balancing the dataset by using SMOTE and random over-sampling in LR, and we observed that the classifiers were performing better than other algorithms. We even implemented using other algorithms like logistic regression, decision tree, and random forest. There are a few fraud detection techniques available today, but none of them is capable of accurately identifying every fraud at every stage. With the help of this strategy, credit card theft may be identified right away. We can protect our information and stop the loss of money by detecting frauds early on by training the machine with data from prior transactions.

References

Bhagwani H, Agarwal S, Kodipalli A, Martis RJ (2021) Targeting class imbalance problem using GAN. In 2021 5th international conference on electrical, electronics, communication, computer technologies and optimization techniques (ICEECCOT). IEEE, pp 318–322.

Dhanush N, Prajapati PR, Revanth M, Ramesh R, Kodipalli A, Martis RJ (2021) Prediction of gold price using deep learning. In: 2021 IEEE 9th region 10 humanitarian technology conference (R10-HTC). IEEE, pp 1–5

Dornadula VN, Geetha S (2019) Credit card fraud detection using machine learning algorithms. In: International conference on recent trends in advanced computing (ICRTAC)

Guha S, Kodipalli A, Rao T (2022) Computational deep learning models for detection of COVID-19 using chest X-ray images. In: Emerging research in computing, information, communication and applications: proceedings of ERCICA 2022. Springer Nature Singapore, Singapore, pp 291–306

Gururaj V, Shriya VR, Ashwini K (2019) Stock market prediction using linear regression and support vector machines. Int J Appl Eng Res 14(8):1931–1934

Gururaj V, Ramesh SV, Satheesh S, Kodipalli A, Thimmaraju K (2022) Analysis of deep learning frameworks for object detection in motion. Int J Knowl Based Intell Eng Syst 26(1):7–16

Kodipalli A, Devi S (2021) Prediction of PCOS and mental health using fuzzy inference and SVM. Front Public Health 1804

Kodipalli A, Guha S, Dasar S, Ismail T (2022a) An inception—ResNet deep learning approach to classify tumours in the ovary as benign and malignant. Expert Syst e13215

Kodipalli A, Devi S, Dasar S, Ismail T (2022b) Segmentation and classification of ovarian cancer based on conditional adversarial image to image translation approach. Expert Syst e13193

Maniraj SP, Saini A, Sarkar SD, Ahmed S (2019) Credit card fraud detection using machine learning and data science. Int J Eng Res Technol 8(09). ISSN: 2278-0181

Rachana PJ, Kodipalli A, Rao T (2022) Comparison between ResNet 16 and inception V4 network for COVID-19 prediction. In: Emerging research in computing, information, communication and applications: proceedings of ERCICA 2022 (pp. 283–290). Singapore: Springer Nature Singapore.

Ruchitha PJ, Richitha YS, Kodipalli A, Martis RJ (2021) Segmentation of ovarian cancer using active contour and random walker algorithm. In: 2021 5th international conference on electrical, electronics, communication, computer technologies and optimization techniques (ICEECCOT). IEEE, pp 238–241

Ruchitha PJ, Sai RY, Kodipalli A, Martis RJ, Dasar S, Ismail T (2022) Comparative analysis of active contour random walker and watershed algorithms in segmentation of ovarian cancer. In: 2022 international conference on distributed computing, VLSI, electrical circuits and robotics (DISCOVER). IEEE, pp 234–238

Sanjana S, Sanjana S, Shriya VR, Vaishnavi G, Ashwini K (2021) A review on various methodologies used for vehicle classification, helmet detection and number plate recognition. Evol Intel 14(2):979–987

Zacharia S, Kodipalli A (2022) Covid vaccine adverse side-effects prediction with sequence-to-sequence model. In: Emerging research in computing, information, communication and applications: proceedings of ERCICA 2022. Springer Nature Singapore, Singapore, pp 275–281

Machine Learning Approaches for Stroke Detection and SMOTE for Imbalanced Data

H. K. Ruchitha, B. S. Sanjana Sharma, Sneha, Trupthi Rao, Ashwini Kodipalli, and Shridhar B. Devamane

1 Introduction

The second biggest cause of death worldwide is stroke and the leading cause of severe disability in the adult population. Therefore, it is essential to accurately predict stroke so that it can be treated early. Reduced brain blood flow is a medical ailment known as a stroke. By which the cells in those areas begin to starve of nutrients and oxygen and die. According to the World Stroke Organization, one in four adults over the age of 25 will experience a stroke at some point in their lives. The first stroke will occur in 12.2 million people globally this year, and 6.5 million will pass away (Yagin et al. 2021; Chen et al. 2021; Emon et al. 2021). More than 110 million of her people worldwide have had a stroke. Ischemic and haemorrhagic strokes are the two types. In haemorrhagic stroke, which accounts for 87% of all strokes, weak blood arteries burst, and haemorrhage into the brain, while blood clots impede outflow in ischemic strokes.

A stroke is a condition that results in motor or sensory impairment in this area of the brain as a result of a blockage or fracture of the brain's vasculature, according to the World Health Organization. Characterized by the clinical manifestations of imbalance, visual disturbances, speech, and cognitive impairment up to coma. Stroke

H. K. Ruchitha · B. S. S. Sharma · Sneha · T. Rao · A. Kodipalli (✉)
Department of Artificial Intelligence and Data Science, Global Academy of Technology, Bangalore, Karnataka, India
e-mail: dr.ashwini.k@gat.ac.in

T. Rao
e-mail: trupthirao@gat.ac.in

S. B. Devamane
Department of Computer Science and Engineering, Global Academy of Technology, Bangalore, Karnataka, India
e-mail: shridhar.devamane@gat.ac.in

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_22

is referred to by the World Health Organization as the coming epidemic of the twenty-first century. Stroke, which plays a key role in hospitalization, can be lethal, cause a range of physical impairments requiring more frequent rehabilitation and care, and lead to personal, social, and economic problems (Hung et al. 2017; Alotaibi and Sasi 2016).

High blood pressure is believed to be the main reason of stroke. Key factors of stroke include high cholesterol, smoking, being obese, and having diabetes. Some of the indications of a stroke are unexpected facial numbness or weakness, arms, or legs (in particular on one side of the body). Suddenly having trouble in speaking or understanding the language. Sudden vision impairment in either one or both eyes. Sudden dizziness, balance issues, or issues with coordination, as well as trouble walking. Severe headache with no apparent cause (Shoily et al. 2019).

It is critical to identify the generality and trouble for diseases such as stroke, which have high incidence of disability and death rate and take appropriate ways to cut back mortality. It is necessary to predict stroke and treat it to prevent permanent damage or death. The machine learning methods are used is currently in high dictate to proactively foretell the patient's state and to avoid the risk of stroke. Examination of affected individuals has uncovered many risk elements thought to be associated with the cause of stroke. The clinical data can then be evaluated and meaningful information obtained. Minimizing costs helps us make better decisions and improve the healthcare sector.

Medical data is digitized, which helps create predictive models for early risk prevention. Without making rigid statistical assumptions, machine learning (ML) approaches are a potent settle of methods that can predict the intricate and unseen interactions between a wide variety of impersonal factors and desired dispassionate outcomes from data. With the expansion of the IT sector, ML algorithms are applied gradually to the realm of medicine. Create disease prediction models using ML algorithms to predict risk of illness, death, and readmission. ML algorithms include Logistic Regression, Decision Trees, and K-Nearest Neighbours (KNN) (Ray et al. 2020; Rachana et al. 2022a; Guha et al. 2022a). With its powerful capacity for thought and guide hard information, ML can handle intricate interactions and nonlinear liaison between variables, making a reliable and effective system predictive models. Provides new methods and techniques for clinical outcome prediction.

Based on an individual's physical state and data from medical reports, three machine learning algorithms were utilized in this study to determine the occurrence of a stroke that may or may not have occurred. In our opinion, machine learning techniques can be utilitarian for better understand the disease and become a better partner in healthcare. Adopting a healthy, balanced lifestyle can help prevent stroke that eliminates bad habits such as regulating body mass index (BMI), typical blood sugar levels, and smoking and drinking and maintaining heart and kidney health.

2 Literature Review

There are a lot of researchers researching in the field of ML related to stroke.

In paper Rao (2022) and Sagarnal and Devamane (2021), they focused on the important sections of the investigated imbalanced data, figured out how knowledge discovery should be done in those situations, and made recommendations for the stages of the learning process that were laid out. They used the Borderline-SMOTE method and Gradient Boosting Tree method giving an accuracy of 0.9339 and 0.9339, respectively.

In Bhoomika et al. (2022) and Kodipalli et al. (2022a), they worked and summarized that more attention should be paid to factors that have a strong influence correlation with Pharmaceutical Supply Chain Initiative (PSCI) in clinical practice, and also the types of traditional Chinese Medicine (TCM) syndrome, TCM predisposition, and PSCI were discussed. Random Forest (RF), Support Vector Machine (SVM), and Gradient Boosting (GDBT) algorithms were used and GDBT gave the highest accuracy of 80.77%.

In Ruchitha et al. (2021) and Kodipalli et al. (2022b), they worked on the relationship between different attributes of stroke and the likelihood of stroke occurring in a human was assessed. They worked on 11 different algorithms: Stochastic Gradient Descent (SGD)—65%, AdaBoost—94%, Gaussian—78%, Quadratic Discriminant Analysis (QDA)—79%, Multi-layer Perceptron (MLP)—79%, K-Neighbours—87%, Logistic Regression (LR)—78%, Gradient Boosting Classifier (GBC)—965, Decision Tree Classifier (DTC)—91%, XGBoost (XGB)—96%, Weighted Voting—97% accuracy was obtained, respectively. In comparison with other machine learning algorithms that are utilized, Weighted Voting has an anticipated maximum accuracy of roughly 97%, according to the performance evaluation.

In paper Ruchitha et al. (2022) and Gururaj et al. (2022), they worked to build an innovative method for creating an automated system for predicting the incidence of strokes that may have various benefits, having predictable outcomes, high forecast accuracy, and prompt forecast reporting. They used Deep Neural Network (DNN), Gradient Boosting Decision Tree (GDBT), Logistic Regression (LR), and Support Vector Machine (SVM) with accuracy of 88.3%, 86.8%, 86.6%, and 83.9%, respectively.

In Guha et al. (2022b) and Rachana et al. (2022b), they worked on developing new models to predict the needs to suggest that a stroke patient may or may be admitted to ICU and as an inpatient using ensemble approaches to the sampled dataset. Algorithms used were Decision Tree (DT), Artificial Neural Network (ANN), Support Vector Machine (SVM), and Logistic Regression (LR) with accuracy 71%, 65%, 74%, and 72%, respectively. DT model appeared to be providing higher accuracy.

In Zacharia and Kodipalli (2022) and Kodipalli and Devi (2021), authors worked on four classifier algorithms: Naive Bayes, J48, K-Nearest Neighbours (KNN), and Random Forest (RF) with accuracy 85.6%, 99.85%, 99.85, and 99.8%, respectively.

In paper Bhagwani et al. (2021) and Dhanush et al. (2021), authors worked on two different decision-making models, the Two-Class Boosted Decision Tree and the

Two-Class Decision Jungle, with accuracy rates of 97.6% and 96.7%, respectively (Sanjana et al. 2021; Gururaj et al. 2019). In this situation, the Two-Class Boosted Decision Tree excels and provides greater accuracy.

3 Research Methodology

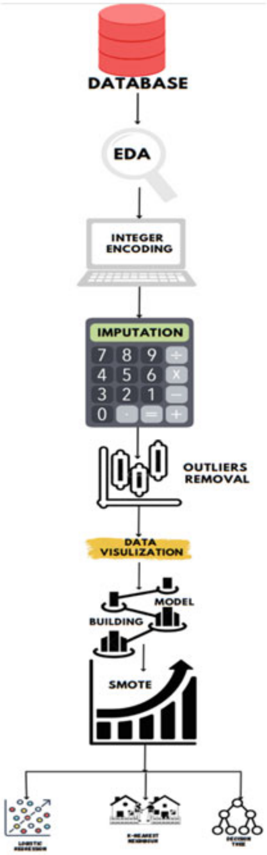
The suggested approach used in this study for the analysis of stroke is:

3.1 Data Description

A dataset collecting information on individuals with and without stroke was taken in this analysis. It is a record containing the data of 5110 individuals. 1 dependent and 10 independent variables are present in this dataset. The dependent variable is stroke. Independent variables are age, heart_diease, ever_married, avg_glucose_level, residence_type, gender, work_type, smoking_status, hypertension, and BMI. The dependent variable represents the clinical outcome indicating whether the patient has a stroke or not.

All features and now described:

- AGE: This characteristic refers to the age of the person. It is a numeric data.
- GENDER: This characteristic indicates the gender of the individual. It is categorical data.
- HYPERTENSION: The presence or absence of high blood pressure is indicated by this characteristic. They are numerical data.
- WORK_TYPE: This attribute describes a person's working environment. Data is categorical.
- RESIDENCE_TYPE: This characteristic describes the living situation of the individual. Data with categories.
- HEART_DISEASE: This characteristic indicates whether or not the person has heart disease. They are numerical data.
- AVG_GLUCOSE_LEVEL: A person's level of glucose status was indicated by this feature. They are data in digits.
- BMI: This characteristic represents a person's body mass index. This data is numerical.
- EVER_MARRIED: This property reflects one's marriage. Data that is categorical.
- SMOKING_STATUS: This characteristic indicates whether the person smokes. Data that is categorized.
- STROKE: This characteristic indicates whether a person has ever had a stroke. Data is in numerical form.



3.2 EDA

EDA is an acronym that stands for exploratory data analysis. Exploratory data analysis is the early study of data to detect patterns, test hypotheses, identify anomalies, and confirm assumptions using abstract statistics and graphs. It helps to get a better understanding of the dataset. Helps detect outliers and unusual events. It helps you understand the dataset variables and the relationship between them.

- .shape: The shape yields a tuple containing the number of corresponding items for each index. The dataset consists of 5110 rows and 12 columns.
- .size: An array's total number of components equals the size of the dataset. The dataset has a size of 61,320.

- .info(): The info() function is used to prints the data frame's information. The number of columns in total, data types, their labels, memory utilized, range index, and each column's number of cells (nonzero values) are all included in the data.
- .describe(): The describe method generates various parameters describing each feature of the dataset like a count, an average, a standard deviation, a maximum, a minimum, the 25th, 50th, and 75th percentiles of the dataset.
- .head(): The head generated the first five rows of the dataset by default.
- .dtypes: The dtypes returns the data type of each of the attributes in the dataset.
- .isnull().values.any(): Returns true when entire dataset has missing values and false when there are no missing values from the dataset.
- .isnull().sum(): This function returns the total number of missing data for each of the attributes of the dataset.

3.3 Integer Encoding

It involves replacing categorical values with numerical values from $1 - n$ to 0 to $n - 1$, where n represents how many different categories are there for each dataset attribute.

3.4 Imputation

Imputation is the process of substituting a specific value from the full dataset for any missing or null values. Here, we are imputing using mean.

3.5 Outlier Removal

A data item or object that considerably differs from the other data items of the dataset. The outcomes may become statistically significant by removing outliers. Visualization, applying mathematical formulas to the dataset, or a statistical method can all be used to identify outliers.

3.6 Data Visualization

An approach for visualizing data that makes the dataset simpler to comprehend, observe, and analyse the data by providing a good, organized pictorial depiction of it. A pair plot is a scatterplot matrix used to visualize the pairwise relationships between several features in a dataset.

3.7 Model Building

Generalizing and learning from training data leads in the formation of a mathematical representation in machine learning. The machine learning model that was constructed is then utilized to evaluate new data and make predictions.

A technique for evaluating the effectiveness of a machine learning system is the train-test split. The approach entails splitting the dataset into two subgroups. The training dataset is the first subset used to fit the model (x). The test dataset (y) is the second subset that takes the dataset's input elements, generates predictions about them, and contrasts them with expected values. The target variable is used here (y) is the stroke attribute, and X contains the remaining dataset characteristics. The training data receives 70% of the dataset, while the testing data receives the remaining 30%. The dataset considered here is a classification dataset.

3.8 SMOTE

A dataset with a balanced distribution, or about equal numbers of samples from each class, is preferable. The classifiers, however, tend to favour the class set with more samples during training when the situation is the opposite, that is when the datasets are unbalanced. In other words, the system becomes biased because classifier will receive instruction using the classification having the most examples. Additionally, because the classifier labelled as belonging to the minority cannot sufficiently train itself, it is unable to identify correctly in subsequent phases. In essence, this is not desirable.

Unbalanced datasets can be balanced by employing resampling techniques. The approach uses various strategies to increase the data of the minority class to achieve classes with the same amount of data (oversampling). The other technique involves deleting the data from the majority class from the data set to produce a balanced dataset (undersampling). However, this strategy will result in information loss if the analysed dataset is too small. Since the dataset used is small, the oversampling technique called SMOTE is used to balance the data labels.

SMOTE stands for synthetic minority oversampling technique which is a potent procedure employed to solve the imbalanced data issue. Smote creates synthetic minority class samples without duplicating them. Based on its closest neighbours, smote chooses instances in the feature space that are close to one another, connects the examples with a line, and then generates a new sample at a point on the line. To be precise, a random sample is originally drawn from the minority class. Then, for that example, the location of K-Nearest Neighbours (k is normally equal to 5). At a specific point in feature space, a synthetic example is established between two examples and their randomly chosen neighbour.

The dataset considered had a data imbalance, as shown in Fig. 1, where class label 1 had only 167 instances and class label 0 had 3321 instances. To solve the issue, SMOTE was applied which resulted in an equal number of instances in both the class labels 0 and 1 which is 3321 providing a balanced dataset as shown in Fig. 2.

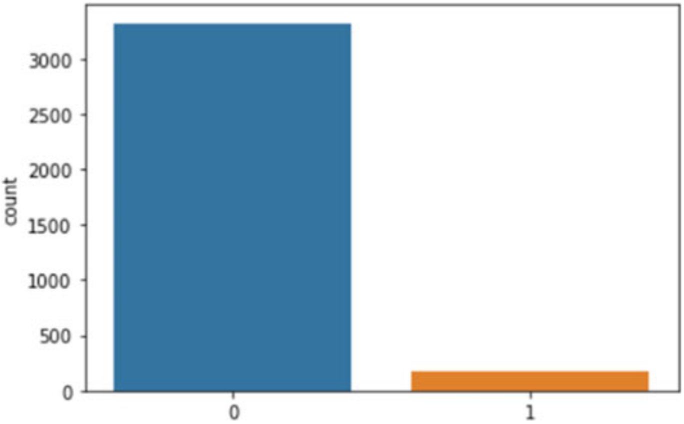


Fig. 1 Count plot before applying SMOTE

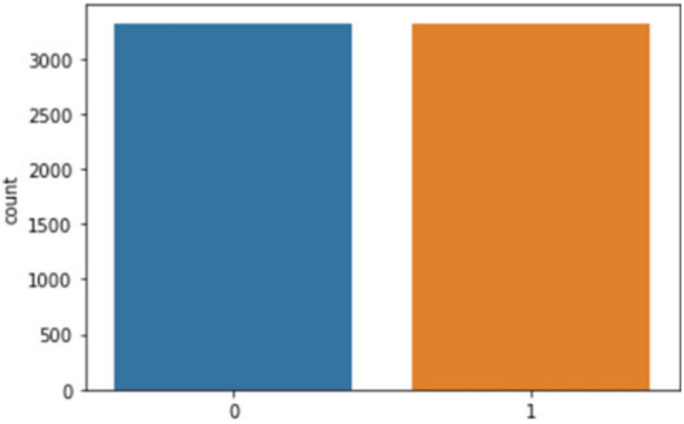


Fig. 2 Count plot before applying SMOTE

4 Ml Algorithms

4.1 Logistic Regression

For conditions demanding linear and bipartite classification, Logistic Regression is a simple and more efficacious approach. It is an amazingly simple classification model to impose and performs magnificently with bipartite separable classes. It predicts binary outcomes as yes or no by applying statistical analysis to previous outcomes from the dataset. This model predicts the data dependent variables by examining the interrelationships interlinking many variables that are independent.

$f(x) = b_o + b_1x_1 + b_2x_2 + \cdots + b_nx_n$, where $f(x) =$ sigmoid function.
$p(x) = \frac{1}{1+e^{-f(x)}}$, where $p(x) =$ probability function.

If $p(x) \geq f(x)$, then $(y = 1)$ and if $p(x) < f(x)$, then $(y = 0)$. $b = b + \alpha \times (y$ - pred$) \times$ pred $\times$ (1 - pred) $\times\ x$, where $\alpha =$ learning rate, $b =$ coefficient.

We have used three types of solvers:

- Limited-memory (Broyden–Fletcher–Goldfarb–Shanno Algorithm) BFGS: it is a member of Quasi-Newton methods algorithms. It approximates the secondary derivative (Hessian) of problems on quantization for which the secondary derivative could not be computed.
- Library for Large Linear Classification: Large Linear Classification Library uses the algorithm of coordinate descent. The foundation of coordinate descent is the looped solution of univariate optimization problems to minimize a multivariate function.
- Newton's Method: Newton's method employs a superior quadratic function minimization, because it uses a second-order approximation (that is, the first and second partial derivatives). A square matrix of second partial derivatives, the Hessian can be thought of as a twisted version of gradient descent.

4.2 KNN

It is one of the effortless machine learning algorithms in machine learning which is based on supervised learning techniques. This technique presumes resemblances between new data points and accessible cases and allocate new cases to the label that most nearly resembles the accessible categories. This technique assembles all accessible data and classifies the new data points based on similarity. This depicts that when a new data point emerges, it can be easily be differentiated into the appropriate label using this method. This could be used for both the problems which is classification and regression, but are fundamentally employed for problems on classification. Larger K values result in smoother separation curves and less model complexity. On the other hand, smaller K values tend to overfit the data and produce a more complex model.

We have used three types of distance metrics:

- Euclidean metrix: It is the length of two points in the Euclidean space. It is the addition of the squared disparities in-between the two data points divided by two, or its square root.
- The Euclidean distance is given by $d = \sqrt{}((x_2 - x_1)^2 + \left(y_2 - y_2\right)^2\big)$, where the coordinates are x and y.
- Minkowski metrix: It is a generalization of the Euclidean and Manhattan distance measurements that include the "order" or "p" parameter, which enables the calculation of other distance measures.
- Considering 2-dimensional space has three points (P_1, P_2, P_3) comprising of points (x_1, x_2, x_3), (y_1, y_2, y_3).
- The Minkowski distance will be given by $(|x_1 - y_1|^p + |x_2 - y_2|^p + |x_2 - y_2|^p)^{1/p}$.
- Manhattan distance: Distance between two points measured along orthogonal axes
- The distance is given by

$$|x_1 - x_2| + |y_1 - y_2|,$$

where x and y are coordinates.

4.3 Decision Tree

Decision Trees are the most effective and popular method of classification and prediction. It is similar to a schema, where the leaf node shows the class label, branch displays the outcome of the test conducted, and the internal node displays the test conducted on each of the attribute. There is a very effective technique for choosing the most optimized attribute for the root and the sub-nodes termed as the (ASM) attribute selection measure.

ASM has two common techniques:

- Gini Index: It a technique under the Decision Tree used to select the most optimized root node and the other nodes using Gini impurity measure. To put it in perspective, a Decision Tree tries to formulate sequential queries so that it divides the data into more manageable groupings.

$$\mathrm{GINI} = 1 - \sum_{i=1}^{c} \left(p_i^2\right).$$

- Entropy:
- It is a technique used under Decision Tree that is used in information theory to identify how pure a set of observation.

$$\text{Entropy} = \sum_{i=1}^{c} p_i * \log_2(p_i).$$

5 Outcome and Discussion

In this section, three classifier algorithms—LR, KNN, and Decision Tree—have been used.

We used three solvers of LR for the dataset, resulting in accuracy values of 0.7787, 0.7881, and 0.7868 for Newton's Method, limited-memory BFGS, and Library for Large Linear Classification respectively as given in Table 1. Library for Large Linear Classification provides the highest accuracy in this case 0.7881.

Since we used three KNN metrics, the accuracy for the Euclidean Minkowski and Manhattan models was 0.8590, 0.8549, and 0.8590, respectively from Table 2. The maximum accuracy is shared by here Euclidean and Minkowski and is 0.8590.

Table 3 shows that, for the dataset we used two ASMs of DCT, the GINI index and the entropy, with accuracy values of 0.8128342245989305 and 0.7941176470588235 before and after feature importance of 0.8121657754010695 and 0.803475935828877, respectively. And after factoring in feature importance, we achieved a highest score.

To access various performance models k-fold cross-validation is used. In the literature, there are numerous cross-validation techniques. In the study makes use of 5-fold cross-validation. This dataset is split into five pieces using this manner, one of which serves as the test set and the other four as the training set. Each component that is still present becomes the test set after the classification procedure is completed using this division. Each component's classification success is determined when the

Table 1 For Logistic Regression

Classification algorithm	Accuracy	Precision	Recall	Specificity	F-1
Limited-memory BFGS	0.7787	0.5455	0.1068	0.9741	0.1787
Library for large linear classification	0.7881	0.5455	0.1115	0.9744	0.1851
Newton's method	0.7868	0.5455	0.1108	0.9744	0.1841

Table 2 For KNN

Classification algorithm	Accuracy	Precision	Recall	Specificity	F-1
Euclidean	0.8590	0.3182	0.1123	0.9656	0.1660
Minkowski	0.8590	0.3182	0.1123	0.9656	0.1660
Manhattan	0.8549	0.3030	0.1047	0.9648	0.1556

Table 3 For Decision Tree

Classification algorithm	Test score before feature importance	Test score after feature importance
GINI index	0.8128342245989305	0.8121657754010695
Entropy	0.7941176470588235	0.803475935828877

Table 4 For k-fold

Classification algorithm	Accuracy (%)
LR solver	
(a) Limited-memory BFGS	76.456
(b) Library for Large Linear Classification	76.345
(c) Newton's method	76.468
KNN metrics	
(a) Euclidean	86.250
(b) Minkowski	86.250
(c) Manhattan	86.533
Decision Tree-ASM	
(a) Gini index	90.106
(b) Entropy	89.837

classification accuracy rate for that component is taken into account. When there is not enough data to apply other more effective techniques, such as three-way splitting (training, validating, testing) or using holdout datasets, machine learning typically introduces cross-validation to model improve the prediction.

As given in Table 4, for limited-memory BFGS, Library for Large Linear Classification and Newton's Method, we obtained accuracy of 76.456%, 76.345%, and 76.468% respectively after using k-fold for LR. Here, Newton's Method has the highest accuracy at 76.468%. With k-fold for KNN, we were able to reach an accuracy of 86.25%, 86.250%, and 86.533% for Minkowski, Manhattan, and Euclidean for KNN metrics, respectively. Here, we have Manhattan's highest accuracy score of 86.533%. Following the use of k-fold for DCT-ASM, we obtained accuracy of 90.106% and 89.837% for Gini index and entropy, respectively. Here, we have the highest Gini index accuracy of 90.106%.

6 Conclusion

In this study, an accurate classification for the dataset of stroke patients has been made. Since dataset is unbalanced, and when the classification's outcomes are analysed using the dataset in its original form, it becomes clear that it is biased towards a specific class label, which is unacceptable for a study. However, it was discovered that the outcomes were more reliable and accurate when the modelling output from the SMOTE applied dataset was analysed. Because of this, it is advised that dataset balancing techniques is used by researchers to get more precise results in the event that they run into an unbalanced dataset issue. Logistic Regression, KNN, and Decision Tree Classifier are three ML methods that were employed in this work to examine the accuracy and contributing features for stroke. From the performance analysis we see that Decision Tree Classifier with k-fold performs better than other methods by achieving 90.106% accuracy for the top attributes. The model can help those who are showing signs that they might be having a stroke. A new age of machine learning (ML)-based decision support for the healthcare sector may be resulted. It has been investigated whether there is a link between these conditions and a person's chance of having a stroke. Therefore, it will be less likely for us to have a stroke if this problem is recognized and treated well from the outset. This would enable medical professionals to begin procedures as soon as is practical to aid stroke patients in regaining as much function as possible.

References

Alotaibi NN, Sasi S (2016) Stroke in-patients. In: Transfer to the ICU using ensemble based model

Bhagwani H, Agarwal S, Kodipalli A, Martis RJ (2021) Targeting class imbalance problem using GAN. In: 2021 5th international conference on electrical, electronics, communication, computer technologies and optimization techniques (ICEECCOT). IEEE, pp 318–322

Bhoomika R, Shahane S, Siri TC, Rao T, Ashwini K, Chodon PK (2022) Ensemble learning approaches for detecting Parkinson's disease

Chen X, Wei C, Wu W, Guo L, Liu C, Lu G (2021) Based on machine learning algorithm: construction of an early prediction model of integrated traditional Chinese and western medicine for cognitive impairment after ischemic stroke

Dhanush N, Prajapati PR, Revanth M, Ramesh R, Kodipalli A, Martis RJ (2021) Prediction of gold price using deep learning. In: 2021 IEEE 9th region 10 humanitarian technology conference (R10-HTC). IEEE, pp 1–5

Emon MU, Keya MS, Meghla TI, Rahman MM, Shamim Al Mamun M, Shamim Kaiser M (2021) Performance analysis of machine learning approaches in stroke prediction

Guha S, Kodipalli A, Rao T (2022) Computational deep learning models for detection of COVID-19 using chest X-ray images

Guha S, Kodipalli A, Rao T (2022) Computational deep learning models for detection of COVID-19 using chest X-ray images. In: Emerging research in computing, information, communication and applications: proceedings of ERCICA 2022. Springer Nature Singapore, Singapore, pp 291–306

Gururaj V, Shriya VR, Ashwini K (2019) Stock market prediction using linear regression and support vector machines. Int J Appl Eng Res 14(8):1931–1934

Gururaj V, Ramesh SV, Satheesh S, Kodipalli A, Thimmaraju K (2022) Analysis of deep learning frameworks for object detection in motion. Int J Knowl Based Intell Eng Syst 26(1):7–16

Hung C-Y, Chen W-C, Lai P-T, Lin C-H, Lee C-C (2017) Comparing deep neural network and other machine learning algorithms for stroke prediction in a large-scale population-based electronic medical claims database

Kodipalli, A., & Devi, S. (2021). Prediction of PCOS and mental health using fuzzy inference and SVM. Front Public Health 1804

Kodipalli A, Guha S, Dasar S, Ismail T (2022) An inception—ResNet deep learning approach to classify tumours in the ovary as benign and malignant. Expert Syst e13215

Kodipalli A, Devi S, Dasar S, Ismail T (2022) Segmentation and classification of ovarian cancer based on conditional adversarial image to image translation approach. Expert Syst e13193

Rachana PJ, Kodipalli A, Rao T (2022) Comparison between ResNet 16 and inception V4 network for Covid-19 prediction

Rachana PJ, Kodipalli A, Rao T (2022) Comparison between ResNet 16 and inception V4 network for COVID-19 prediction. In: Emerging research in computing, information, communication and applications: proceedings of ERCICA 2022. Springer Nature Singapore, Singapore, pp 283–290

Rao T, Devamane S, Moumen A (2022) Machine learning approaches for stratification of Parkinson's disease

Ray S, Alshouiliy K, Roy A, AlGhamdi A, Agrawal DP (2020) Chi-squared based feature selection for stroke prediction using AzureML

Ruchitha PJ, Richitha YS, Kodipalli A, Martis RJ (2021) Segmentation of ovarian cancer using active contour and random walker algorithm. In: 2021 5th international conference on electrical, electronics, communication, computer technologies and optimization techniques (ICEECCOT). IEEE, pp 238–241

Ruchitha PJ, Sai RY, Kodipalli A, Martis RJ, Dasar S, Ismail T (2022) Comparative analysis of active contour random walker and watershed algorithms in segmentation of ovarian cancer. In: 2022 international conference on distributed computing, VLSI, electrical circuits and robotics (DISCOVER). IEEE, pp 234–238

Sagarnal C, Devamane SB, Hosamani R, Rao T (2021) Deep learning approaches for COVID-19 diagnosis

Sanjana S, Sanjana S, Shriya VR, Vaishnavi G, Ashwini K (2021) A review on various methodologies used for vehicle classification, helmet detection and number plate recognition. Evol Intel 14(2):979–987

Shoily TI, Islam T, Jannat S, Tanna SA, Alif TM, Ema RR (2019) Detection of stroke disease using machine learning algorithms

Yagin FH, Cicek IB, Zeynep Kucukakcali H (2021) Classification of stroke with gradient boosting tree using smote-based oversampling method

Zacharia S, Kodipalli A (2022) Covid vaccine adverse side-effects prediction with sequence-to-sequence model. In: Emerging research in computing, information, communication and applications: proceedings of ERCICA 2022. Springer Nature Singapore, Singapore, pp 275–281

Prediction of Cost for Medical Care Insurance by Using Regression Models

J. Ruth Sandra, Sanjana Joshi, Aditi Ravi, Ashwini Kodipalli, Trupthi Rao, and Shoaib Kamal

1 Introduction

Around 1 billion individuals spend more than 10% of their take-home income on health, while half a billion were forced into poverty by out-of-pocket expenditure for medical treatments (https://www.who.int/news/item/12-12-2021-more-than-half-a-billion-people-pushed-or-pushed-further-into-extreme-poverty-due-to-health-care-costs). Hospital expenses include medicines, consultation, rent for a room, ambulance services, and equipment utilized during testing and operation. Medical cost is the total expense of preventing, treating, and recovering from an ailment; the insurance covers the whole or a part of the risk of a person incurring medical expenses (https://www.investopedia.com/terms/m/medical-expenses.asp; Hockenberry et al. 2012). These costs amount to a significant charge by the end of the treatment. Therefore, insurance aids in providing financially secure medical

J. R. Sandra · S. Joshi · A. Ravi
Department of Information Science and Engineering, Global Academy of Technology, Bangalore, India
e-mail: 1ga19is045@gat.ac.in

S. Joshi
e-mail: 1ga19is049@gat.ac.in

A. Ravi
e-mail: 1ga19is002@gat.ac.in

A. Kodipalli (✉) · T. Rao · S. Kamal
Department of Artificial Intelligence and Data Science, Global Academy of Technology, Bangalore, India
e-mail: dr.ashwini.k@gat.ac.in

T. Rao
e-mail: trupthirao@gat.ac.in

S. Kamal
e-mail: shoaib.kamal@gat.ac.in

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_23

assistance. Investing in a good insurance scheme at early stages of life can profit in huge benefits due to the interest rates offered by the corporations. Insurance also helps in overcoming the drawback of medical inflation with the inflation rates rising to higher horizons; hospital expenses are one of the major expenditures burdening and financially draining lives. The recent onset of the pandemic has put to light many hospitalization incidents. The main cause was with respect to respiratory functioning, and the people were burdened by medical expenditure carried out during the treatment and recovery at the hospital. People with insurance were more in beneficiary than the non-insured people. Insurance offers protection from financial loss during medical emergencies. Traditional methods used for calculating health insurance cost consume huge amount of time (Kan et al. 2019; Thongpeth et al. 2021). The regression models compared in the study include linear regression, LASSO regression, polynomial regression, ridge regression, SVM regressor, decision tree regressor, and random forest regressor (Iqbal et al. 2021). The paper is organized in the following manner. Section 2 discusses the methodology, Sect. 3 describes the obtained results, Sect. 4 presents the analysis of results, and the article is concluded in Sect. 5.

2 Literature Survey

A generalized regression model was used to predict the cost of healthcare of smokers at the time of cessation; the data was collected by individuals who were enrolled in cessation therapies (Panda et al. 2022; Panay et al. 2019). The cost was significantly varied when people quit smoking to continuous smokers, but this cannot not be used for short-term cost analysis since the smoking habit has a long-term effect. Kaushik et al. (2022) implemented penalized linear regression for prediction of healthcare expenses to achieve better accuracy compared to the traditional linear regression. They drew a conclusion suggesting LASSO regression to be a suitable predicting algorithm for betterment of risk adjustment for cost prediction. Lahiri and Agarwal (2014) and Shyamala Devi et al. (2021) propose a comparative study between statistical models and traditional machine learning models for computing the medical costs for chronic diseases. Random forest regression gave outstanding results with metrics of RMSE of 0.3542 and 0.3125. Rachana et al. (2022a) and Guha et al. (2022a) have mainly contribute by exploring the usage of machine learning algorithms for forecasting the insurance cost in the healthcare industry. The research uses different regression models to estimate the cost of healthcare insurance. Gradient boosting model gives better results than the others with an accuracy of 86 regression models applied on various other metrics which results in random forest giving the highest result of 0.85388; this has facilitated insurers and patients. Rao (2022) has considered a dataset consisting of patients; health records regarding the billing details; and medical exams from a Japanese hospital. They designed a new regression system for medical expenses prediction which presents the intention of decision, and their results outperformed the pre-existing models of ANN. Sagarnal et al. (2021) and

Bhoomika et al. (2022)—in this paper, an AI and an ML models are used to predict the health insurance cost incurred by individuals and resulted in an accuracy of 92.72 using key performance metrics for an ANN-based regression model. Kodipalli et al. (2022a) and Ruchitha et al. (2021) performed a deep rooted analysis on the features of the expenditure dataset to deduce insightful statistics claiming the rise in medical costs with respect to the calculated conditional probabilities. Kodipalli et al. (2022b) and Ruchitha et al. (2022)—the performance of the models in this paper is said to be 8886 which shows 0.14 as the *F*-static value. Gururaj et al. (2022) and Guha et al. (2022b) propose a prediction system for health insurance using machine learning. The dataset used contains parameters that help in prediction collected from various healthcare facilities in the USA. The reasons for the increase in cost were also deduced using linear regression, the accuracy obtained is 81.3 higher risk at hospitalization than non-smokers, and therefore, their treatment too varied on this factor and increased the medical expenditure. Rachana et al. (2022b) devised a telegram integrated chatbot that was trained to evaluate the insurance premium based on communication with the user. Among numerous algorithms implemented to train the model, XGBoost produced the best outcome. Zacharia and Kodipalli (2022) considered a dataset consisting of the parameters of BMI, age, insurance charges, and gender and implemented regressor algorithms. Random forest regressor yielded the best result against other algorithms for *R*-squared metric. Kodipalli and Devi (2021)—this research has incorporated numerous regression models out of which polynomial regression has outperformed giving 5100.53 as the RMSE value and 0.80 as the *R*-squared value in comparison with the other models. Bhagwani et al. (2021) have shown significance using various predicting models to calculate the health costs. The authors performed a deep analysis on various factors affecting the cost for two successive years. The results conclude that the model performs better than the other traditional models. Dhanush et al. (2021) proposed a system that includes assessment of a wide range of machine learning models for the prediction of healthcare spending. The authors judge the reporting and the quality of the models to evaluate the risk associated with each one. Sanjana et al. (2021) have shown importance on how accurate the prediction of healthcare costs should be considered. A deep neural network is developed to calculate the future costs based on the parameters collected from German. The performance of various algorithms is calculated, and the authors have concluded that ridge regression gives the best result. Gururaj et al. (2019) and Kodipalli (2018) have concluded the important parameters which lead to high cost of medical insurance by collecting data from various healthcare centres across Korea. Machine learning algorithms like logistic regression, random forest, and XGBoost were implemented. The main characteristic is that this research gives results for all the diseases.

3 Methodology

3.1 Dataset Description

The dataset used in this research is taken from Kaggle; this is obtained from numerous finance sections of the medical facilities, public surveys, multinational banks, and insurance providers. The features of the dataset in columns are age—varies from infant to an adult to consider for various types of medical treatments, sex human anatomy is different in males and females where in hormonal changes play a defining factor for nursing care, children the number of children/dependants, BMI indicates the body's mass which is a pre requisite for providing certain medication, smoker smoking is one of the deal breakers as it is a staunch habit which can cause serious abnormalities, region the location background of person can explain certain lifestyle habits and ailments and economic conditions and charges is predicted considering all the above factors, all of which are in a csv file format. The dataset considered contains a total of 1338 entries. It is split in the ratio of 80:20 for the training set and the testing set, respectively.

3.2 Dataset Analysis

The processed data has undergone training of regression models to predict the cost of insurance against various parameters. Pictorial representation in terms of bar graphs and scatterplots is plotted for insightful information gathering.

In Fig. 1, the graph shows the number of smokers and non-smokers on the basis of region. It is observed that the upper limit count for smoker is less than 100, whereas the non-smoker is greater than 250.

In Fig. 2, charges versus region is shown; this can help to estimate the medical expense. The insurance is calculated and predicted taking region as a parameter. The region is divided into the classes of southwest, southeast, northwest, and northeast.

3.3 Architecture Framework

The implementation for this research includes open-source libraries such as NumPy, matplotlib, and Pandas for complex mathematical computation of values, graphical representations, and dataset analysis, respectively, to arrive at meaningful analytics and predict accurate outcomes (Fig. 3).

Regression Models:

Linear Regression: It is a learning algorithm where prediction of the dependent variable is computed based on the regressor variable. The aim of executing the

Fig. 1 Smoker versus count

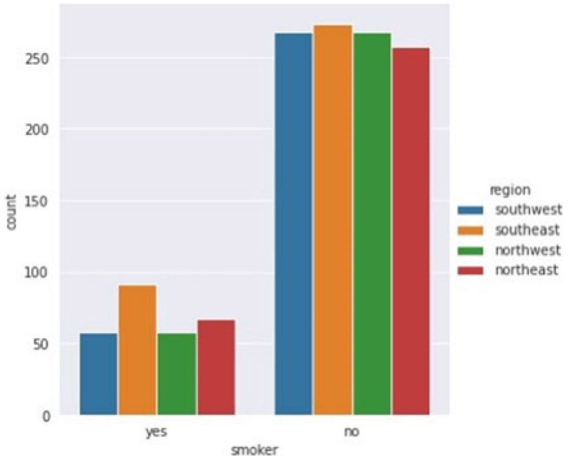
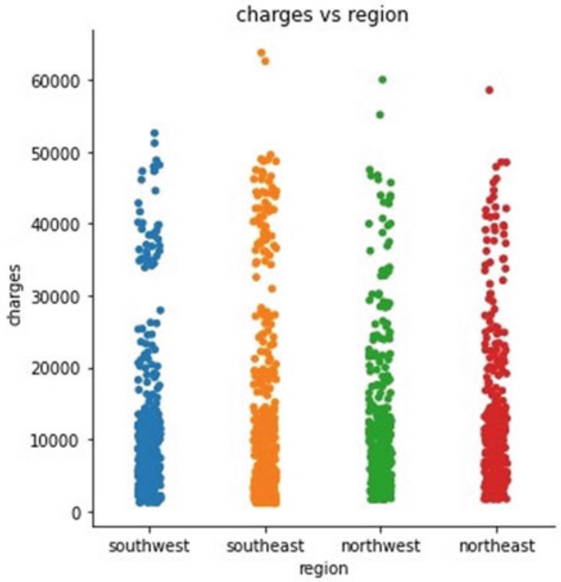


Fig. 2 Graphical representation of charges versus region

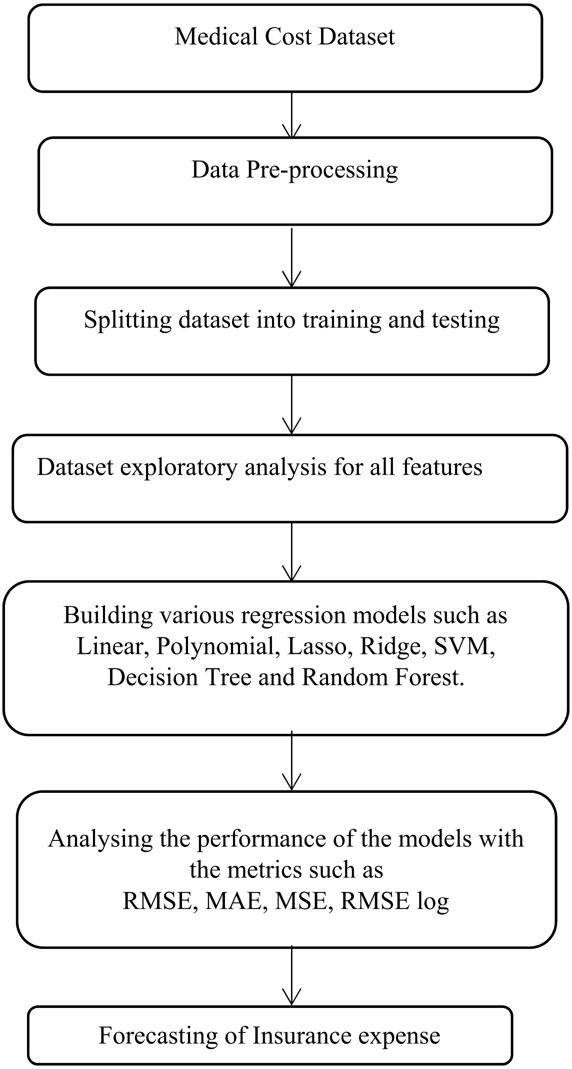


prediction task is fulfilled by obtaining the best-fit line based on the criterion variable. The hypothesis function computation is implemented using the formula.

$$y = \theta_1 + \theta_2 \cdot x \tag{1}$$

$$J = \frac{1}{n}\sum_{i=1}^{n}\left(\text{pred}_i - y_i\right)^2 \tag{2}$$

Fig. 3 Architecture framework


Medical Cost Dataset
Data Pre-processing
Splitting dataset into training and testing
Dataset exploratory analysis for all features
Building various regression models such as Linear, Polynomial, Lasso, Ridge, SVM, Decision Tree and Random Forest.
Analysing the performance of the models with the metrics such as RMSE, MAE, MSE, RMSE log
Forecasting of Insurance expense

Polynomial Regression: The depiction of the relationship between the criterion variable and the predictor variable is evaluated for an nth degree polynomial. The algorithm hopes to solve the complications associated with nonlinear datasets by deriving the best-fit curve. The hypothesis function computation is implemented using the formula.

$$y = b_0 + b_1 x_1 + b_2 x_1^2 + b_2 x_1^3 + \cdots + b_n x \tag{3}$$

LASSO Regression: The motivation of LASSO regression implementation is to achieve better prediction through the process of shrinkage. Shrinkage refers to the data points reduced to a central point, typically known as mean. It is one of the regularization approaches. The algorithm is favourable in overcoming overfitting of the model but does not contribute to feature selection.

$$\sum_{i=1}^{M}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{M}\left(y_i - \sum_{j=0}^{p} w_i \times x_{ij}\right)^2 + \lambda \sum_{j=0}^{p}|w_j| \tag{4}$$

Ridge Regression: The objective of ridge regression is to minimize the model complexity by regularization. It applies restraint on the coefficients to regularize the large values thereby lowering multicollinearity. The loss function is given by

$$\sum_{i=1}^{M}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{M}\left(y_i - \sum_{j=0}^{p} w_i \times x_{ij}\right)^2 + \lambda \sum_{j=0}^{p} w_j^2 \tag{5}$$

Support Vector Regressor: SVM attempts to determine the best-fit line within a threshold value, known as the hyperplane, rather than minimizing the error rate. SVMs are preferred for larger datasets; the computational speed of the training model is considerably faster.

$$\text{MIN}\frac{1}{2}x||w||^2 \tag{6}$$

$$|y_i - w_i x_i| \le \varepsilon \tag{7}$$

Decision Tree Regressor: Decision tree regression trains a model in a tree structure formation consisting of decision nodes at each computation. Since regression algorithm's motivation is to predict a continuous valued output, the decision tree detects the branches based on true or false.

$$\varepsilon_i = y_i - \hat{y}_i \tag{8}$$

$$\text{RSS} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{9}$$

Random Forest Regressor: Random forest regressor is based on the implementation of multiple decision trees and arriving at a final outcome on the basis probability. The major drawback of decision of overfitting of a model can be tamed by the applying random forest regressor.

$$fi_j = \frac{\sum_{j:\text{node } j \text{ splits on feature } i} ni_j}{\sum_{k \in \text{ all nodes}} ni_k} \tag{10}$$

Performance Metrics:

RMSE: The difference between the predicted and actual values is obtained for each observation. The values are squared, and the average is taken for all the observations. The square root of the value obtained provides the final result.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \overline{y_i})^2}{n}} \tag{11}$$

MAE: In an observation, the predicted value is subtracted from the true value to obtain the absolute error. The mean of all the absolute errors for a particular set of observations and predictions gives the result. It is commonly used in regression problems and is also called the L1 loss function.

$$\text{MAE} = \frac{\sum_{i=1}^{n} y_i - \overline{y_i}}{n} \tag{12}$$

MSE: The predicted value is subtracted from the actual value. The differences are squared for all the observations, and the final value is obtained by taking the sum of all the squared values to the total number of observations. The value obtained is always positive, and if the obtained result is zero, it depicts that the system has no errors.

$$\text{MSE} = \frac{\sum_{i=1}^{n}(y_i - \overline{y_i})^2}{n} \tag{13}$$

RMSLE: The value is obtained by taking the logarithmic values of the predicted and observed values. The average of the difference between the observed and predicted values is obtained. The final value is deduced by taking the square root of the previous calculation. One is added to both the values.

$$\text{RMSLE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(y_i + 1) - \log(x_i + 1))^2} \tag{14}$$

R-Squared Score: It is one of the commonly used metrics to measure the performance of the models as it produces accurate results. It shows how far is the predicted value from the actual value for the observations. If the value is negative, it means that the model is not trained properly.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(\hat{y}_i - y_i\right)^2}{\sum_{i=1}^{n} (y_i - \overline{y_i})^2} \tag{15}$$

4 Results

Comparison of the metric values with various regression algorithms implemented in the model are shown in Table 1. The column-wise representation denotes the performance metrics against the row-wise regression models. The values indicate the respective output determined against the algorithms. Random forest produces the best result with *R*-squared score of 0.8643297.

In Fig. 4, the graph represents all the other regression models apart from random forest, and it shows a slight variation when plotted against the performance metrics of R2 score. A gradual increase in the score can be observed in the graph from linear regression to decision tree.

In Fig. 5, the graph showcases the regression algorithms implemented against the performance metric of R2 score. A drastic spike in the performance can be observed where random forest.

5 Conclusion

Health insurance is evidently a top priority currently. The financial overload that comes with the absence of insurance can be a major threat to an individual's life savings. Hence a system assists in forecasting the costs induced under a medical crisis to achieve an estimate of expenditure for planning and management in advance. The proposed system focuses on achieving the same objective by studying the outcomes of various regression algorithms to establish the best prediction. The inference drawn from the research is that random forest yielded the best outcome with the *R*-squared score of 0.8643297.

Table 1 Analysis of regression models against the metrics

	Mean absolute error (MAE)	Mean squared error (MSE)	Root mean squared error (RMSE)	Root mean squared log error (RMSLE)	R2 score
Linear regression	4123.4359203657195	32,802,945.33428212	5727.3855583749655	8.653014433049888	0.7622204758053099
Polynomial regression	4123.4359203657195	32,802,945.33428212	5727.3855583749655	8.653014433049888	0.7626204758053099
Ridge regression	4223.378180245106	32,665,221.898791034	5715.349674236129	8.650910759700318	0.7632187950912097
LASSO regression	41,743.149736231281	32,628,199.585052606	5712.10990659779	8.650343745185461	0.7734871596559035
SVM regressor	4123.4359203657195	32,802,945.33428212	5727.3855583749655	8.653014433049888	0.7682204758053099
Decision tree regressor	4124.4197236870605	32,799,274.661468703	5727.065100159828	8.65295847956512	0.7622470835024588
Random forest regressor	2525.2876418635083	18,716,431.406932157	4326.249115218881	8.372456190434972	0.8643297390154836

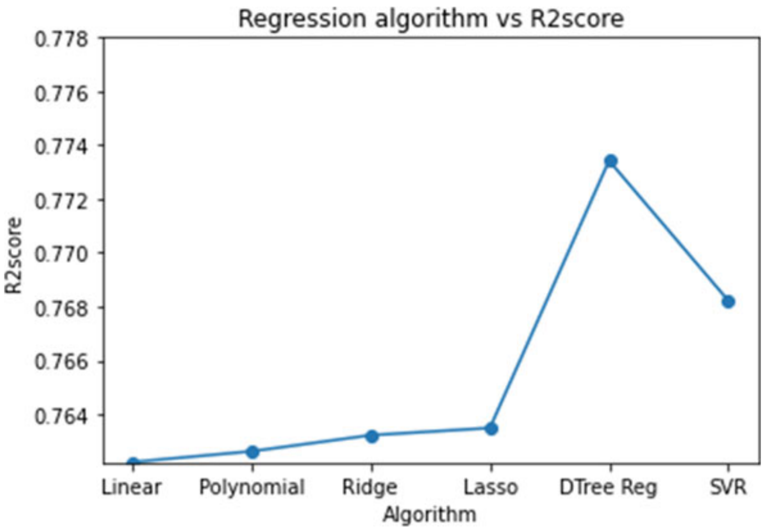


Fig. 4 Regression versus *R*-squared score

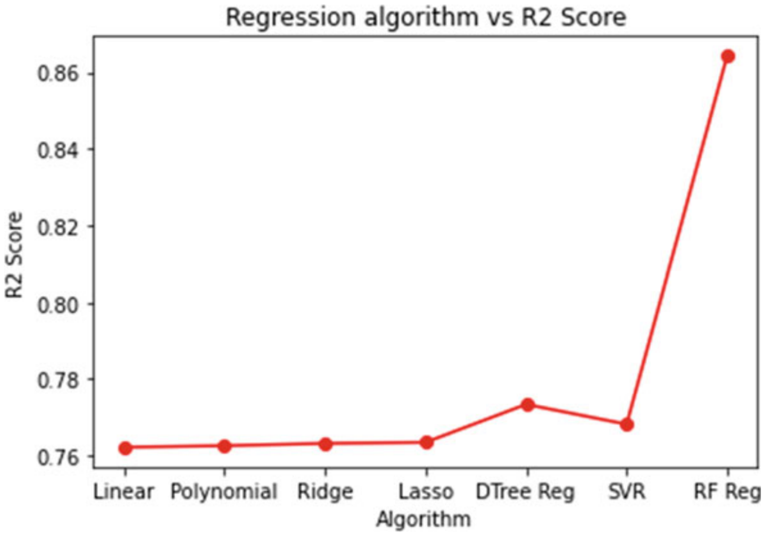


Fig. 5 Regression versus *R*-squared score

References

Bhagwani H, Agarwal S, Kodipalli A, Martis RJ (2021) Targeting class imbalance problem using GAN. In: 2021 5th international conference on electrical, electronics, communication, computer technologies and optimization techniques (ICEECCOT). IEEE, pp 318–322

Bhoomika R, Shahane S, Siri TC, Rao T, Kodipalli A, Chodon PK (2022) Ensemble learning approaches for detecting Parkinson's disease

Dhanush N, Prajapati PR, Revanth M, Ramesh R, Kodipalli A, Martis RJ (2021) Prediction of gold price using deep learning. In: 2021 IEEE 9th region 10 humanitarian technology conference (R10-HTC). IEEE, pp. 1–5

Guha S, Kodipalli A, Rao T (2022a) Computational deep learning models for detection of COVID-19 using chest X-ray images

Guha S, Kodipalli A, Rao T (2022b) Computational deep learning models for detection of COVID-19 using chest X-ray images. In: Emerging research in computing, information, communication and applications: proceedings of ERCICA 2022. Springer Nature Singapore, Singapore, pp. 291–306

Gururaj V, Shriya VR, Ashwini K (2019) Stock market prediction using linear regression and support vector machines. Int J Appl Eng Res 14(8):1931–1934

Gururaj V, Ramesh SV, Satheesh S, Kodipalli A, Thimmaraju K (2022) Analysis of deep learning frameworks for object detection in motion. Int J Knowl Based Intell Eng Syst 26(1):7–16

Hockenberry JM, Curry SJ, Fishman PA, Baker TB, Fraser DL, Cisler RA et al (2012) Healthcare costs around the time of smoking cessation. Am J Prev Med 42(6):596–601

https://www.investopedia.com/terms/m/medical-expenses.asp

https://www.who.int/news/item/12-12-2021-more-than-half-a-billion-people-pushed-or-pushed-further-into-extreme-poverty-due-to-health-care-costs

Iqbal J, Hussain S, AlSalman H, Mosleh MA, Sajid Ullah S (2021) A computational intelligence approach for predicting medical insurance cost. Math Probl Eng

Kan HJ, Kharrazi H, Chang HY, Bodycombe D, Lemke K, Weiner JP (2019) Exploring the use of machine learning for risk adjustment: a comparison of standard and penalized linear regression models in predicting health care costs in older adults. PLoS ONE 14(3):e0213258

Kaushik K, Bhardwaj A, Dwivedi AD, Singh R (2022) Machine learning-based regression framework to predict health insurance premiums. Int J Environ Res Public Health 19(13):7898

Kodipalli A (2018) Cognitive architecture to analyze the effect of intrinsic motivation with metacognition over extrinsic motivation on swarm agents. Int J Electr Comput Eng 8(5):3984

Kodipalli A, Devi S (2021) Prediction of PCOS and mental health using fuzzy inference and SVM. Front Public Health 1804

Kodipalli A, Guha S, Dasar S, Ismail T (2022a) An inception—ResNet deep learning approach to classify tumours in the ovary as benign and malignant. Expert Syst e13215

Kodipalli A, Devi S, Dasar S, Ismail T (2022b) Segmentation and classification of ovarian cancer based on conditional adversarial image to image translation approach. Expert Syst e13193

Lahiri B, Agarwal N (2014) Predicting healthcare expenditure increase for an individual from medicare data. In: Proceedings of the ACM SIGKDD workshop on health informatics, pp 73–79

Panay B, Baloian N, Pino JA, Peñafiel S, Sanson H, Bersano N (2019) Predicting health care costs using evidence regression. Multidisc Digital Publ Inst Proc 31(1):74

Panda S, Purkayastha B, Das D, Chakraborty M, Biswas SK (2022) Health insurance cost prediction using regression models. In: 2022 International conference on machine learning, big data, cloud and parallel computing (COM-IT-CON), vol 1. IEEE, pp 168–173

Rachana PJ, Kodipalli A, Rao T (2022a) Comparison between ResNet 16 and inception V4 network for Covid-19 prediction

Rachana PJ, Kodipalli A, Rao T (2022) Comparison between ResNet 16 and inception V4 network for COVID-19 prediction. In: Emerging research in computing, information, communication and applications: proceedings of ERCICA 2022. Springer Nature Singapore, Singapore. pp 283–290

Rao T, Devamane S, Moumen A (2022) Machine learning approaches for stratification of Parkinson's disease

Ruchitha PJ, Richitha YS, Kodipalli A, Martis RJ (2021) Segmentation of ovarian cancer using active contour and random walker algorithm. In: 2021 5th international conference on electrical, electronics, communication, computer technologies and optimization techniques (ICEECCOT). IEEE, pp 238–241

Ruchitha PJ, Sai RY, Kodipalli A, Martis RJ, Dasar S, Ismail T (2022) Comparative analysis of active contour random walker and watershed algorithms in segmentation of ovarian cancer. In: 2022 international conference on distributed computing, VLSI, electrical circuits and robotics (DISCOVER). IEEE, pp 234–238

Sagarnal C, Devamane SB, Hosamani R, Rao T (2021) Deep learning approaches for COVID-19 diagnosis

Sanjana S, Sanjana S, Shriya VR, Vaishnavi G, Ashwini K (2021) A review on various methodologies used for vehicle classification, helmet detection and number plate recognition. Evol Intel 14(2):979–987

Shyamala Devi M, Swathi P, Purushotham Reddy M, Deepak Varma V, Praveen Kumar Reddy A, Vivekanandan S, Moorthy P (2021) Linear and ensembling regression based health cost insurance prediction using machine learning. In: Smart computing techniques and applications. Springer, Singapore, pp. 495–503

Thongpeth W, Lim A, Wongpairin A, Thongpeth T, Chaimontree S (2021) Comparison of linear, penalized linear and machine learning models predicting hospital visit costs from chronic disease in Thailand. Inf Med Unlocked 26:100769

Zacharia S, Kodipalli A (2022) Covid vaccine adverse side-effects prediction with sequence-to-sequence model. In: Emerging research in computing, information, communication and applications: proceedings of ERCICA 2022. Springer Nature Singapore, Singapore, pp 275–281

Load Frequency Control Using PID and Particle Swarm Optimization-ITAE

M Aruna

1 Introduction

Inevitably, electricity contributes to the growth of the country. Power systems, which are made up of generation, transformers, transmission, and loads, are extraordinarily large and extraordinarily complex systems. According to what was needed to meet demand at the load center, the load was changed. In order to maintain the stability and dependable functioning of the power system, the planned controllers are therefore required for the management of demand-related deviations (Manjit Bahadur Singh 2019; Narendra Kumar Jena 2020).

The goal of the work is to maximize fluctuation frequency for the necessary load requirement. ITAE performance standards dictate the effective tuning of values using the PID controller with optimization. The process includes calculating the gain of the PID controllers analytically with first-order lag plus time delay (FOLPD). Power system performance qualities including a decrease in overshoot, quick reactions, rejection of any load disturbances, and retaining resilience may be achieved in this way. To decrease the performance of the integral of time absolute error (ITAE), workers are chosen (Awouda et al. 2010).

A model has been created to examine the system's enhanced dynamic responses while utilizing a PID controller. The PSO model has been used to compare the performance characteristics. This study employs the PSO method to reduce frequency error to zero in quick response using the Simulink model (Singh et al. 2019). To discover the dynamic performance for various load disturbances, the study was based on automated generation control using conventional PID controller and fuzzy PID controller. AGC is a crucial component of the power system that keeps the frequency between the generating and load systems constant (Nagendra et al. 2020). The entire power system is controlled by the simulation load frequency control utilizing fuzzy

M. Aruna (✉)
Department of EEE, Nitte Meenakshi Institute of Technology, Bangalore, India
e-mail: aruna.m@nmit.ac.in

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_24

gain programmed with the PI controller. The goal of the study is to enhance the step load change system's dynamic performance (Cam et al. 2005).

The load frequency control modifies the generator's output according to variations in system frequency. Within the suggested boundaries, the LFC enables the power system's planned frequency. LFC offers rapid transient recovery and less system overrun. But, controllers do not always react right away, and they might not be able to modify loads more quickly. To match the system frequency, the standard PID values are tweaked utilizing several techniques. The suggested controllers' overshoots and settling periods are less pronounced than those of the traditional PID controllers' outputs. The techniques employed include adaptive weight particle swarm optimization, adaptive acceleration coefficients, and adaptive neuro-fuzzy inference system. Cutting-edge method is used to evaluate the traditional PID controller for the analysis of load frequency control system (Bahgaat et al. 2014).

A power system's load frequency control (LFC) using improved particle swarm optimization-based methodologies is given. The PSO techniques offer a number of benefits, but they can also have problems, such as local optimum entrapment because to early convergence (i.e., exploration problem). In order to improve PSO's capacity to be exploited, this research suggests a crossover operation scheme-based improvement to the PSO framework. The IPSO-optimized self-tuning PID controllers have been the study's main emphasis. Using MATLAB-Simulink, a comparison between a traditional proportional integral controller and the suggested PSO-based controller is made (Kr et al. 2010).

The suggested study is centered on improved load frequency control stability and efficiency of power generation utilizing a particle swarm optimization technique. The national electricity power system of Zambia is composed of hydroelectric units in Areas 1, 2, and 3, and a thermal power plant in Area 4. Due to integrated hydro and thermal facilities, the system is extremely complicated and nonlinear. The PSO and PID controller were used to determine the step response of frequency deviation of the power system, and the results were compared with existing isolated AGC system (Musonda et al. 2018). The simulation using MATLAB/Simulink explains how to carry out LFC optimization and compare the response results with existing primary frequency management of isolated area AGC system.

The goal of load frequency management is to reduce steady-state errors and transient changes to zero. Reactive power is dependent on the fluctuation in voltage when the real power in the power system varies, which also has an impact on the system frequency. As a result, actual power and reactive power are each separately managed. LFC is in charge of the active power. Applying IAE and ITAE criteria helps to maintain the system's optimum frequency because the major goal is to balance the generation with regard to load changes and losses (Kushwaha et al. 2016).

The power systems turn out to require the use of optimization techniques. Modern men have developed regular routines during the previous two decades, and as a result, the current culture likewise practices routinely repeated cycles or patterns in everyday life. The transmission of electrical energy from the generating side to the load side is one of the main purposes of the power system. The transfer of power between the two ends must be kept in the active and reactive power balances. By keeping an eye on the frequency deviation, it is possible to track changes in the active power. The system becomes unstable if this frequency deviates significantly from its nominal value of (50 + 0.50) Hz. Controlling the system frequency becomes essential in order to maintain stability Thus, automatic generation control (AGC) refers to the regulation of frequency and active power (Law 2015).

Understanding the particle swarm optimization (PSO) approach for implementing frequency fluctuation control has been aided by the literature review. Power system optimization (PSO) allows for the tuning of PID controller gains, load frequency control (LFC), and corrective action. The error function is constructed utilizing frequency deviation and area control error to get substantial performance (ACE). Using a controller whose gain values are determined by an algorithm, the area control error may be decreased and the performance index can be adjusted.

The load frequency control regulates the load's relationship to the generator in order to maintain a fixed frequency and adhere to pre-stated parameters. For the user to receive dependable and high-quality power, load frequency management is crucial. Constant frequency is recognized as the characteristic of such a regular operating system for such systems. The primary goal of a load control center's LFC is to keep system frequency within the power plants' optimum generating power within planned power in order to track load changes. The major goal of this study is to reduce the frequency fluctuations and transient response of the system in order to safeguard the stability of the power system and maintain the overload.

2 Mathematical Model of Power System

Using components of the power system, such as the governor, turbine, generator, and load, one may construct a model of the power system. In order to further examine regulation using LFC, this model was constructed in MATLAB-Simulink utilizing equivalent transfer functions of each power system component. The normal operating frequency in India is 50 Hz and the nominal frequency variation of ± 2.5 Hz. The whole system gets affected because of the different types of system limitations.The frequency and speed of generation match the load motor speed with frequency. The purpose of this work is to design optimized parameters of PID controller values to obtain through the PSO algorithm for constraining constant frequency for any variation in load demand. The power should be maintained in pre-restricted value to minimize the error of the system.

2.1 Load Frequency Control

To give users reliable electricity, load frequency control, or LFC, is crucial in power systems. The power requirement varies randomly and often for every second depending on the loads. The necessity for quick generation regulation arises from

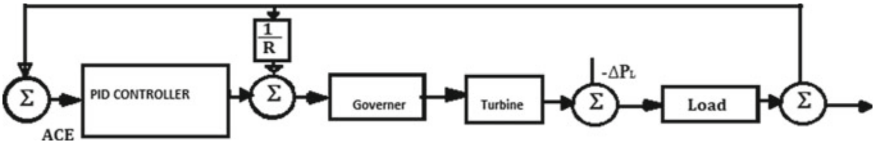


Fig. 1 Block diagram for the power system's load frequency control model

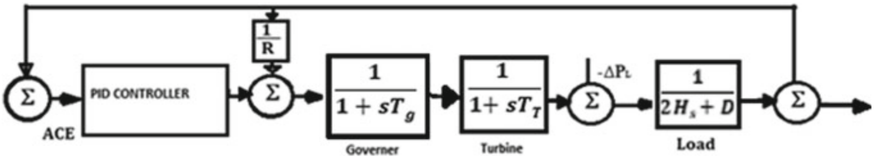


Fig. 2 Transfer function of load frequency control model of power system

changes in load demand. Controlling the power generation makes it feasible to achieve synchronizeds power balance. By employing the appropriate controllers and voltage for balancing reactive power and frequency to manage actual power balance within pre-restricted values, the impacts of the irregular load demand may be negated. By balancing the frequency constant, the LFC assists in controlling the power flow in the power system. LFC controls two control loops, a primary loop and a secondary loop, which regulate system frequency and megawatts of power.

The block diagram for the power system's load frequency control model is shown in Fig. 1. Figure 2 displays the load frequency control model's transfer function.

$$\boldsymbol{T_g} = \text{Governor time.}$$
$$\boldsymbol{T_t} = \text{Integral time.}$$
$$\boldsymbol{T_d} = \text{Derivative time.}$$

2.2 PID Controller

Figure 3 illustrates the PID controller's block diagram. The PID controller receives its input from the error, $e(t)$. The inaccuracy arises from the creation of set value input power and sample value output. The PID controller's job is to generate the right control signal to lower the error.

The derivative gain 'k_d' is used to reduce the overshoot.

The transfer function equation of proportional gain 'k_p' is $k_p * e(t)$.

The transfer function equation of integral gain 'k_i' is, $k_i * \int_0^t e(t)\mathrm{d}t$.

The transfer function equation of derivative gain 'k_d' is $k_d * \frac{\mathrm{d}e(t)}{\mathrm{d}t}$.

The PID controller transfer function is

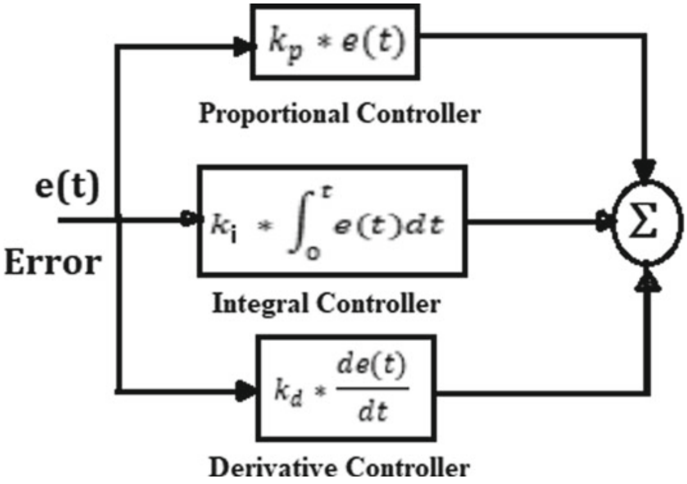


Fig. 3 Proportional, integral, and derivative (PID) controller block diagram

$$G_{\mathrm{PID}}(s) = k_p + k_i * \frac{1}{s} + k_d * s.$$

The following formula provides an explanation of the PID controller's continuous form:

$$u(t) = k_p e(t) + k_i \int_0^t e(t)\mathrm{d}t + k_d \frac{\mathrm{d}e(t)}{\mathrm{d}t}.$$

The proportional gain constant K_p is used to control the system signal to optimize the error of the system. As a result, the system time constant, and steady state errors are reduced. The output response and desired response are offset as a result of them. Increases in proportional gain 'k_p' can be used to decrease offset but do so at the expense of oscillations, which are easy to control but often induce steady-state inaccuracy.

The offset error is eliminated with the integral gain '*ki,*' but overshoot is still there. With the integral controller, steady-state error is decreased to zero. However, the system's reaction is also erratic, sluggish, and occasionally unreliable. For high-order systems with identically large time constants, the integral controller works well. It is quite slow to react yet does not exhibit steady-state inaccuracy.

By utilizing 'P controller' and 'I controller,' PI controller offers a quick reaction and zero steady-state error. The transient responsiveness is enhanced, and the steady-state error may be reduced to zero by employing PI controller. Offset is unaffected by a PI controller's significantly quicker response time than a 'I' controller's action.

Because the steady-state error will never be zero, PD controllers are useless. However, the PD controller enhances the stability of the closed-loop system for higher order operations. Fast response, zero steady-state error, and low offset are all characteristics of the PID controller that can approach the required performances. Although there is a modest demand for PD controllers, they are applicable everywhere and may be utilized in any system. When controlling systems with long time constants, PD controllers are effective. With respect to the action of the P controller, it results in the least offset and a significantly quicker reaction.

PID controllers have found widespread applicability across all fields. However, because adjusting a PID controller is challenging, sluggish factors like pH, temperature, etc., are usually under control. As a result of the foregoing analysis, PI controllers assist in tracking the discrepancy between a sample of the process output value and a pre-set point control signal value. As a result, tweaking is difficult in all applications.

With the integral time-multiplied absolute error criteria, it is possible to obtain the best result from a typical PID controller while reducing error and providing quick response (ITAE). The benefits of ITAE include producing fewer oscillations, smaller overshoots, and less complexity. Sensitive and selectivity characteristics that provide the desired output are additional benefits of the ITAE index.

2.3 Evaluation Assessment

To minimize error and manage controller tuning, the ITAE approximation model technique is used. The function for the best outcomes is thought to be ITAE. For the PID control system, the integral of time multiplied by the absolute value of error (ITAE) serves as the evaluation function.

The well-defined integral of time absolute error [ITAE] is

$$\text{ITAE} = \int\limits_{0}^{\infty} t|e(t)|\mathrm{d}t.$$

PID controllers are tuned for initial error for a step response using the time '*t*', which is huge. The ITAE value is used to assess the various metrics for gauging the responsiveness of the control system. Prior to the optimization, an evaluation function must be chosen. Compared to other evaluation methods, ITAE can examine the issues with more competence. It has been established that the PID control system's ITAE evaluation function is superior. In this study, the evaluation function of a PID control system is chosen to be the double measurement value ITAE. The ITAE is represented as

$$\text{ITAE} = \sum_{\boldsymbol{k}=0}^{\boldsymbol{m}} \boldsymbol{kT}|e(k)|T,$$

$$\mathrm{ITAE} = \int\limits_{0}^{\infty} t|\Delta f| + |\Delta P_L|\mathrm{d}t,$$

where 'Δf' are frequency deviation, 'ΔP_L' is change in MW of power.

2.4 Tuning of Controller

The controller's method and the parameters they acquired meet the required performance standards. With the aid of well-set parameters, the reaction that is achieved is extremely quick and stable. The strategies based on optimization are used to achieve the fine-tuning.

2.5 Traditional Trial-and-Error Method of Tuning

The fine-tuning point is established using the conventional trial-and-error tuning procedure. The replies to steady state with error zero can be solved using this approach, but it is exceedingly challenging. Since setting the parameter settings and ongoing tuning, both take a lot of time. It takes a lot of time, requires efforts, and uses a monotonous approach since it requires continual monitoring for the answer output. The PID controller must be used in a trial-and-error fashion to establish and achieve the ideal value. There is no guarantee that the outcomes will be optimal. If optimization is not achieved, the entire case is simple. All of these difficulties push people to engage in more sophisticated techniques in order to get ideal results.

2.6 Advanced Techniques for Optimization

Future usage of improved methods is encouraged by an understanding of the efforts necessary and required with quick time of operation to fine tune to reach the controller's optimal settings. By studying and assessing the particle swarm optimization (PSO) algorithm with ITAE method used for the given problem of reducing the frequency variation for steady-state responses with zero error, numerous optimization processes such as neural network, fuzzy logic and ANFIS, particle swarm optimization (PSO) algorithm, and artificial intelligence, are anticipated.

2.7 Particle Swarm Optimization (PSO)

This approach is based on a computational intelligence method that employs random variable selection to explore the parameter. This method is based on the mobility and intelligence of swarms. This strategy is population-based and relies on the behavior of swarms.

Typically, this strategy locates the quickest route for their search for food. An algorithm is developed to produce this behavior. Each of the many particles it uses is treated as a point in N-dimensional space. Each particle is always rushing to find space in pursuit of a meaningful answer as the highest value of the collected swarm. Every particle in the hunt of a certain path concentrates, and by dealing the bird to that path, it is clearly described by the notion of social interface.

3 Case Study

The following factors should be taken into consideration for a power system with an isolated power plant. The following data are taken into account for the power plant. Turbine rated power is 250 MW nominal frequency; governor time constants are T_g = 0.2 s, T_t = 0.5 s, generator inertia constants are $H = 5$ s, and governor speed control is $R = 0.05$ per unit. There is an abrupt 50 MW load shift with a ΔP_L of 0.2 per unit.

The case study work may be done theoretically and through MATLAB simulation, and it can be analyzed with load variations, the PID parameters fixed for a quicker steady-state time response, and the error for the dynamic response characteristics reduced. For fine-tuning with speedier results, the steps of a step-by-step way to PID tuning and a step-by-step method to PID-PSO-ITAE tuning were followed.

3.1 A Step-by-Step Method to PID Tuning

(1) Decide how many PID loops to use. Choose the tuning order if PID loop > One loop.

(2) Within the range of 1.0 and 5.0, it is advised to initialize the controller parameters k_p, k_i, and k_d using a random population. PID controllers have three linked controller parameters; therefore, the population size k_p, k_i, and k_d (100 * 3) is taken into account.

(3) Determine the fastest tune-acting loop. The fastest loops are the one where the process variable (PV) changes fastest in response to a set point (SP) change.

(4) With cascade loops: First tune the inner loop and then tune the outer loop.

(5) To tune the current PID controllers and then the speed PID controller.

3.2 A Step-by-Step Method to PID-PSO-ITAE Tuning

(1) Select the algorithm to simulate in MATLAB. This detail describes the tuning method for standard algorithms.
(2) The integral time absolute error (ITAE) technique is used to assess the fitness values of the system.
(3) Reduce (fitness function) ITAE to a minimum.

$$= \text{minimization of} \int_{0}^{t} t = k_p * |\text{ACE}| + \frac{k_i}{s} * |\text{ACE}| + s * k_d * \mathrm{d}t,$$

where $|\text{ACE}| = |\Delta f| + \Delta P_L$,
$t =$ total MATLAB simulation time.

(4) Ascertain the best values for the selected population size with use of above equation minimize (fitness function) of ITAE.
(5) Replace the outdated k_p, k_i, and k_d controller parameters with the new ones until the required number of iterations (100), at which point the updated k_p, k_i, and k_d controller parameter set is chosen as the best optimized set of controller parameters.

3.3 Working behaviour of the PID-PSO-ITAE controller

Figure 4 gives the working performance of the PID-PSO-ITAE controller.

4 Result

The case study power plant's LFC was considered and simulated using MATLAB programming and the chosen power plant's Simulink model. MATLAB software was used to assist design the system. The effectiveness of the planned coordinated strategy is linked to a few different methods.

4.1 Model of LFC for the Variation of Load with PID Control of the Power Plant

Figure 5 depicts the LFC simulation model for the power plant's PID load-variation control. The simulation is used to create the graph. This is depicted as follows.

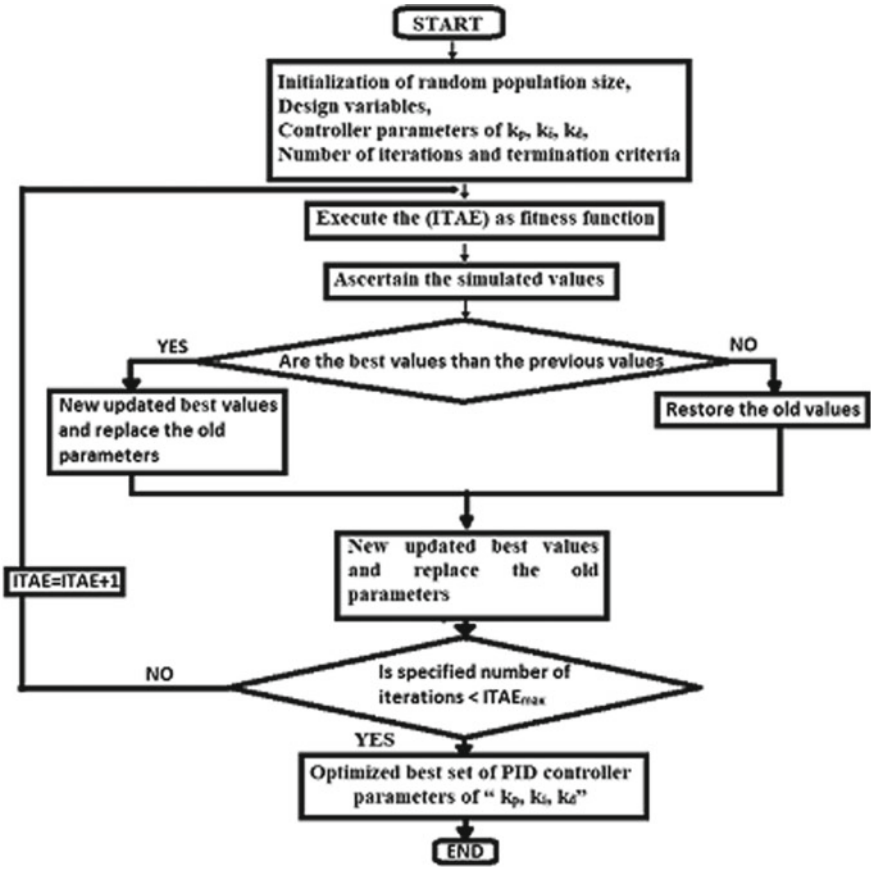

START
Initialization of random population size,
Design variables,
Controller parameters of k_p, k_i, k_d,
Number of iterations and termination criteria
Execute the (ITAE) as fitness function
Ascertain the simulated values
YES
Are the best values than the previous values
NO
New updated best values and replace the old parameters
Restore the old values
New updated best values and replace the old parameters
ITAE=ITAE+1
NO
Is specified number of iterations < $ITAE_{max}$
YES
Optimized best set of PID controller parameters of " k_p, k_i, k_d"
END

Fig. 4 Represents the step by step working process of the PID-PSO-ITAE controller

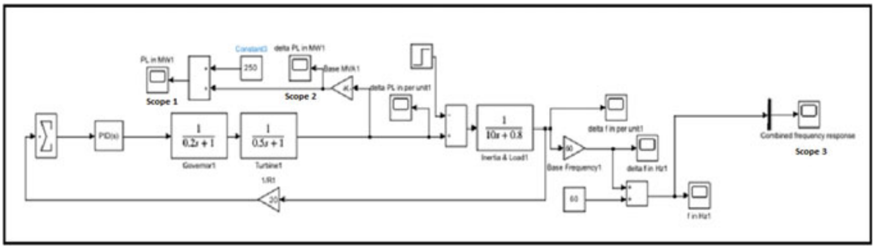


Fig. 5 Simulation model of LFC for the variation of load with PID control of the power plant

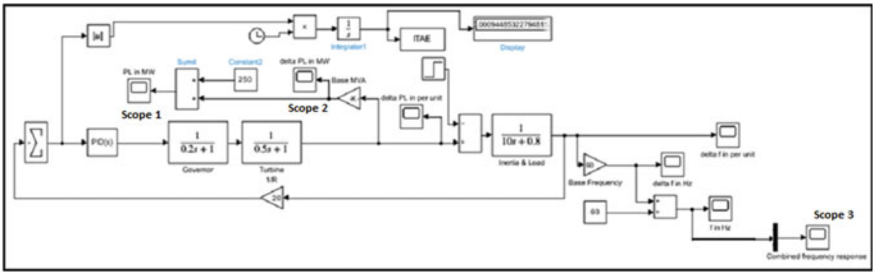


Fig. 6 Simulation model of LFC for the variation of load with PID-PSO-ITAE control of the power plant

4.2 Model of LFC for the Variation of Load with PID-PSO-ITAE Control of the Power Plant

Figure 6 depicts the LFC simulation model for the power plant's PID-PSO-ITAE load-variation control. The simulation is used to create the graph. The Simulink model was created to replicate LFC of step-time response under PID-PSO-ITAE control, therefore lowering error, speeding up response, and shortening steady-state settling time. Figure depicts the LFC simulation model with integral absolute error requirements. The simulation is used to create the graph.

4.3 Combined Model of LFC for the Variation of Load with PID and PID-PSO-ITAE Control of the Power Plant

Figure 7 depicts the combined LFC simulation model for the power plant's PID and PID-PSO-ITAE load-variation control. The graph is created using simulation using the Simulink model and analysis of PSO and ITAE using MATLAB technique. The Simulink model was created to replicate LFC of step-time response using a PID-PSO-ITAE controller in order to lower error, have a quick reaction, and have a quick steady-state settling time.

4.4 Response of Variation of Load in MW of LFC of Power Plant Using PID Controllers

Two graphs will be displayed in the scope. One will be the output of frequency, and the other will be the output of power from a load that is controlled by a PID controller. In Scope 1 and Scope 2, the graph will be presented, correspondingly.

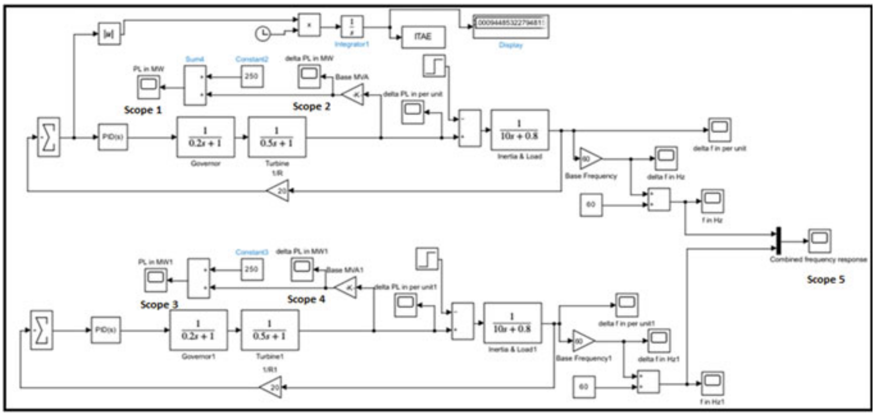


Fig. 7 Combined simulation model of LFC for the variation of load with PID and PID-PSO-ITAE control of the power plant

The simulation model of LFC for the change of load with PID management of the power plant is shown in Fig. 5. The graph for the fluctuation of load in MW rises from 250 to 300 MW using PID controllers is derived from the simulation model and is displayed below.

As according to Fig. 8, the simulation provides a graph of the fluctuation in load for the power plant's LFC using only PID controls. The overshoot with a 50 MW load rise from 0.5 to 4 s can be seen in Graph 8 from Scope 3. With a shift in the load from 250 to 300 MW, LFC aids in frequency management to restore regular operation.

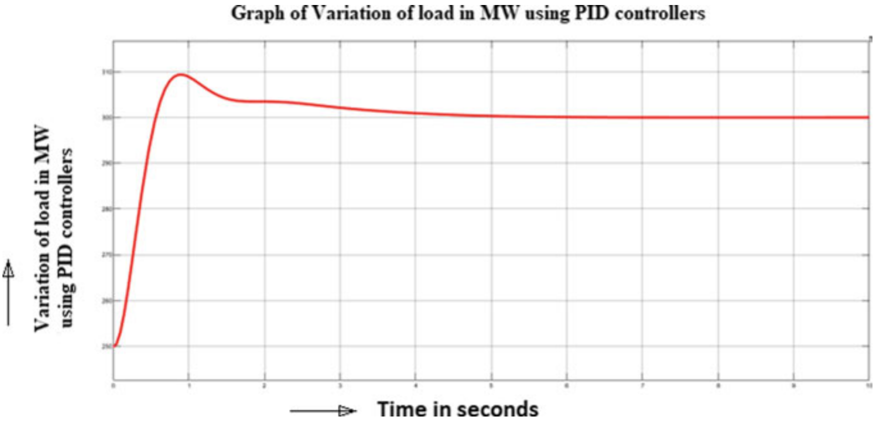


Fig. 8 Graph of variation of load in MW of LFC of power plant using PID controllers

4.5 Response of Variation of Load in MW Using PID-PSO Controllers

From the siumulation results, the proposed controller gives very good dynamic response at diffrent load conditions.

Figure 9 clarifies simulation which explains the variation of load in MW of LFC of power plant using PID-PSO-ITAE controllers only. The graph shows Scope 1 shows overshoot with the increase in load of 50 MW from zero second to 2 s reach steady state after 2 s. LFC helps to control frequency through PID-PSO-ITAE controllers to bring back to normal with the variation of load from 250 to 300 MW and takes constant load after 2 s.

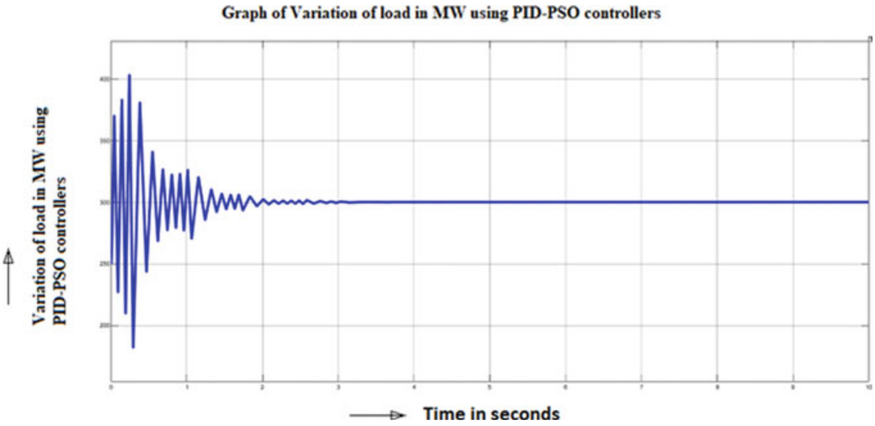
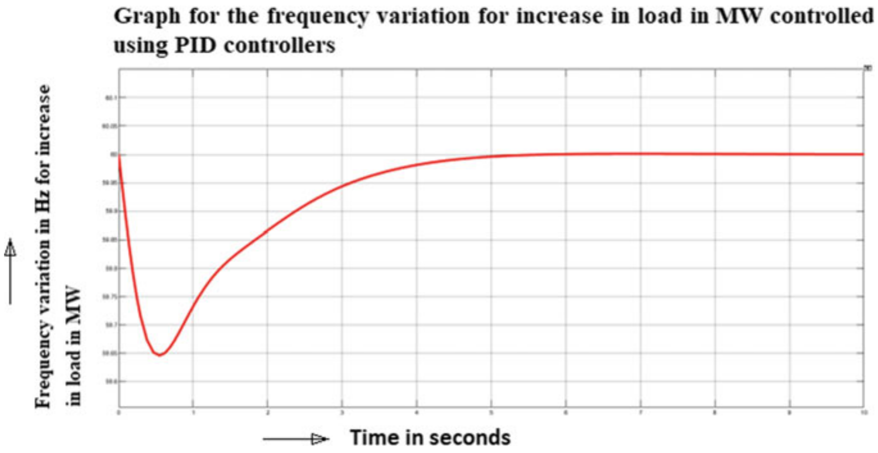
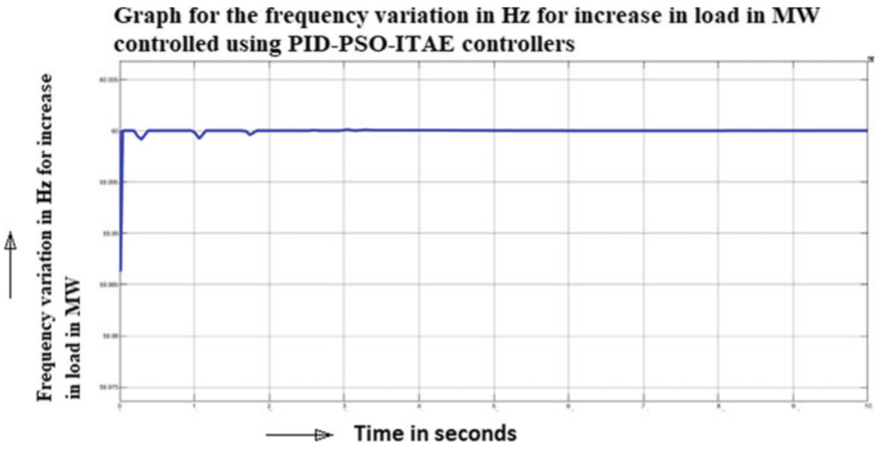


Fig. 9 Graph of variation of load in MW of LFC of power plant using PID-PSO controllers

4.6 Frequency Variation in Hz for Increase in Load in MW Controlled Using PID Controllers

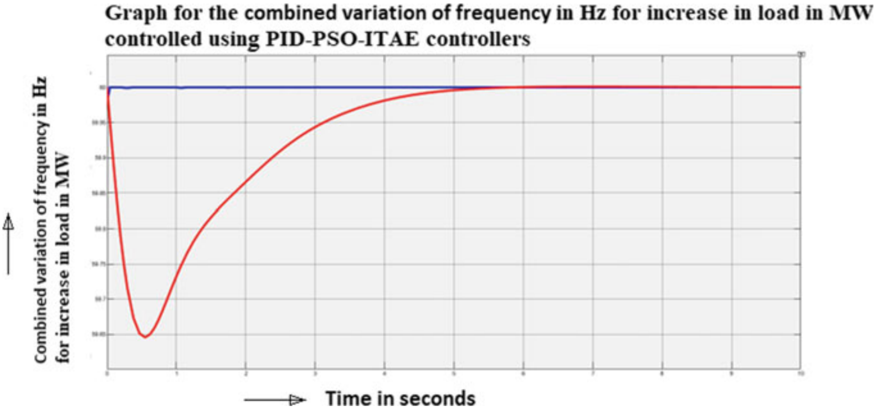


4.7 Graph for the Frequency Variation in Hz for Increase in Load in MW Controlled Using PID-PSO-ITAE Controllers



The power plant system error (ACE) is significantly decreased by the PID-PSO-ITAE controller's interaction. LFC with PID by itself is unable to supply the system frequency within the allowable bounds (i.e., 50 ± 0.2 Hz). The simulation findings demonstrate that the coordinated PID-PSO-ITAE controller strategy, which is recommended for secondary frequency control, significantly decreases the system error (ACE) when compared to primary frequency control.

4.8 Graph for the Combined Variation of Frequency in Hz for Increase in Load in MW Controlled Using PID-PSO-ITAE Controllers



Both the plots provided by the PID and PID-PSO-ITAE controller combined with PID-PSO-ITAE controllers are employed to alter frequency in Hz in response to a rise in MW consumption. The PID-PSO-ITAE controller's interaction significantly lowers the power plant system error (ACE). LFC with PID cannot control the system frequency within the acceptable limit (i.e., 50 0.2 Hz) by itself. The proposed controller reduces the system error extensively when compared to the primary frequency controller.

5 Conclusion

The ITAE empirical approach shown is used to LFC of a power plant analysis using provided data. With this method, the dynamic system achieves the best controller improvements. The frequency response of a specific power plant research was constructed using an ideal LFC strategy that was synchronized with PID-PSO-ITAE in this study. In order to improve the PID controller both for power output regulation and frequency control of the power plant study, a recently developed particle swarm optimization intelligent algorithm is used. To illustrate the efficacy of the suggested technique, the effect of control design was investigated. This was accomplished through with a comparison analysis of earlier and much more modern advanced optimization algorithms. From the simulation results, the proposed controller gives good dynamic response of the system with less load fluctuations.

References

Awouda AEA et al (2010) New PID tuning rule using ITAE criteria. Int J Eng (IJE) 3(6)

Bahgaat NK et al (2014) Load frequency control in power system via improving PID controller based on particle swarm optimization and ANFIS techniques. Int J Syst Dyn Appl: 1–24.https://doi.org/10.4018/ijsda.2014070101

Cam E et al (2005) Load frequency control in two area power systems using fuzzy logic controller. In: Energy conversion and management. Elsevier, pp 233–243.https://doi.org/10.1016/j.enconman

Gautam SK et al (2010) Improved particle swarm optimization based load frequency control in a single area power system. In: Annual IEEE India conference (INDICON), 978-1-4244-9074-5

Kushwaha K et al (2016) Optimization of load frequency control by simulated annealing applying IAE and ITAE criteria. IJSDR 1(6). ISSN: 2455-2631

Law YW (2015) Security games for risk minimization in automatic generation control

Musonda G et al (2018) Load frequency control optimization using PSO (a case of the power system in Zambia). Int J Eng Res Technol (IJERT) 7(06). ISSN: 2278-018

Narendra Kumar Jena NCP (2020) Application of fractional order cascaded controller. Int J Renew Energy Res

Nagendra M et al (2020) Automatic generation control of two area power system with hybrid control technique. Int J Eng Res Technol (IJERT) 9(09). ISSN: 2278-0181

Singh MB (2019) Design and application of PID-PID dual loop controller for load frequency control

Singh R et al (2019) Comparison of automatic load frequency control in two area power systems using PSO algorithm based PID controller and conventional PID controller. Int Conf Appl Phys Power Mater Sci. https://doi.org/10.1088/1742-6596/1172/1/012054

Efficacy of Papworth Method to Improve Quality of Life and Exercise Tolerance in Asthma

Neha S. Patil and T. Poovishnu Devi

1 Introduction

Chronic inflammatory asthma affects the airways and causes episodes of wheezing, shortness of breath, chest tightness, and coughing. These episodes typically occur at night or in the early morning. Additionally, it manifests as temporary and reversible airway obstruction, bronchial hyper-responsiveness, and airway inflammation. These symptoms are typically accompanied by a generalized but varied airway obstruction. Variations in exposures, such as allergens, airway irritants, or respiratory illnesses, are to blame for this variation in symptoms (Wu et al. 2019). According to the current Global Burden of Disease (GBD, 1990–2019), the overall burden of asthma in India is 34.3 million, accounting for 13.09% of the global burden. Additionally, it mentions that 13.2 deaths per 1000 people were attributable to asthma in India (Singh et al. 2022). Additionally, 5–10% of people globally have asthma (Pourdowlat et al. 2019). Health-related characteristic that are physical, mental, emotional, and social components of a person's illness or medical condition are linked. Asthma symptoms leads to impaired quality of life and daily-life physical activity due to its asthma. It will disturb daily living by interfering with ability to work efficiently, sleep cycle, being socially active, and importantly participating in activities due to their concern or fear of triggering asthma. Ultimately, this will diminish functional ability and quality of life as well. People with moderate to severe chronic asthma exhibit reversible airway bronchoconstriction. Asthma individuals show fear of triggering symptoms

Purpose: To find the efficacy of Papworth method to improve quality of life and exercise tolerance in asthma.

N. S. Patil (✉)
Krishna College of Physiotherapy, KIMSDU, Karad, India
e-mail: nehapatil2960@gmail.com

T. Poovishnu Devi
Department of Cardiopulmonary Physiotherapy, KIMSDU, Karad, India

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_25

and have negative attitude toward exercise and inadequate health professional advice in these patients. This will limit their physical activities and reduced physical fitness. Asthma individuals with airflow obstruction have breathlessness on exertion and reduced exercise tolerance. In case of bronchial asthma, depending upon severity of disease, exercise performance is limited. If physical activity is heavy enough, then individuals will develop an acute exacerbation of asthma (McFadden 1984).

In addition to this individual with severe pre-existing airflow obstruction, as exercise continues, expiratory flow limitation will get worsen, leads to further decreasing expiratory time, and so increases in the work of breathing. Some asthmatic individuals have difficulty in increasing ventilation sufficiently and remove carbon dioxide load generated by breathing muscles. In 40–90% of asthma individuals, exercise-induced bronchospasm (EIB) occurs and reduces exercise tolerance, which causes individuals to adopt sedentary lifestyle leading to decreased quality of life. Reduced activity due to breathlessness will lead to peripheral muscle deconditioning, which is one of the important factors in limiting exercise capacity (Sanz-Santiago et al. 2020). The inability to engage in the degree of physical activity can worsen quality of life in people with chronic lung disorders, and the relationship between activity restriction and quality of life in people with asthma is evident (Gazzotti et al. 2013). In between 60 and 75% of people with uncontrolled asthma, this restriction on physical activity is found, which has an adverse effect on employment and daily activities. Physical activity intensity and the likelihood of an exacerbation occurring are adversely correlated (Vermeulen et al. 2016). Respiratory symptoms brought on by exercise have a significant impact on daily life. The main factor contributing to an increased lung volume in asthma is a prolonged inspiratory muscle activity. The respiratory muscles' ability and strength may decrease because of this hyperinflation (Thomas et al. 2009). As asthma is chronic condition which interferes with quality of life and exercise tolerance, so it is important to find out cost-effective alternative for the management of asthma due to increased prevalence and rising health care cost as well (Zampogna et al. 2020). Therefore, to provide comprehensive asthma treatment, it is necessary to rule out and address the factor affecting quality of life and exercise tolerance. According to GINA 2019, asthmatics require stepwise approach which contains assessment treatment planning and review of response. Along with this modification of risk factors, controller therapy is adjusted accordingly. But this is inadequate for asthma management as pharmacological management has some side effects. So, in addition to medical care, Papworth method provides good adjuvant to control asthma symptoms. Asthma subjects require comprehensive approach to deal with these hyperinflation and hyperventilation which are common manifestations of asthma.

Papworth method is adjuvant to medical care in managing asthma symptoms. It focuses on dysfunctional breathing including hyperinflation and hyperventilation. Papworth method contains five components. First breathing training, which teaches the proper minute and tidal volumes in accordance with present metabolic activity and the creation of a suitable breathing pattern. Specific Papworth technique diaphragmatic breathing is taught to reduce unwarranted usage of supplementary breathing muscles. In this strategy, we advise people with asthma to breathe via their nose rather than their mouth. Additionally, it aids in the eradication of learned and practiced habits like yawning and signing. The second is education, which emphasizes the physical regulation of stress responses, their bodily detection, and particularly how they interact with breathing patterns. Training in both specific and general relaxation is the third component. The incorporation of "proper" breathing and relaxation techniques into daily activity is the fourth component. Techniques are taught in a sequential fashion, starting with a semi-recumbent position and moving to sitting, standing, and eventually doing daily tasks. The skills are initially taught in a semi-recumbent position, then on to sitting, then standing, and finally while performing daily tasks. Finally, teaching and practicing breathing and relaxation techniques in speech. Fifth component is home exercise. Encouragement is given to practice two times a day. Papworth method is emerging as a low-cost alternative and found to be effective in management of asthma which is a chronic condition and helps to improve exacerbations, dyspnea, and dysfunctional breathing (Holloway and West 2007). Asthma symptoms are not managed by medications alone, so Papworth method is easy and safe intervention for asthma control as it is also an adjuvant to medical care. Papworth method reduces these asthma symptoms, improves dysfunctional breathing arising from hypocapnia, and will help to improve quality of life and exercise tolerance which is measured using asthma-related quality of life questionnaire (ICC score 0.95) and 6-min walk test, respectively (Juniper et al. 1993). Adding Papworth method intervention to standard medical care to the asthma subjects is the primary objective of this study (Pourdowlat et al. 2019).

2 Materials and Methodology

This study's experimental study design has a 70-person sample size. For six months, this study is being done at the KIMS hospital in Karad. Simple random sample is the study's sampling strategy (Lottery method). Patients with mild-moderate asthma, age range of 20–45 years, either sex, use of bronchodilators, or no addictions were required for inclusion in the study. Patients with severe asthma who needed emergency medical attention, any illnesses that simulated asthma and its symptoms, and chronic illnesses such as cancer, anemia, or cardiovascular conditions; recent patients who have undergone cardiothoracic or thoracic surgery; and any unforeseen illness or disability that would prevent participation in exercise testing or training or in an intervention program within the previous 12 weeks.

3 Procedure

The protocol committee for the project gave its approval, and the institutional ethical committee of the Krishna Institute of Medical Sciences, a deemed university in Karad, granted its approval as well. Between August 2021 and July 2022, patients who visited the doctors for routine clinic visits were screened and recruited. Patients were screened using inclusion and exclusion criteria. The research plan randomly assigned the patients who consented to participate to the experimental group and control group. The eligible patients were divided into two groups at random. All patients gave their informed consent. Each participant answered the research questionnaires after receiving their assignment and gave physiological data. At every regular clinic appointment, data were gathered, which took around 60 min. Outcome measures were used for the pre-test. The clinic's case managers provided routine care and instruction to all the patients. For the length of 6 months, group A received regular medical care and Group B received both standard medical care and the Papworth method intervention.

Participants were encouraged to identify relaxed strategies of overcoming the hurdles to minimize asthma symptoms. Written materials were given to each patient to urge them to follow their intervention plan. The intervention program was encouraged for participants to complete. The outcome measures were assessed at the beginning, three and six months afterward. Then, statistical analysis was completed.

4 Data Analysis

Interpretation: The post-mean values of groups A and B of the AQLQ are displayed in this graph (Fig. 1 and Table 1).

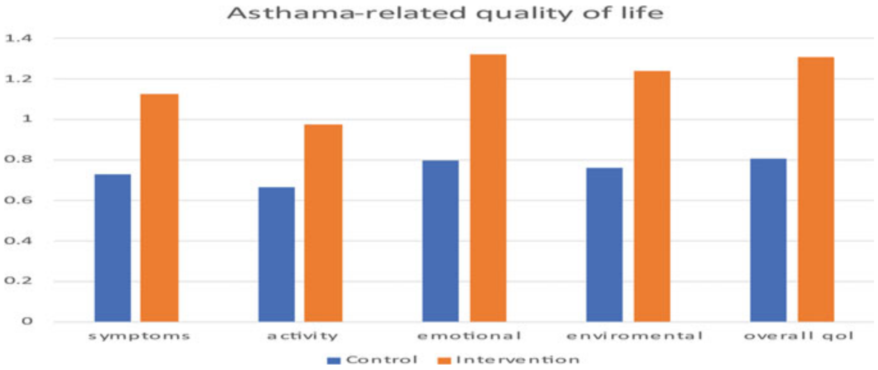


Fig. 1 Asthma- related quality of life

Table 1 Comparison within groups of mean (95% CI) of change in asthma-related quality of life questionnaire (AQLQ) score and 6MWT from baseline readings and after 6 months of intervention period between Papworth methods that is intervention group and control group

Domains	Intervention group	Control group	*T* test	*P* value
AQLQ				
Overall QOL	1.30(1.2–1.55)	0.80(0.55–1.03)	19.69	0.0001
Symptoms	1.27(0.95–1.60)	0.75(0.53–0.97)	12.37	0.0085
Activity	0.99(0.78–1.20)	0.71(0.45–0.98)	11.29	0.0564
Emotional	1.35(1.08–1.65)	0.91(0.57–1.25)	14.46	0.0110
Environmental	1.20(0.80–1.60)	0.86(0.60–1.12)	12.28	0.0005
6MWT	615 ± 122	494 ± 102	10.732	0.0056

5 Discussion

In this study, the effectiveness of the Papworth approach to enhance asthmatic patients' quality of life and exercise tolerance will be evaluated. At the 6-month assessment, it was discovered that the Papworth approach was linked to enhancements in the primary outcome (asthma-related quality of life) over those observed in the control group. The Papworth technique for breathing training had ideal effects on exercise tolerance and the improvement of asthma sufferers' quality of life, according to the current study. This is consistent with a 2019 study by Guitti Pourdowlat, Roghyeh Hejrati, and Somayeh LookZadeh Holloway et al. that employed the Papworth method of relaxation in 30 asthma patients and used the SF36 questionnaire to evaluate quality of life. They claimed that practicing relaxation using the Papworth approach. They stated that relaxation training by the Papworth approach sounded to be able to control stressful settings in individuals with asthma to prevent illness attacks and improve the quality of life (Pourdowlat et al. 2019). As in Guitti Pourdowlat's study, we also evaluated participants' quality of life (QOL) and exercise tolerance, as well as the impact of the Papworth approach on these variables. In our study, quality of life also increased. Another earlier study conducted by Elizabeth A. Holloway and Robert J. West and published in 2007 shows that the Papworth method appears to improve respiratory symptoms and dysfunctional breathing compared to usual care, but that there was no significant change in lung function, suggesting that the Papworth method does not improve the long-term physiological underpinnings of asthma but rather its manifestations. Asthma-related quality of life was chosen as the study's main outcome because it is the metric that most accurately captures how patients feel about their asthma and how it affects their everyday lives. The Papworth technique was consistently linked to improvement in patient-centered variables such symptoms, activity, emotional, and environmental measurements in this study where we examined a variety of outcome parameters.

The 6MWT parameters and the asthma-related quality of life questionnaire (AQLQ) were the study's outcome measures. The intervention was carried out by the team of specialists involved to enhance outcome metrics. The Papworth technique has enhanced quality of life and exercise tolerance as benefits. Such a strategy should be used to raise the quality of life and increase exercise tolerance in people with asthma.

6 Conclusion

Improve the quality of life and exercise tolerance in mild to moderate asthma by using the Papworth technique as an adjunctive beneficial treatment.

7 Limitation

The study population may have been biased as a result of the limited enrollment, which is a possibility. Another potential drawback is the lack of formal monitoring of participants randomly assigned to the control group for any changes in their exercise routine.

References

Bruurs ML, van der Giessen LJ, Moed H (2013) The effectiveness of physiotherapy in patients with asthma: a systematic review of the literature. Respir Med 107(4):483–494. https://doi.org/10.1016/j.rmed.2012.12.017. Epub 18 Jan 2013. PMID: 23333065

Gazzotti MR, Nascimento OA, Montealegre F, Fish J, Jardim JR (2013) Level of asthma control and its impact on activities of daily living in asthma patients in Brazil. J Bras Pneumol 39:532–538

Grygus I (2017) The role of physical activity in the rehabilitation of patients suffering from mild persistent bronchial asthma. Phys Educ Sport Health Culture Mod Soc 2(38):140–150

Holloway EA, West RJ (2007) Integrated breathing and relaxation training (the Papworth method) for adults with asthma in primary care: a randomised controlled trial. Thorax 62(12):1039–1042

Juniper EF, Guyatt GH, Ferrie PJ, Griffith LE (1993) Measuring quality of life in asthma. Am Rev Respir Dis 147(4):832–838. https://doi.org/10.1164/ajrccm/147.4.832. PMID: 8466117

McFadden Jr ER (1984) Exercise performance in the asthmatic. Am Rev Respiratory Dis 129(2P2):S84–S87

Mims JW (2015) Asthma: definitions and pathophysiology. Int Forum Allergy Rhinol 5(Suppl 1):S2–S6. https://doi.org/10.1002/alr.21609. PMID: 26335832

Pourdowlat G, Hejrati R, Lookzadeh S. The effectiveness of relaxation training in the quality of life and anxiety of patients with asthma. Adv Respir Med. 87(3):146–151. https://doi.org/10.5603/ARM.2019.0024. PMID: 31282555

Sanz-Santiago V, Diez-Vega I, Santana-Sosa E, Lopez Nuevo C, Iturriaga Ramirez T, Vendrusculo FM, Donadio MV, Villa Asensi JR, Pérez-Ruiz M (2020) Effect of a combined exercise program on physical fitness, lung function, and quality of life in patients with controlled asthma and exercise symptoms: a randomized controlled trial. Pediatr Pulmonol 55(7):1608–1616

Singh S, Salvi S, Mangal DK, Singh M, Awasthi S, Mahesh PA, Kabra SK, Mohammed S, Sukumaran TU, Ghoshal AG, Barne M (2022) Prevalence, time trends and treatment practices of asthma in India: the global asthma network study. ERJ Open Res 8(2)

Thomas M, McKinley RK, Mellor S, Watkin G, Holloway E, Scullion J, Shaw DE, Wardlaw A, Price D, Pavord I (2009) Breathing exercises for asthma: a randomised controlled trial. Thorax 64(1):55–61

Vermeulen F, Garcia G, Ninane V, Laveneziana P (2016) Activity limitation and exertional dyspnea in adult asthmatic patients: What do we know? Respir Med 117:122–130. https://doi.org/10.1016/j.rmed.2016.06.003. Epub 9 June 2016. PMID: 27492522

Wu TD, Brigham EP, McCormack MC (2019) Asthma in the primary care setting. Med Clin 103(3):435–452

Zampogna E, Zappa M, Spanevello A, Visca D (2020) Pulmonary rehabilitation and asthma. Front Pharmacol 11:542. https://doi.org/10.3389/fphar.2020.00542. PMID: 32435190; PMCID: PMC7219266

CNN Approach for Plant Disease Detection—Krishi Snehi

A. Lavanya, N. Ganavi, and M. R. Sowmya

1 Introduction

In addition to being the foundation of the national economy, agriculture is essential for raising farmer income. There are many plant diseases that harm plants, but because their early symptoms are often missed, social and financial losses result. Despite the possibility that scientists may provide a more effective solution using photos and videos of crops that provide a better perspective of diseases, farmers still struggle to identify plant diseases at an early stage. India is falling behind even though its production is enormous, because there is no alternative method described in the literature that can deal with the precise identification of crop diseases. Pathogenic organisms cause most crop infections, and current research does not offer fertilizer optimization to increase crop quality.

Before a technique is introduced to identify plant infections, diagnosing plant diseases mainly relies on knowledgeable agronomists or phytopathologists. In order to address the illness before it spreads out of control, they created a checklist of requirements based on study and field observations of plants. However, even professionals with specialized optical equipment can be unable to recognize early-stage infections due to the wide diversity of infectious symptoms and significant variations in the same symptom among other species. Convolutional neural networks (CNN), an improved version of ANN, are widely employed in image classification and speech recognition due to their capacity to handle large amounts of data and create clever connections between the neurons.

Given that most farmers are members of the illiteracy group, farmers currently require advice from agriculture experts to address any such adverse condition seen in the crops in a way that is simple for them to understand. The one with the technical knowledge can really help these types of farmers in detecting the disease and

A. Lavanya · N. Ganavi · M. R. Sowmya (✉)
Department of Computer Science and Engineering, NMIT, Yelahanka, Bangalore 560064, India
e-mail: sowmyamr@gmail.com

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_26

controlling measures using Krishi-Snehi application in their own language, provided in speech format as well. Another category of farmers, who are capable of reading and writing their regional language but lack English literacy, can still use this application where they access to information in their own regional language (Kannada, Hindi).

2 Literature Review

The paper (Swathika et al. 2021) provides a reference for further research on the subject, reviews earlier work on CNN image classification, and introduces a new module for paddy disease classification. Convolutional neural network layers were used to create the classification module, which offers accuracy of up to nearly 70%.

According to Joshi and Saha (2019), for a machine to make predictions or decisions and provide outcomes without being given instructions, a set of training data and testing data is necessary. The algorithms take images, segment them to extract information from them, and then use the features to identify the ailment the plant is suffering from.

A reasonably large repository is reportedly primarily maintained in English, according to paper Cynthia et al. (2019). Most of the web content is written in English. This clearly demonstrates a barrier to the benefits of the Internet for the 64% of people around the world who do not speak English. The situation is more serious in developing nations like India, China, and Pakistan, where 76% of the population lacks literacy in English and hence deprived from the advantages of Internet facilities.

Madhulatha and Ramadevi (2020) particularly focused on detecting plant diseases that will lower crop loss and hence maximize production efficiency. Their approach uses deep learning (DL) technology to identify early-stage plant disease symptoms and classifies plants as diseased based on the symptoms using a deep learning (DL) technique.

Marzougui et al. (2020b) applied a computer approach on Deep Learning systems based on artificial neural networks, the branch that also enables the early identification of plant illnesses by using convolutional neural networks (CNNs) which are familiar with some of the prominent architectures, particularly the "ResNet" architecture. They made use of an enhanced dataset that included pictures of both healthy and sick leaves (each leaf is manually cut and placed on a uniform background). Their deep learning approach performed admirably for a variety of object detection issues. Ghosh S et al. (2014) developed an user friendly application which had a communication in Indian languages through voice. Guan (2021) created a novel method for detecting plant diseases by merging four CNN models. Sritharan et al. (2022) model identified various plant diseases and deployed a mobile technology that can make appropriate pesticide recommendations. Yadhav et al. (2020) focused on categorizing the plant illnesses, with the help of TensorFlow backend system using the convolutional neural network model. It also recommended fertilizers to increase crop yields.

3 Proposed Methodology

The primary goal is to identify whether the crop is diseased or healthy. Krishi Snehi is an application which classifies the uploaded leaf image of crops as diseased or healthy and if diseased, identifies the type of disease and provides with respective biological and chemical control measures in the user selected language and the same is available in audio format which can be controlled with audio button.

Data Collection: The code was applied to the Kaggle online kernel for training loss and validation computations, and the dataset was acquired from the online Plant Village Kaggle dataset. Dataset includes training data and testing data of 20k images, which contained 1254 images from the testing set and 19,384 images from the training set.

The stages involved in fully identifying and classifying diseases are shown in Fig. 1:

1. Input image: Images that are captured for analysis are kept in backup memory. Both the training dataset and the testing dataset were created using images that came from a reliable source.
2. Preprocessing: The perfect data for processing is provided by the image's clarity. Using a variety of smoothening filters and enhancement techniques, the contrast quality of the image is further improved after using image clipping to remove the necessary portion of the image to create a clear image.
3. Feature extraction: Four convolutional layers are used to form the convolutional neural network. These four hidden layers are used to extract features of diseases. Each hidden layer consists of the following layers:

   3.1 Convolution layer: Through convolution layer, features and relationships between pixels are learned by extracting features from the image using tiny squares of input data. Convergence of the input picture matrix and the filter matrix can be used to achieve this.

   3.2 Activation function: A mechanism of activation that makes the node active when the input value is above the threshold and if the input is below zero then output will be zero itself. There are two activation functions used: ReLu and Softmax.

   3.3 Max pooling. A pooling technique called max pooling chooses the top element from the feature map area that the filter is focusing on. As a result, a feature map with the most important features from the prior feature map would be the output after the maximum pooling level.

4. Creation of model: A total of 19,384 images were utilized to train the four convolutional layers of the convolutional neural network sand remaining 1254 images were used to test the proposed model during 20 epochs, producing an accuracy of 0.9630, i.e., 96%. The trained model was stored in "auto_chloro_model.h5" for faster access.

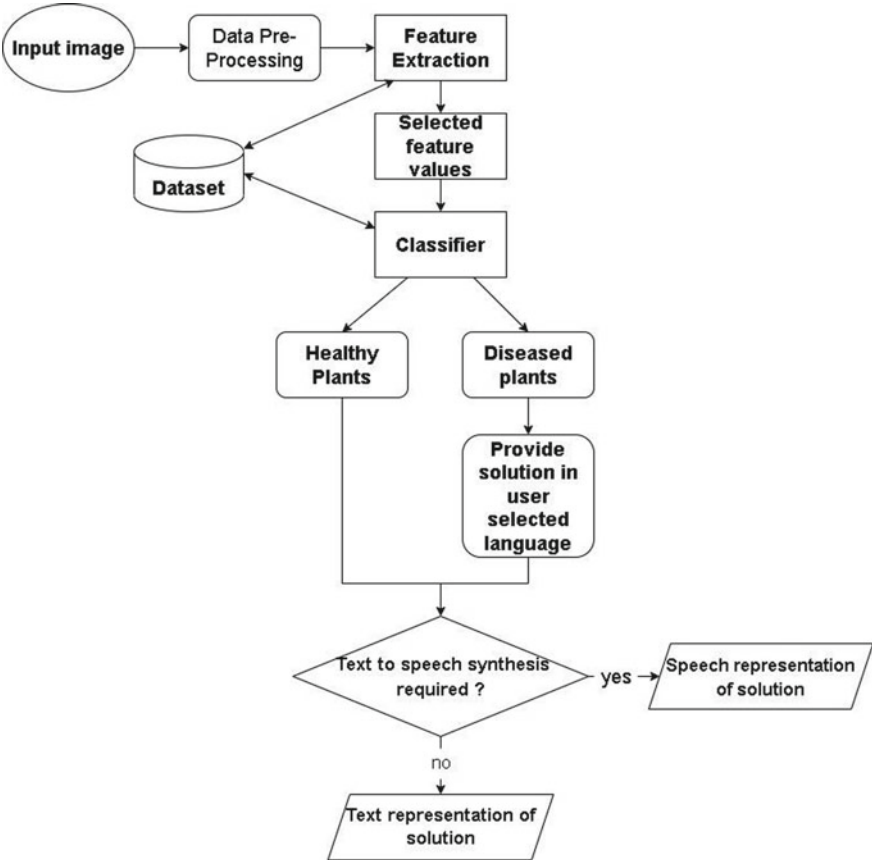
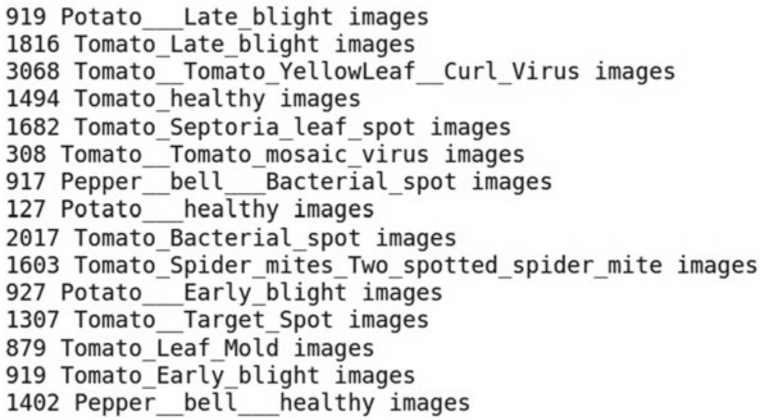

Input image
Data Pre-Processing
Feature Extraction
Selected feature values
Dataset
Classifier
Healthy Plants
Diseased plants
Provide solution in user selected language
Text to speech synthesis required ?
yes
Speech representation of solution
no
Text representation of solution

Fig. 1 Workflow of the plant disease detection

5. Classifier: Two fully connected dense layers with the proposed activation function for classifying plants as healthy or diseased are used after a fully connected layer to reduce the image's three dimensions to one in order to determine the probabilistic value.

The user can get to know the results in his preferred language (English, Kannada, and Hindi) and can even access it in audio format with the aid of the audio control button provided (Fig. 2).

Connection of React App with the Built CNN model: The built and trained CNN model is connected to a React app to provide better user interface with all the responsive buttons to upload images and to get the results in text format or in speech if required as shown in.



```
919 Potato___Late_blight images
1816 Tomato_Late_blight images
3068 Tomato__Tomato_YellowLeaf__Curl_Virus images
1494 Tomato_healthy images
1682 Tomato_Septoria_leaf_spot images
308 Tomato__Tomato_mosaic_virus images
917 Pepper__bell___Bacterial_spot images
127 Potato___healthy images
2017 Tomato_Bacterial_spot images
1603 Tomato_Spider_mites_Two_spotted_spider_mite images
927 Potato___Early_blight images
1307 Tomato__Target_Spot images
879 Tomato_Leaf_Mold images
919 Tomato_Early_blight images
1402 Pepper__bell___healthy images
```

Fig. 2 15 classes of disease classification

4 Results

The final product is a web-based application built using React. Js that enables the users with a better interface. This application has an outcome of detected disease, corresponding to chemical and biological controls in user selected language. The proposed model gave us an accuracy of 0.9630 which is 96%.

The graph as shown in Fig. 3 clearly shows that there is an increase in accuracy and depicts that the model was well-trained.

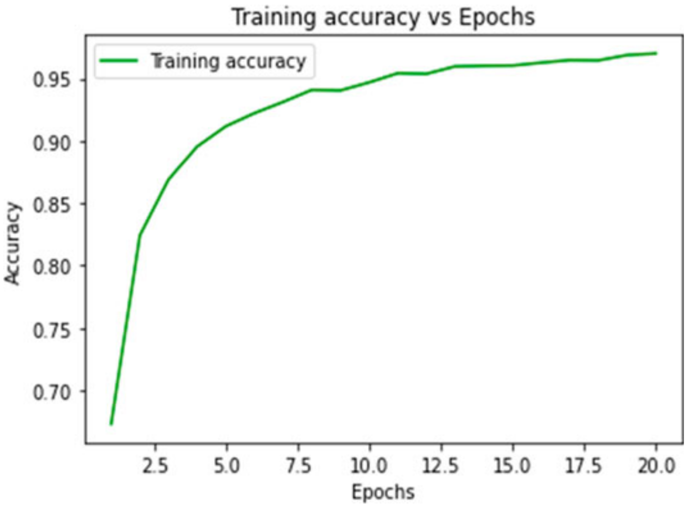


Fig. 3 Epochs versus accuracy

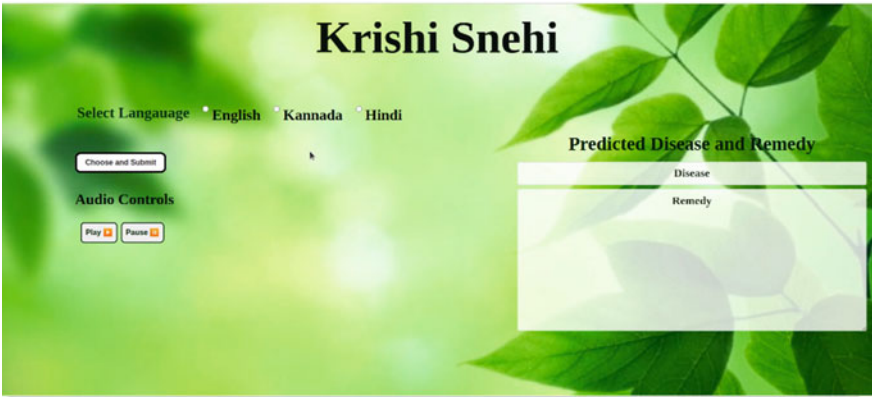
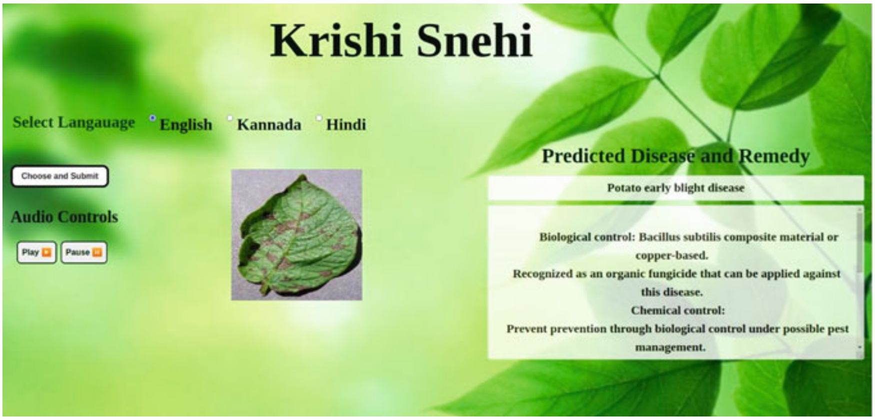


Fig. 4 Krishi-Snehi interface



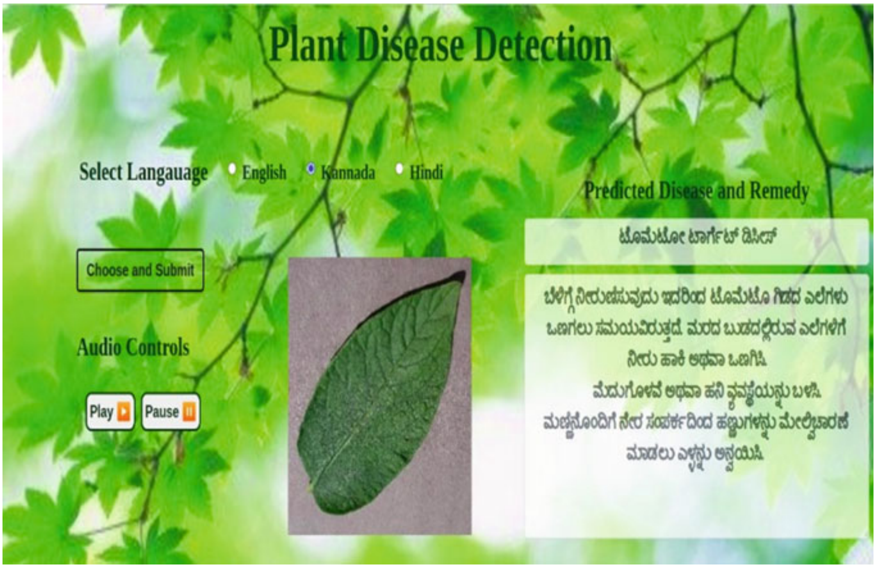
Potato early blight disease

Biological control: Bacillus subtilis composite material or copper-based.
Recognized as an organic fungicide that can be applied against this disease.
Chemical control:
Prevent prevention through biological control under possible pest management.

Fig. 5 Disease prediction and its corresponding control measures in English

The interface of the Krishi Snehi is as shown in Fig. 4 which shows the option of selecting the language and audio. The disease and the remedy is displayed after uploading the image of the leaf.

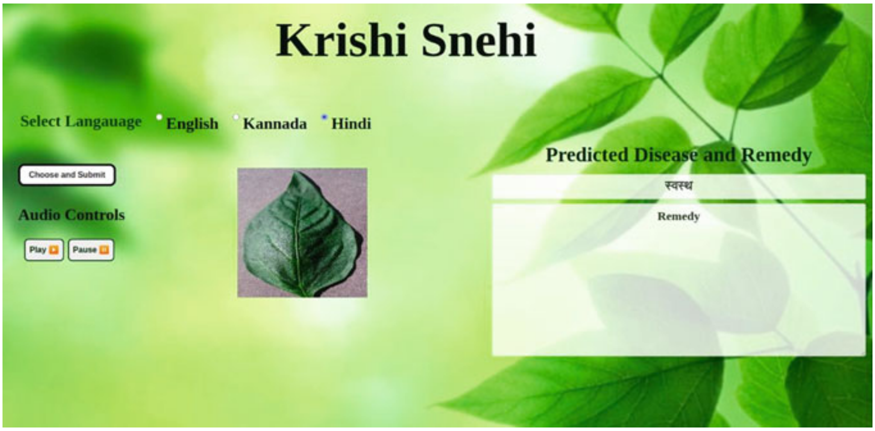
The detection process is carried out, firstly by selecting the language (English, Kannada, and Hindi) and then choosing the test image from the secondary storage. The application then displays the predicted disease and its corresponding biological and chemical controls if the image selected is of a diseased leaf as shown in Figs. 5 and 6 else displays "healthy", if the selected image of a plant is of a healthy leaf as shown in Fig. 7.



ಟೊಮೆಟೋ ಟಾರ್ಗೆಟ್ ಡಿಸೀಸ್

ಬೆಳಿಗ್ಗೆ ನೀರುಣಿಸುವುದು ಇದರಿಂದ ಟೊಮೆಟೊ ಗಿಡದ ಎಲೆಗಳು
ಒಣಗಲು ಸಮಯವಿರುತ್ತದೆ. ಮರದ ಬುಡದಲ್ಲಿರುವ ಎಲೆಗಳಿಗೆ
ನೀರು ಹಾಕಿ ಅಥವಾ ಒಣಗಿಸಿ.
ಮೆದುಗೊಳವೆ ಅಥವಾ ಹನಿ ವ್ಯವಸ್ಥೆಯನ್ನು ಬಳಸಿ.
ಮಣ್ಣಿನೊಂದಿಗೆ ನೇರ ಸಂಪರ್ಕದಿಂದ ಹಣ್ಣುಗಳನ್ನು ಮೇಲ್ವಿಚಾರಣೆ
ಮಾಡಲು ಎಳ್ಳನ್ನು ಅನ್ವಯಿಸಿ.

Fig. 6 Disease prediction and its corresponding control measures in Kannada



स्वस्थ

Remedy

Fig. 7 Output shows the chosen leaf is healthy in Hindi

5 Conclusion

Although there are plenty of image classifying algorithms, CNN has proved to be one of the most powerful image processing and classifying algorithms that yields better performance. This proposed model made use of four convolutional layers that increased the outcome of the training. This model having relatively high accuracy of 96% can be really used to assist the farmers in disease identification. For now, this model is restricted to three crops such as potato, tomato, and bell pepper with 15 different classes. Future research should try increasing the size of the datasets and increasing the number of crops for which the model can be trained.

References

Cynthia ST, Hossain KM, Hasan MN, Asaduzzaman M, Das AK (2019) Automated detection of plant diseases using image processing and faster R-CNN algorithm. In: 2019 international conference on sustainable technologies for industry 4.0 (STI). Dhaka, Bangladesh

Ghosh S, Garg AB, Sarcar S, Sridhar PS, Maleyvar O, Kapoor R (2014) Krishi-Bharati: an interface for Indian farmer. In: Proceedings of the 2014 IEEE students' technology symposium. Kharagpur

Guan X (2021) A novel method of plant leaf disease detection based on deep learning. In: 2021 6th international conference on intelligent computing and signal processing (ICSP)

Joshi PK, Saha A (2019) Detection and classification of plant diseases using soft computing techniques. In: 2019 4th international conference on recent trends on electronics, information, communication & technology (RTEICT)

Madhulatha G, Ramadevi O (2020) Recognition of plant diseases using convolutional neural network. In: 2020 fourth international conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)

Marzougui F, Elleuch M, Kherallah M (2020a) A deep CNN approach for plant disease detection. In: 21st international Arab conference on information technology (ACIT), Arab

Marzougui F, Elleuch M, Kherallah M (2020) In: 21st international Arab conference on information technology (ACIT). Arab

Santra S, Bhowmick S, Paul A, Chatterjee P, Deyasi A (2018) Development of GUI for text-to-speech recognition using natural language processing. In: 2018 2nd international conference on electronics, materials engineering & nano-technology (IEMENTech)

Shah JP, Prajapati HB, Dabhi VK (2016) A survey on detection and classification of rice plant diseases. In: 2016 IEEE international conference on current trends in advanced computing (ICCTAC). Bangalore

Sritharan L, Anjanan M, Gamage A (2022) Plant diseases detection using image processing and suggest pesticides and managements. In: 2022 IEEE 7th international conference for convergence in technology (I2CT)

Swathika R, Srinidhi S, Radha N, Sowmya K (2021) Disease identification in paddy leaves using CNN based deep learning. In: 2021 third international conference on intelligent communication technologies and virtual mobile networks (ICICV)

Yadhav SY, Senthilkumar T, Jayanthy S, Kovilpillai JJ (2020) Plant disease detection and classification using CNN model with optimized activation function. In: 2020 international conference on electronics and sustainable communication systems (ICESC)

Lung Cancer Classification and Prediction Based on Statistical Feature Selection Method Using Data Mining Techniques

S. Kavitha, N. H. Prasad, K. Sowmya, and Ramavathu Durga Prasad Naik

1 Introduction

The term data mining refers to the technique of removing or refining previously undiscovered forecasting information from massive databases. It entails various data sorting through numerous of datasets in order to explore significant and necessary information. Finding novel, intriguing, and unique sequences or patterns in stored data is a common goal of data mining. It usually discovers the sheer within the data that is beyond simple analysis using an algorithm. Data mining can be applied to both knowledge discovery and prediction processes. Data mining is an information-delivery method that transforms data access for introspection and anticipated navigation. Goals of data mining: prediction process: it makes predictions by drawing inferences from current data. Gain by using previously saved values in a form to uncover hidden or anticipated outcomes. It describes and simulates the fundamental characteristics of the data in the database with an emphasis on pattern recognition and the successive submission of user interpretation.

The term "cancer" refers to a category of diseases in which the body's cells begin to multiply and expand out of control. These cells can then invade and attempt to kill healthy tissues. The cells develop carcinogenic or malignant properties as a result of DNA damage. This harm may develop over time, may result from errors

S. Kavitha (✉) · N. H. Prasad · K. Sowmya
Department of MCA, Nitte Meenakshi Institute of Technology, Bangalore, India
e-mail: Kavitha.s@nmit.ac.in

N. H. Prasad
e-mail: hod-mca@nmit.ac.in

K. Sowmya
e-mail: sowmya.k@nmit.ac.in

R. D. P. Naik
Rajiv Gandhi University of Knowledge Technologies, Santhanuthalapadu, Andhra Pradesh, India

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_27

made during typical cell growth, or may be brought on by an outside factor, such as nicotine. Cancer cells can spread to other parts of the body, where they can multiply and form new tumours. This is referred to as metastases. It occurs when cells enter the bloodstream or lymphatic vessels.

When lung cells increase improperly and create a tumour, a lung cancer is the result. On chest X-rays or CT ('CAT') scans, this might be detected as a lump or nodule. Smokers are much more likely to develop lung cancer than non-smokers. However, it can also occur in those who have never smoked. Asbestos, second hand smoke, and radon are other lung cancer risk factors. Lung cancer occurs if family history of lung cancer, cigarette exposure, genetics and air pollution. Although receiving a lung cancer diagnosis is terrifying, treatments have greatly advanced in recent years. Generally speaking, detecting lung cancer signs early is important since persons with the disease have a better prognosis important. However, lung cancer can rarely occur. Lung cancer often arises without symptoms; hence, screening is also a crucial tool for early lung cancer detection in patients who are in danger.

Lung cancer is categorised according to the anatomic site as follows: squamous cell carcinoma and small cell carcinoma are the most common types of central lung cancer. Peripheral lung cancer primarily consists of giant cell carcinoma and adenocarcinoma. Histopathology-specific classification: small cell carcinoma, commonly known as SCLC (15–20%); squamous cell carcinoma, 10%; adenocarcinoma, 30–40%; and non-small cell lung cancer (NSCLC, 80–85%); sarcoma and lymphoma are two more malignant lung neoplasms.

The leading cause of high mortality in the present world is estimated to be lung cancer, which is also the deadly disease. Humans are more strongly affected by lung cancer than is envisaged. We can predict lung cancer through chest X-rays that frequently fail to differentiate between cancer and other illnesses, such as a lung abscess, prevent them from providing a conclusive diagnosis (collection of pus that forms in the lungs). Lung cancer also, can be predicted through CT scan uses X-rays and a computer to create detailed images of the inside the body. Cancer cells in active growth can be located using a PET-CT scan. This can aid in making the correct diagnosis and treatment decision. An operation called a bronchoscopy enables a medical professional to view the interior of your airways and take a little sample of cells.

2 Literature Review

Lung cancer is the world's leading cause of cancer death, and treatment is decided by the type and stage of cancer discovered in the patient (Lee et al. 2011). Early diagnosis of this disease is the most promising approach to improving patients' survival rates (Maleki et al. 2021). Data mining techniques, such as categorization and prediction, can be utilised in clinical care to determine the progression of disease rules from multidimensional time-series data (Fang et al. 2015). Based on two feature selection algorithms, a method for predicting the risk of lung cancer is proposed: Fisher and

ReliefF, as well as BP Neural Networks (Xie et al. 2015). The identification of divergence between oncogenic tumours is essential for the diagnosis and treatment of cancer. The goal of this study was to develop a computer method for classifying lung cancer tumours based on the structural and physicochemical characteristics (1497 attributes) of protein sequences acquired from genes identified by microarray analysis (Ramani and Jacob 2013).

Combining Tetrakis Carboxy Phenyl Porphine (TCPP) and well-known machine learning algorithms, a novel approach to the early identification of lung cancer is mentioned. Because cancer cells have more low-density lipoproteins on their surface and have a porous cancer cell membrane, the porphyrin tetrakis carboxy phenyl porphine (TCPP) can be used to identify cancer cells (Kancherla and Mukkamala 2012). Using image processing techniques, lung scan histopathology images can be utilised to categorise lung cancer. In the system for prediction, information from lung pictures are extracted (Shanthi and Rajkumar 2021). Based on the importance of lung cancer survival, the researchers gathered information from four feature categories (population, recognition, treatment, and result) of cancer patients. Analyses of the patients' predicted survival rates show that, of the classification algorithms, Decision Tree C5.0 has the highest accuracy (Dezfuly and Sajedi 2015).

Although biopsy is the traditional method to genetically characterise lung cancer tumours, it is an invasive and painful procedure for the patient. The development of personalised medicine has changed the therapeutic strategy from classical chemotherapy and radiotherapy to a genetic modification targeted therapy (Morgado et al. 2021). The literature describes a number of lung cancer diagnostic techniques that use support vector machines to anticipate normal and abnormal lung cancers (Senthil and Ayshwarya 2018). A standard dataset for lung cancer is used in experiments. The obtained solutions are compared to those obtained by the complete feature set, the F-score, and the correlation-based feature selection methods. The obtained solutions are validated with three different classifiers, such as the support vector machine (SVM), the backpropagation neural network (BPNN), and the K-nearest neighbour (KNN). The comparison results demonstrate the suggested intelligent system's strong diagnosing capabilities and its potential as a tool for the early detection of lung cancer (Lu et al. 2014).

Clinical data analysis and patient care are carried out through the Healthcare Analytics (HcA) procedure. The course of therapy is determined by the evaluation of clinical information gathered from Electronic Health Records (EHRs), pharmaceutical costs, costs associated with research and development, and patient claims (Pradeep and Naveen 2018). Computed tomography (CT) is the greatest imaging method for use in the medical industry and helps doctors locate malignant cells precisely. The authors suggest an automated method to segment and analyse the lungs, classifying each lung as normal or cancerous (Priya and Joseph Jawhar 2020). The suggested model is a classification-based, effective method for detecting lung cancer disorders that makes use of machine learning techniques. Although the method produced good results, using it takes significant computational skill (Chauhan and Jaiswal 2016). LeNet, AlexNet, and VGG-16 deep learning models are used in the study to detect lung tumours. An open dataset of computed tomography (CT) images

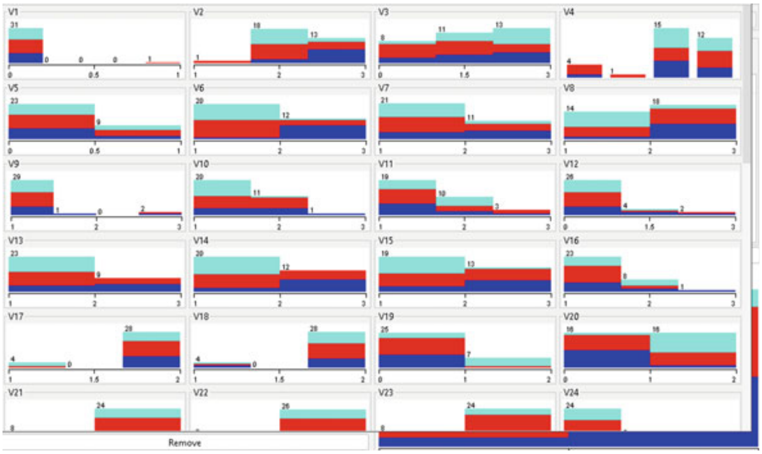


Fig. 1 Attributes of the dataset

was used for the tests. Convolutional neural networks (CNNs) have been utilised in the experiment for classification and feature extraction (Toğaçar et al. 2020).

3 Dataset Description

For the proposed methodology, we used a dataset from the UCI data repository as cancer dataset Lung Cancer. The dataset is having a totally 57 and 32 characteristics (Fig. 1), instances, respectively, as well as one class attribute in its entirety. The important purpose of our proposed study is to classify and evaluate the performances of classification techniques.

4 Methodology

4.1 Data Preprocessing

In the field of machine learning, making the right decision will be successful only if handling the missing values is done properly. Removing null values is very necessary step to find the detection in lung cancer. In the present study, KNN for preprocessing step is applied.

4.2 Feature Extraction

In the field of machine learning, feature extraction from input data is used to collect the relevant data for cancer to predict patient health condition for interpretation using Weka tool (Fig. 1). In the Table 1. Features are extracted for all the 56 classes maximum, minimum, mean, and StdDev. The feature extraction for the dataset is calculated as to decrease the complexity of detecting lung cancer (Fig. 2).

5 Performance Evaluation Results

5.1 Classification Using Machine Learning Algorithms

After the feature selection, classification is the important step in the machine learning field. It is the process of arranging the information into logical groups. In this paper, we have applied Zero-R, Decision table, JRIP rules classification algorithms to predict cancer for medical data. In order to compare classification accuracy between features of various classes, the Zero-R, Decision table, JRIP algorithms are used.

As shown in Table 2, ROC area and True Positive rate of Class 2 instance are higher when compared to the other classes (Fig. 3).

As shown in Table 3, ROC area of Class 3 and True Positive rate of Class 1 instance are higher when compared to the other classes (Fig. 4).

As shown in Table 4, ROC area of Class 3 and True Positive rate of Class 1 instance are higher when compared to the other classes (Fig. 5).

Comparative result investigation of classification algorithms for lung cancer data is summarised in Table 5. In the table, Destination table classification algorithm has given the best result and rate of correctly classified.

6 Conclusion

In the field of healthcare system, because cancer cells are so complicated, predicting lung cancer is one of the most difficult medical problems to solve. If the lung cancer prediction is too late, significantly increasing the death rate of the patient. Cancer can be cured if it is found and treated early enough. In the proposed methodology employed to forecast the onset of lung cancer. The main goal of this system is to give customers an early warning, so they can save time and money. Positive results from the performance evaluation of the suggested approach show that oncologists can use Decision table to help in lung cancer.

Table 1 Feature selection of lung cancer data

Sl. No.	Maximum	Minimum	Mean	StdDev
V1	1	0	0.031	0.177
V2	3	1	2.375	0.554
V3	3	0	2.031	1.031
V4	4	1	15	12
V5	1	0	0.281	0.457
V6	3	1	2.188	0.738
V7	3	1	2.125	0.751
V8	3	1	2.406	0.756
V9	3	1	1.156	0.515
V10	3	1	1.406	0.56
V11	3	1	1.5	0.672
V12	3	0	0.906	0.856
V13	3	1	1.938	0.801
V14	3	1	2.188	0.738
V15	3	1	2.219	0.751
V16	3	1	1.131	0.535
V17	2	1	1.875	0.336
V18	2	1	1.875	0.336
V19	2	0	0.469	0.842
V20	2	0	1.094	0.963
V21	2	1	1.75	0.44
V22	2	1	1.813	0.397
V23	2	1	1.75	0.44
V24	3	1	1.344	0.653
V25	3	1	1.719	0.581
V26	3	1	1.938	0.801
V27	3	2	2.313	0.471
V28	3	1	2.063	0.564
V29	3	1	2.156	0.677
V30	3	1	1.438	0.716
V31	3	1	2.313	0.896
V32	3	1	2.313	0.896
V33	3	1	2.719	0.634
V34	3	1	1.719	0.729
V35	3	1	1.625	0.707
V36	3	1	1.688	0.535
V37	3	1	1.688	0.535

(continued)

Table 1 (continued)

Sl. No.	Maximum	Minimum	Mean	StdDev
V38	11	1	18	2
V39	3	1	1.969	0.474
V40	3	1	2	0.44
V41	3	1	1.594	0.615
V42	3	1	1.75	0.622
V43	3	1	2.063	0.354
V44	3	1	1.875	0.492
V45	3	1	1.969	0.4
V46	3	1	1.938	0.435
V47	3	2	2.063	0.246
V48	3	2	2.063	0.246
V49	1	3	2	0.319
V50	1	3	2	0.359
V51	1	3	2	0.508
V52	3	1	1.844	0.448
V53	3	1	1.844	0.448
V54	1	2	1.563	0.504
V55	2	1	1.813	0.397
V56	1	2	1.719	0.457

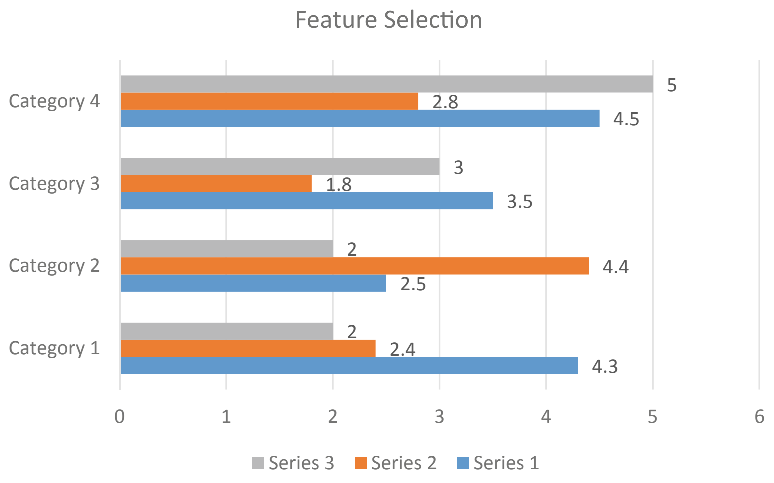


Fig. 2 Feature selection for the lung cancer dataset

Table 2 Zero-R classification result

Class	TP rate	FP rate	Precision	Recall	F-measure	MCC	ROC area
1	0.000	0.000	1.000	0.001.0000	1.000	0.401	0.259
2	1.000	1.000	0.406	0.000	0.578	0.393	0.359
3	0.000	0.000	1.000	0.000	1.000	0.464	0.300
Weighted Avg.	0.406	0.406	1.000	0.406	1.000	0.417	0.312

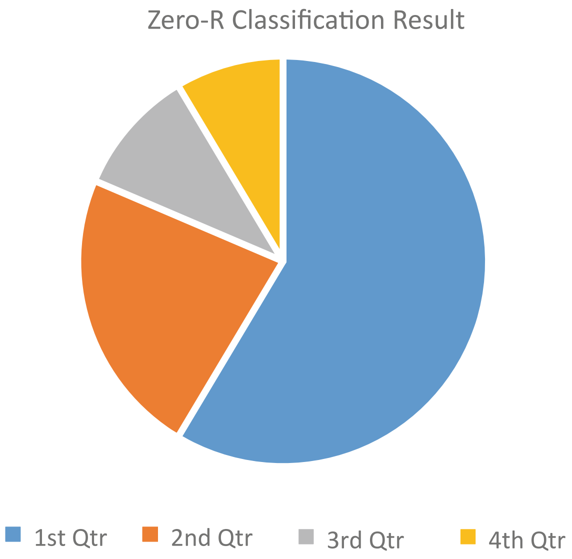


Fig. 3 Zero-R classification result

Table 3 Decision table classification result

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area
1	0.778	0.174	0.636	0.778	0.700	0.572	0.838
2	0.462	0.316	0.500	0.462	0.480	0.148	0.522
3	0.500	0.182	0.556	0.500	0.526	0.328	0.845
Weighted Avg.	0.563	0.234	0.556	0.563	0.556	0.323	0.712

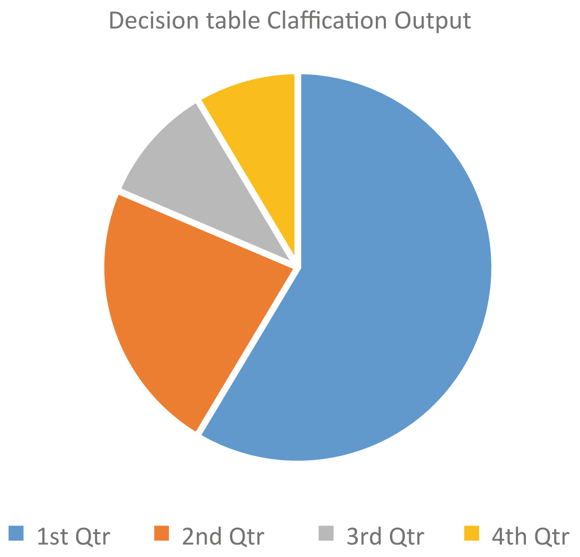


Fig. 4 Decision table classification

Table 4 JRIP rules classification result

Class	TP rate	FP rate	Precision	Recall	F-measure	MCC	ROC area
1	0.556	0.217	0.500	0.556	0.526	0.328	0.691
2	0.538	0.474	0.438	0.538	0.483	0.064	0.474
3	0.400	0.091	0.667	0.400	0.500	0.367	0.711
Weighted Avg.	0.500	0.282	0.527	0.500	0.500	0.233	0.609

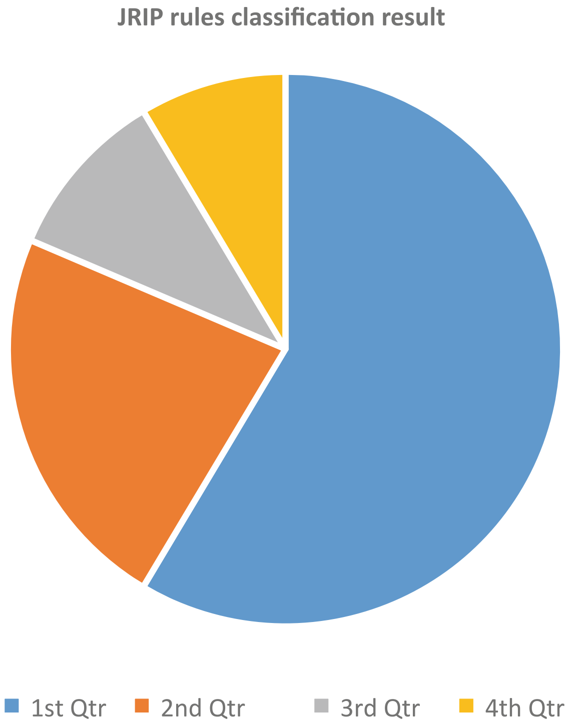


Fig. 5 JRIP rules classification

Table 5 Comparative result analysis of classification algorithms for lung cancer data

Name of the model	JRIP rules	Decision table	Zero-R
Correctly classified instances (%)	50	56.25	40.625
Incorrectly classified instances (%)	50	43.75	59.375
Kappa statistic	0.2312	0.3402	0
Mean absolute error	0.3714	0.3662	0.4409
Root mean squared error	0.4941	0.4271	0.4701
Relative absolute error (%)	84.2375	83.0551	100
Root relative squared error (%)	105.1063	90.8377	100
Time taken to build model (s)	0.02	0.05	0

References

Chauhan D, Jaiswal V (2016) An efficient data mining classification approach for detecting lung cancer disease. In: 2016 international conference on communication and electronics systems (ICCES). IEEE, pp 1–8

Dezfuly M, Sajedi H (2015) Predict survival of patients with lung cancer using an ensemble feature selection algorithm and classification methods in data mining. J Inf 1(1):1–11

Fang L, Zhao H, Wang P, Yu M, Yan J, Cheng W, Chen P (2015) Feature selection method based on mutual information and class separability for dimension reduction in multidimensional time series for clinical data. Biomed Signal Process Control 21:82–89

https://github.com/lpfgarcia/ucipp/blob/master/uci/lung-cancer.arff

Kancherla K, Mukkamala S (2012) Feature selection for lung cancer detection using SVM based recursive feature elimination method. In: European conference on evolutionary computation, machine learning and data mining in bioinformatics. Springer, Berlin, pp 168–176

Lee IH, Lushington GH, Visvanathan M (2011) A filter-based feature selection approach for identifying potential biomarkers for lung cancer. J Clin Bioinform 1(1):1–8

Lu C, Zhu Z, Gu X (2014) An intelligent system for lung cancer diagnosis using a new genetic algorithm based feature selection method. J Med Syst 38(9):1–9

Maleki N, Zeinali Y, Niaki STA (2021) A k-NN method for lung cancer prognosis with the use of a genetic algorithm for feature selection. Expert Syst Appl 164:113981

Mary Adline Priya M, Joseph Jawhar S (2020) Advanced lung cancer classification approach adopting modified graph clustering and whale optimisation-based feature selection technique accompanied by a hybrid ensemble classifier. IET Image Process 14(10):2204–2215

Morgado J, Pereira T, Silva F, Freitas C, Negrão E, de Lima BF, Oliveira HP (2021) Machine learning and feature selection methods for egfr mutation status prediction in lung cancer. Appl Sci 11(7):3273

Pradeep KR, Naveen NC (2018) Lung cancer survivability prediction based on performance using classification techniques of support vector machines, C4. 5 and Naive Bayes algorithms for healthcare analytics. Procedia Comput Sci 132:412–420

Ramani RG, Jacob SG (2013) Improved classification of lung cancer tumors based on structural and physicochemical properties of proteins using data mining models. PLoS ONE 8(3):e58772

Senthil S, Ayshwarya B (2018) Predicting lung cancer using datamining techniques with the AID of SVM classifier. In: 2018 second international conference on green computing and internet of things (ICGCIoT) IEEE, pp 210–216

Shanthi S, Rajkumar N (2021) Lung cancer prediction using stochastic diffusion search (SDS) based feature selection and machine learning methods. Neural Process Lett 53(4):2617–2630

Toğaçar M, Ergen B, Cömert Z (2020) Detection of lung cancer on chest CT images using minimum redundancy maximum relevance feature selection method with convolutional neural networks. Biocybernetics Biomed Eng 40(1):23–39

Xie NN, Hu L, Li TH (2015) Lung cancer risk prediction method based on feature selection and artificial neural network. Asian Pac J Cancer Prev 15(23):10539–10542

Optimal Solution for Distance Detection Using Deep Learning Techniques on Embedded Devices

Uday Kulkarni, Shashank Hegde, Abhishek Hosamani, Vivek P. Marakumbi, Ganesh Vernekar, Akshay R. Shanbhag, S. M. Meena, and Sunil V. Guralhosur

1 Introduction

A sophisticated artificial neural network called a deep neural network (DNN) (Szegedy et al. 2013) has several layers between the input and output levels; these artificial neurons that make up these various layers are interconnected to address a real-world task like image classification, annotation, or object detection, and these neurons process the input using extensive calculations carried out across layers. Input in deep neural networks is routed between nodes and given an appropriate weights. A node with a higher weight will have a greater impact on the nodes in the next layer; these weighted inputs are combined to create an output in the final layer. Due to the massive quantity of data being processed and the numerous difficult mathematical computations required by DNN to attain improved accuracy, deep learning systems require strong hardware. In such cases, DNN models are quantized and transferred on to edge devices in order to meet its computing requirements and make it power-efficient without compromising on model accuracy.

Our topic of discussion focuses on finding the object, mainly detecting the person and finding the distance between those persons in image and in live feed using a camera. A computer vision method called object detection is used to find occurrences of objects in images or videos. We have researched the various object detection algorithms (Girshick et al. 2014) and techniques listed below, which concentrate on

U. Kulkarni (✉) · S. Hegde · A. Hosamani · V. P. Marakumbi · G. Vernekar · A. R. Shanbhag · S. M. Meena · S. V. Guralhosur
KLE Technological University, Hubballi, India
e-mail: udaykulkarni@kletech.ac.in

S. M. Meena
e-mail: msm@kletech.ac.in

S. V. Guralhosur
e-mail: svguralhosur@kletech.ac.in

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_28

creating boundary boxes around objects using a variety of techniques, such as anchor boxes, boundary boxes, or masking regions of interest. All of these techniques are covered in our literature review.

The proposed method discussed in paper is to use an object detection technique called YOLOv7 (Wang et al. 2017) which is the fastest and most accurate real-time object detection model for computer vision tasks. Its improved architecture surpasses all previous YOLO (Rekha et al. 2020) versions as well as all other object detection models in terms of both speed and accuracy on the MS COCO dataset. Boundary boxes are formed around each object mentioned in the dataset using the model with YOLOv7 weights. We are interested in the boundary boxes generated around people, and we use the Euclidean distance formula to calculate the distance between these boundary boxes. Using quantization, a technique is used to convert the tensors or parameters to lower bit width from 32 bit floating point values, and our final model is quantized using PyTorch 8-bit quantization API leading to 8-bit model which is later ported onto edge devices like mobile phones, and proposed workflow is depicted in Fig. 1.

The paper is organized into various sections listed below. Section 2 focuses on comparison of available object detection algorithms and background study of YOLO architecture and its modules. In Section 3, we analyze the previous versions of YOLO with YOLOv7 and discuss its architecture. Section 4 discusses about the proposed methodology wherein the image is fed to the model, object is detected, and distance between two objects belonging to people class is calculated with the model being quantized from 32 bit to 8 bit, and the model is ported to edge device. In Section 5, the results are discussed on YOLOv7 architecture before and after quantization of model in various operating systems concerning the accuracy, flops, and quantization loss. Section 6 tells the conclusion and gives the future scopes.

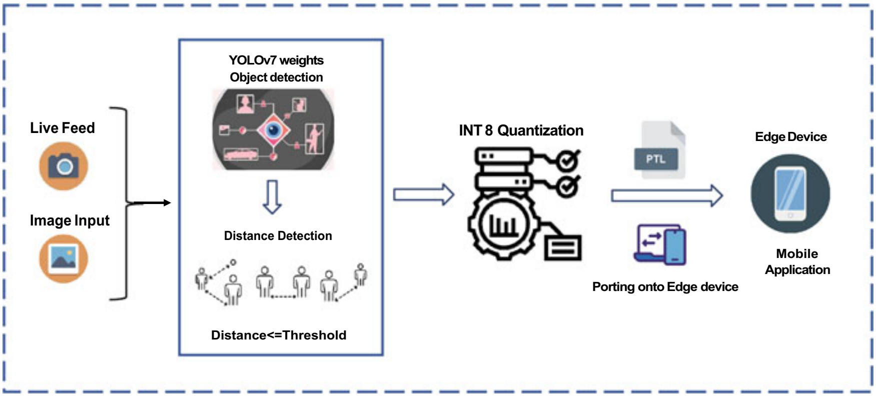


Fig. 1 Workflow of proposed methodology

2 Background Study

Several techniques have been developed in finding distance between two objects which are categorized under class people. Among notable systems, (Wang et al. 2022) several systems are evaluated and considered for our system development. The first object detection model used to detect objects in an image is regions with convolutional neural network (R-CNN) (Chen 2017; Girshick et al. 2014). It uses selective search algorithm which tries to identify patterns in the image, and based on the image, it proposes various regions. R-CNN takes an input image and extracts all possible region proposals, i.e., it simultaneously detects the object scores and objects that are detected are drawn with bounding boxes which generate sub-segments to have multiple regions of an image. This technique combines similar regions based on size, enclosure, texture, color, and shape compatibility to form larger regions. These regions produce final object locations (region of interest). So it was used to solve image detection tasks, but it is slow. It takes more time to train the model. So generally, we opt for a faster network, and one such faster network is Fast R-CNN.

In one of the works of "Fast R-CNN" (Girshick 2015; Wang et al. 2017) which proposes a fast region-based convolutional neural network method (Fast R-CNN) for object detection. Object proposals determine bounding boxes to determine the objects and locations in the image. A Fast R-CNN takes an image, and some object proposals are given as input to convolutional neural network (CNN). This image is passed to multiple convolutional layers and max pooling layers to form a feature map. The region of interest (ROI) projection is used to map the object proposals to the feature map. It finds the coordinates of proposals on a feature map corresponding to the original image. So Fast R-CNN improves the training time and detection time compared to R-CNN.

Similarly, in Faster R-CNN (Chen and Gupta 2017; Roh and Lee 2017), the first step is to generate anchor boxes, which captures all types of objects in an image and all anchor boxes are drawn to each object. When anchor boxes are generated, intersection over union (IOU) computes the intersection over union on this predicted bounding box. The anchor box with more than 50% IOU is labeled as the foreground class. The anchor box with less than 50% IOU is labeled as a background class. The task of region proposed network (RPN) is to detect foreground and background classes. The output of RPN gives a feature map of those anchor boxes that are labeled as foreground class. Now the feature map is fed to the region of interest (ROI) pooling layer in order to reduce all feature maps to the same size. At last, there is a classifier and a regressor, the task of a classifier is to find whether there is an object which is an image, and the task of the regressor is to draw the boundary box on the object that is detected. As a result, Fast R-CNN has an extra CNN for obtaining the regional proposal, but RPN is trained by extracting all anchors in the mini-batch of size 256 from a single image, and this image may contain multiple samples which have similar features in the network that takes more time to achieve convergence.

In one of the works "SSD: Single Shot multibox Detector" (Liu et al. 2016; Zhao et al. 2021), it discusses how convolution filters and feature maps are the two main components for object detection, how SSD removes the region proposal network to speed up the process and ultimately improve performance, and how multi-scale features and default boxes are the modifications that are applied onto SSD to achieve minimal accuracy loss. In terms of accuracy, these enhancements enable the approach to compete with Fast R-CNN even when using pictures of lesser quality. This comparison shows that while it surpasses the accuracy of the Fast R-CNN and achieves real-time processing speed, and shallow layers in a neural network might not produce enough high-level characteristics to make prediction for tiny objects.

You Only Look Once (YOLO) is the state-of-the-art object detection algorithm which is used to detect objects in the field of computer vision. YOLO algorithm (Shafiee et al. 2017) is used in various fields such as social distancing in order to calculate the distance between the objects with class people and identifying face mask on people's face, and it has high inference speed, high accuracy, and learning capabilities.

3 YOLO Architecture Description You Only Look Once (YOLO) is One of the Most Efficient

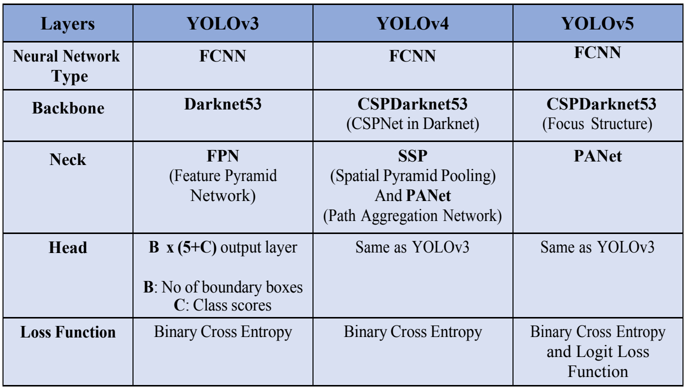
Model architectures and object detection algorithms. The primary reason for its popularity is that it makes use of one of the greatest neural network designs to create high accuracy and overall processing speed. YOLO architecture (Aggarwal 2021) heavily relies on fully connected neural network (FCNN). YOLO framework consists of three significant parts:

- Head
- Neck
- Backbone.

The backbone takes out the significant features from a training image which is fed to the head part via the neck. The neck part gathers feature maps and generates feature pyramids. Head consists of some fully connected layers and output layers which are essential for the detection and classification of images.

The newer versions of YOLO were released like YOLOv3, YOLOv4, and YOLOv5 which displayed better performance than previous one with some minute changes in the architecture which enhanced the performance, speed, and accuracy of the model. The components of YOLO family are discussed below in Fig. 2.

YOLOv7 (Yolov7 Paper Explanation: Object Detection and YOLOv7 Pose 2022) is procured from scaled YOLOv4 (Bochkovskiy et al. 2004) and YOLO-R. Further



Layers	YOLOv3	YOLOv4	YOLOv5
Neural Network Type	**FCNN**	**FCNN**	**FCNN**
Backbone	**Darknet53**	**CSPDarknet53** (CSPNet in Darknet)	**CSPDarknet53** (Focus Structure)
Neck	**FPN** (Feature Pyramid Network)	**SSP** (Spatial Pyramid Pooling) And **PANet** (Path Aggregation Network)	**PANet**
Head	**B x (5+C)** output layer **B**: No of boundary boxes **C**: Class scores	Same as YOLOv3	Same as YOLOv3
Loss Function	Binary Cross Entropy	Binary Cross Entropy	Binary Cross Entropy and Logit Loss Function

Fig. 2 Evolution of YOLO family

experiments were done on these base models to develop new and efficient architecture with few improvements and changes. We will discuss them one by one:

A. *Extended Efficient Layer Aggregation Network (E-ELAN)*

The E-ELAN is one of the most significant operational blocks in the backbone of YOLOv7. The architecture is designed considering the significant factors like activations, gradient path, and memory access cost that affect the performance of the model. While maintaining the original gradient path, the proposed E-ELAN uses shuffle and merge multiplicity to improve the neural network's ability for learning. The E-ELAN design improves learning for the framework using the ELAN computational block as a foundation (Fig. 3).

B. *Scaling of Models in YOLOv7*

Different applications demand different types of models having accuracy and speed according to their requirements. Some applications may require accurate models, and others may need higher model speed. A prominent technique for scaling models is called network architecture search (NAS). Researchers employ it to repeatedly loop over parameters in quest of the most effective scaling factors.

C. *Planned Re-parameterized Convolution*

Re-parameterization is a method for enhancing the model after training. It lengthens the training process but yields better inference outcomes. To finalize the model, the types of re-parameterization used are model-level and module-level ensembles.

Model-level re-parameterization is done by taking different datasets with same properties and training multiple models to take average of weights for the final model.

Re-parameterization at the module level has become quite popular in research. This approach divides the model training process into a number of parts. To create the final model, the results are combined.

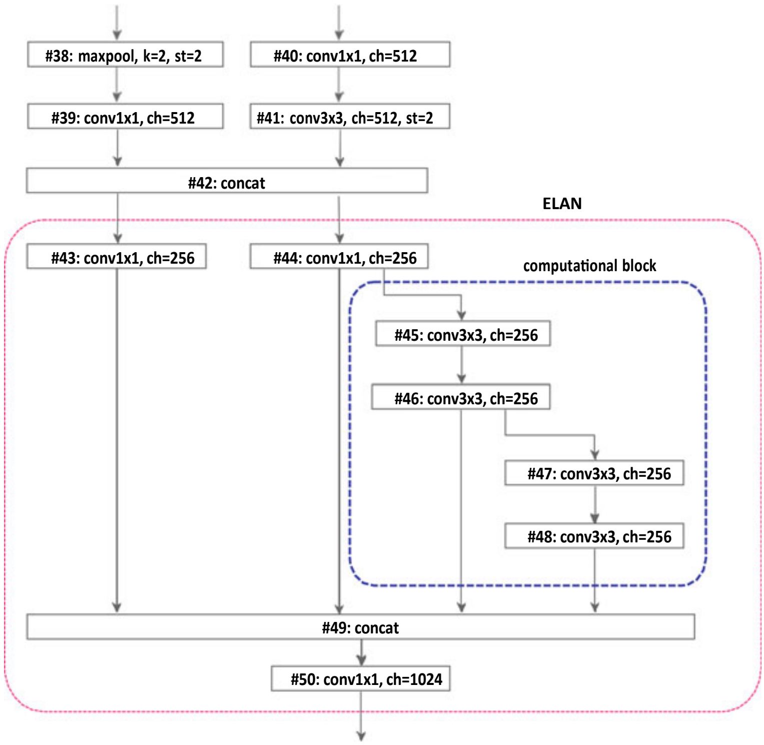


Fig. 3 E-ELAN (Yolov7 Paper Explanation: Object Detection and YOLOv7 Pose 2022)

4 Proposed Methodology

A. *Object Detection*

There is a computer vision library known as OpenCV which takes input as image taken from the cameras and locates all instances of objects in an image having known set of object categories. The image is fed to the multiple convolutional layers and max pooling layers, and applying ReLU function will give an output image drawn with bounding box. The given image is divided into N * N grid cells. For each grid cell, the feature vector is calculated with attributes O_{obj} (object is present with Boolean value as 0 or 1), B_x, B_y (centroid of grid cell having coordinates x and y axis, B_h, B_w (height and width of the boundary box), and C_1, C_2 (objects belonging to particular class). The centroid of the bounding box is calculated by mean distance between top left most coordinates (x_1, y_1) and bottom right most coordinates (x_2, y_2). When multiple bounding boxes are detected for single object, then intersection over union (IOU) is used to keep only the bounding box with high probability and discard all other bounding box having other probabilities lower than high probability. This is also known as non-max suppression. The boundary box with its class probability is assigned to all objects in the given image (Mathurkar et al. 2021; Zhao et al. 2021).

$$
\begin{aligned}
y = [\,O\_obj\,] \\
[\,B\_x] \\
[\,B\_y] \\
[\,B\_h] \\
[\,B\_w] \\
[\,C1] \\
[\,C2]
\end{aligned}
\tag{1}
$$

B. *Distance Detection*

The distance formula used to calculate distance between two objects that are detected under class people is Euclidean distance. The model creates boundary boxes around the input frame's objects, and the centroids of those boundary boxes are used to measure the distance between adjacent boundary boxes, particularly those that include people. The model's output will indicate if the distance is found to be lesser than a specific threshold which is set in advanced (Mathurkar et al. 2021; Salagrama et al. 2022).

Using the relevant mathematical equations shown below, the model predicts all four coordinates (Figs. 4 and 5).

$$
\begin{aligned}
C_x &= \sigma(P_x) + m_x \\
C_y &= \sigma \sum (P_y) + m_y \\
C_h &= q_h e^{P_h} \\
C_w &= q_w e^{P_w}
\end{aligned}
\tag{2}
$$

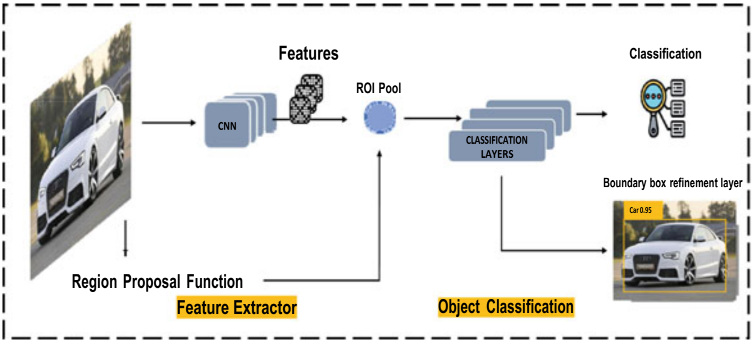


Fig. 4 Object detection using YOLOv7

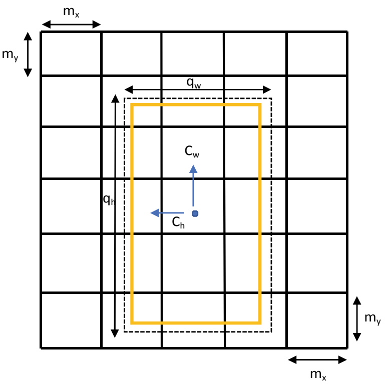


Fig. 5 Location prediction with coordinates in boundary box

where C_y and C_x are y and x coordinates of center, whereas C_w and C_h are width and height of our prediction. P_x, P_y, P_h, and P_w, are the outputs of grid cell. m_x and m_y are the upper left coordinates of the network. q_h and q_w are anchors dimensions of the bounding box.

To find the distance ($\text{dist}_{\text{boundarybox}}$) between the centers of each bounding boxes which represent a person, we use the following formula:

$$\text{dist}_{\text{boundarybox}} = \sqrt[2]{(X_{\max} - X_{\min})^2 + (Y_{\max} - Y_{\min})^2} \tag{3}$$

C. *Quantization*

DNN demands high computational cost and memory requirements to tackle those problems. The model generated is huge and consists of float parameters leading to more consumption of memory. To lower the computational burden and memory needs without compromising accuracy, the quantization approach is used. To produce the desired output, the model uses YOLOv7 weights and a distance detection technique. To obtain a quantized model, it applies the PyTorch 8-bit post-training quantization approach. PyTorch supports multiple quantization methods for deep learning models. In most of the cases, the model is trained using FP32 and then converted to the INT8 model (Uday Kulkarni et al. 2021).

The conversion process from floating point to fixed point uses the following mapping equation, where x is the input value and scale is used to scale the input value to quantization range:

$$Q(x, \text{scale}, \text{zeropoint}) = \text{round}(x/\text{scale} + \text{zeropoint}) \tag{4}$$

where scale is calculated using Eq. (5)

$$\text{scale} = \frac{2^{b-1} - 1}{a - b} \tag{5}$$

Here in above Eq. (5), b represents bit width and a and b represent lower limit and upper limit in quantization range.

It is important to note that there is no difference between how the floating point zero is represented before and after quantization. It guarantees that quantization errors will not be generated in specific operations, such as zero padding, because the zero before quantization corresponds to a fixed point value after quantization. The quantized tensor can use multiple operations, so that the quantized tensor can easily perform operations and can also be serialized and parallelized.

Mainly after quantizing the actual model compared to floating point 32, the model size is reduced by $4x$, and the memory bandwidth requirement is also reduced by $4x$. Hardware support for INT8 operation makes its calculations generally 2–4 times faster than floating point operations making it easy for porting onto edge devices for all kinds of computer vision-based applications.

Despite the fact that quantization reduces model size and computing expense, it also results in small amount of information loss known as quantization error, which is the difference between the input value and the value after quantization. A measurement statistic termed the signal-to-quantization noise ratio (SQNR) (Hasso et al. 2019) is used to assess how quantization loss affects the model.

$$\text{SQNR} = 10\log_{10}\left(\frac{\sigma_x^2}{\sigma_q^2}\right) \tag{6}$$

where σ_x and σ_q are the input value and the quantized value. It is evident that the SQNR must be reduced in order to maintain the model accuracy. With the SQNR metric taken into account, lowering the FLOPS, which are typically calculated using the number of multiplication and addition operations that model conducts, would be another crucial part of quantization. The multiplication accumulate operations (MACs) are significantly reduced with the quantization, and in each layer, a total number of operations (TO) are calculated using Eq. (7) (Fig. 6).

$$\text{TO} = k(cwh + 1)\left(\frac{M - w + 2p_w}{s_w} + 1\right)\left(\frac{N - w + 2p_h}{s_h} + 1\right) \tag{7}$$

Where K is the number of kernels, M is the input tensor's width, and N is its height, the variables c stand for the number of channels in the input tensors, w and h for the width and height of the convolution kernels, and p_w and p_h for the input tensor's padding width and height.

Porting onto Edge Devices

One of the most widely discussed technological developments since the Internet of Things (IoT) is edge machine learning (Edge ML) (Zhai et al. 2020; Vita et al. 2021) with good reason. The network was not yet prepared to handle the explosion of smart devices linked to the cloud that came along with the emergence of IoT. Companies ignored important cloud computing concerns including security because of clogged

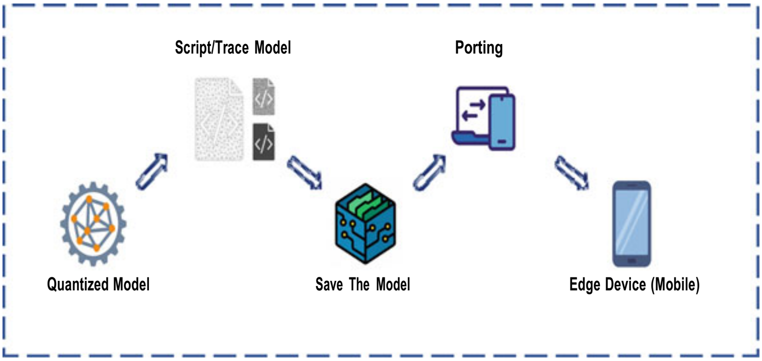


Fig. 6 Porting the model onto edge device (Android mobile)

cloud networks. Edge protect privacy allows novel interactive use cases, and ML models must increasingly be run on edge devices. We made use of PyTorch Mobile which allows users to make smooth transition from developing a model to deploying it while remaining inside the PyTorch environment. The model is quantized to reduce the size and computational cost of the model. Then the model is passed into script function which examines the source code and compiles it as TorchScript code with the help of TorchScript compiler and returns a script function. The model is further optimized for mobile before it is passed to function which saves the optimized model in .ptl format. Then, the ptl model is deployed in app developed in Android Studio to get the inference from Android application.

5 Results

A. *Dataset Description*

The Microsoft Common Objects in Context (MS COCO) (Lin et al. 2014) is a huge dataset designed by Microsoft. It contains 328,000 images of 80 different classes of everyday things and person. Numerous computer vision applications, including object identification, face detection, posture estimation, and others, heavily rely on the visual dataset COCO. It entails identifying the items that are present, localizing them in 2D and 3D, figuring out their characteristics, and describing the relationships between them. As a result, the dataset may be used to train algorithms for object identification and object categorization.

B. *Experimental results*

Our work demonstrates that how the optimized application on a low-end device (e.g., mobile) and high-end device (e.g., servers) phone could detect distance based

Fig. 7 COCO dataset (Lin et al. 2014)

on methodology explained above. Output can be generated by passing images, prerecorded videos, or by sending the live feed from the camera on both laptop and mobile phone (Fig. 7).

The output is provided by the quantized model after porting to an edge device as shown in Figs. 8 and 9, respectively.

As it is obvious, the boundary box formed around a person on the screen is red when the distance between them is lesser than the predetermined threshold, in contrast to items that are identified and are represented by various color boundary boxes as depicted in Fig. 8. The border box surrounding nearby people will be highlighted in red, whereas in all other circumstances, it will be yellow.

Similar to the other output image snapshot in Fig. 9, the model creates the output by detecting all objects in the frame and determining their likelihood of being that object. This results in the desired output with a red boundary box around people whose distance is less than the predetermined threshold. Table 1 illustrates how much the model has been quantized in terms of model size. The model size was 273.5 MB before quantization, and after quantization, it was 72.5 MB with hardly any accuracy loss.

6 Conclusion and Future Scope

The primary goal of the article was to identify the optimal approach for distance detection on edge devices. It is accomplished by using the YOLOv7 model, which detects objects in photographs or videos more quickly and accurately than previous

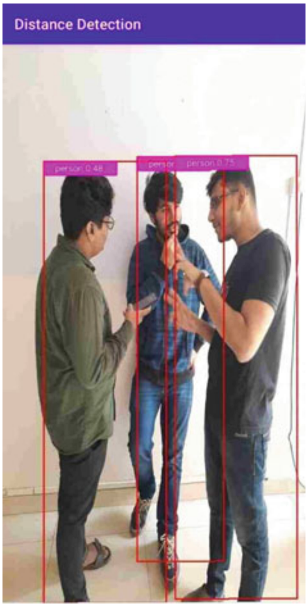


Fig. 8 Output image showing person in close proximity with a red boundary box

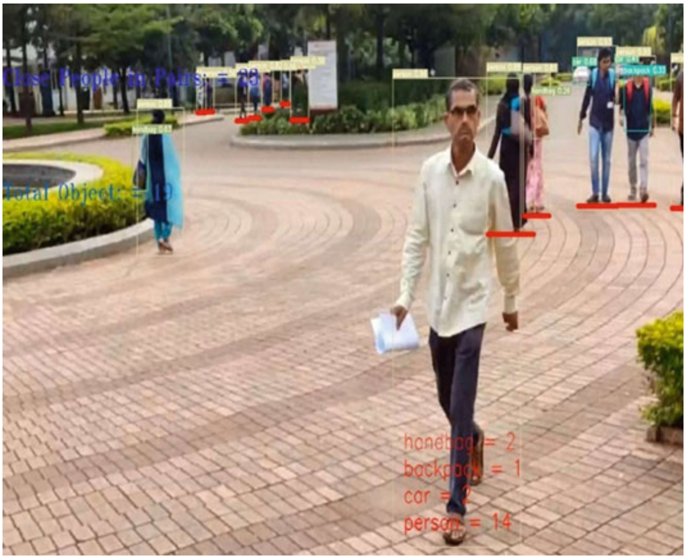


Fig. 9 Output image of live feed

Table 1 Results of before and after quantization

Model	No. of bits	Model size (MB)	Mean avg. precision
YOLOv7	32-bit	273.5	59.5
Quantized YOLOv7	8-bit	72.5	56.3

models. The boundary boxes created for these trained items are then processed to determine the Euclidean distance between people exclusively. This model is later quantized and ported on an edge devices.

To name one part of the work that may be improved, the model could be quantized more aggressively to minimize bit representation to reduce the size, but maintaining accuracy and training the model to include more things would be challenging.

References

Aggarwal (2021) Yolo Explained. https://medium.com/analyticsvidhya/yolo-explained-5b6f4564f31, 7 Jan 2021. [Online; accessed 2022–12–05].

Bochkovskiy A, Wang C-Y, Liao H-YM (2020) Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934

Chen X, Gupta A (2017) An implementation of faster rcnn with study for region sampling. arXiv preprint arXiv:1702.02138

Chen C, Liu M-Y, Tuzel O, Xiao J (2017) R-cnn for small object detection. In: Asian conference on computer vision. Springer, Berlin, pp 214–230

Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448

Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587

Hasso A, Jacksi K, Smith K (2019) Effect of quantization error and sqnr on the adc using truncating method to the nearest integer bit. In: 2019 international conference on advanced science and engineering (ICOASE). IEEE, pp 112–117

Kulkarni U, Meena SM, Gurlahosur SV, Bhogar G (2021) Quantization friendly mobilenet (qf-mobilenet) architecture for vision based applications on embedded platforms. Neural Netw 136:28–39

Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dolla´r P, Zitnick CL (2014) Microsoft coco: common objects in context. In: European conference on computer vision, Springer, Berlin, pp 740–755

Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) Ssd: Single shot multibox detector. In: European conference on computer vision, Springer, Berlin, pp 21–37

Mathurkar G, Parkhi C, Utekar M, Chitte PH (2021) Ensuring social distancing using machine learning. In: ITM web of conferences, vol 40. EDP Sciences, p 03049

Rekha BS, Marium A, Srinivasan GN, Shetty SA (2020) Literature survey on object detection using yolo. Int Res J Eng Technol (IRJET)

Roh M-C, Lee J-y (2017) Faster-rcnn for accurate object detection. In: 2017 fifteenth IAPR international conference on machine vision applications (MVA). IEEE, pp 514–517

Salagrama S, Kumar HH, Nikitha R, Prasanna G, Sharma K, Awasthi S (2022) Real time social distance detection using deep learning. In: 2022 international conference on computational intelligence and sustainable engineering solutions (CISES). IEEE, pp 541–544

Shafiee MJ, Chywl B, Li F, Wong A (2017) Fast yolo: a fast you only look once system for real-time embedded object detection in video. arXiv preprint arXiv:1709.05943

Szegedy C, Toshev A, Erhan D (2013) Deep neural networks for object detection. Adv neural Inf Proc Syst 26

Trivedi S (2020) Object detection—A modular approach. https://medium.com/visionwizard/object-detection-4bf3edadf07f, may 4 2020. [Online; accessed 2022–12–05]

Vita FD, Nocera G, Bruneo D, Tomaselli V, Giacalone D, Das SK (2021) Porting deep neural networks on the edge via dynamic k-means compression: A case study of plant disease detection. Pervasive Mob Comput 75:101437 (2021)

Wang X, Shrivastava A, Gupta A (2017) A-fast- rcnn: hard positive generation via adversary for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2606–2615

Wang C-Y, Bochkovskiy A, Liao H-YM (2022) Yolov7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696

Yolov7 Paper Explanation: Object Detection and YOLOv7 Pose. https://learnopencv.com/yolov7-object-detection-paper-explanation-and-inference/ 2 Aug 2022. [Online; Accessed 2022-12-05]

Zhai S, Shang D, Wang S, Dong S (2020) Df- ssd: an improved ssd object detection algorithm based on densenet and feature fusion. IEEE Access 8:24344–24357

Zhao Z-Q, Zheng P, Xu S-t, Wu X (2019) Object detection with deep learning: a review. IEEE Trans Neural Netw Learn Syst 30(11):3212–3232

Zhao Z, Wang K, Ling N, Xing G (2021) Edgeml: an automl framework for real-time deep learning on the edge. In: Proceedings of the international conference on internet-of-things design and implementation, pp 133–144

Cloud Data Security Using Hybrid Encryption with Blockchain

Kalaiselvi Soundappan and Govind Sreekar Shenoy

1 Introduction

Nowadays, cloud server services (CSs) have been popularly adopted by industries or organizations. This enormous growth of CS allows industries or organizations to outsource their data and IT services to cloud service providers (CSPs). The cloud environment has received significant attention from academic and industry communities. CS is easy for the organization to share the different services securely. The cloud's data are outsourced to the cloud servers through various intermediaries, and it is shared among the cloud data requesters at any time on public clouds. The cloud user can access the data anytime and anywhere in the cloud environment. CS is a network access service that provides on-demand network access to assess the resources and data accessible on the internet. Data security, privacy, and service availability are outsourced to third parties who may be protected via CS services. Cloud DOs may securely move their communications to the cloud while maintaining high security. However, CS uses attribute encryption to guarantee secure message transmission, leaving the permission element unmet. In general, the data saved in CS are enormous. The cost of data integrity verification will be higher, or the user's communication and computing resources will be wasted if the whole dataset is downloaded to validate data integrity. In many data mining and machine learning applications, downloading the cloud data entirely from the cloud server is not necessary.

K. Soundappan (✉)
MSC- Computer Science, Liverpool John Moors University, Chennai, India
e-mail: s.s.kalaiselvi@gmail.com

G. S. Shenoy
Dept. of Information Science Engineering, Nitte Meenakshi Institute of Technology, Bengaluru, India
e-mail: govind.shenoy@nmit.ac.in

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_29

Security has gotten the most on-request issues inside the cloud's CS climate. The cloud users have been liberated from the board's different data, and this administration is dealing with the TPA. Rather than that, this outsider evaluator still has some specific security difficulties and issues. During this review, the TPA can gain admittance to the data in the cloud. Cloud services also give users actual control of their re-appropriated data, which provides a command over security issues toward data stockpiling accuracy in the cloud. To benefit from user solicitation, TPA users do not have to trust to get to CS services regarding security.

2 Benefits of Cloud Computing

Cloud service providers typically implement a comprehensive security framework, but the involvement of third parties forces them to raise the security level. A trustworthy distributed computing administration strives to build a highly secure cloud system with a robust encryption system that can safeguard sensitive data.

1. Protection against DDoS.
2. Data security.
3. Compliance and regulation.
4. Data breaches prevention.
5. Data loss prevention.
6. Account hijacking prevention.
7. Reduction of malicious threats.
8. Proper due diligence.
9. Reduction in impacts of shared resources.

3 Design Goals

To build a safe and fine-grained data deletion mechanism in cloud-based data security to allow the data stored in the cloud to follow a policy. Even if the DO's secret key is compromised, erased data cannot be retrieved. Specifically, the following objectives should be met by this research:

- **Support fine-grained**: policy-based data deletion so that data may be purged accurately and flexibly. In particular, the proposed hybrid encryption policy should reflect Boolean expressions including AND, OR, and threshold operations.
- **Effectiveness**: The system activities for developers should be done with minimal compute and storage overheads (Fig. 1).

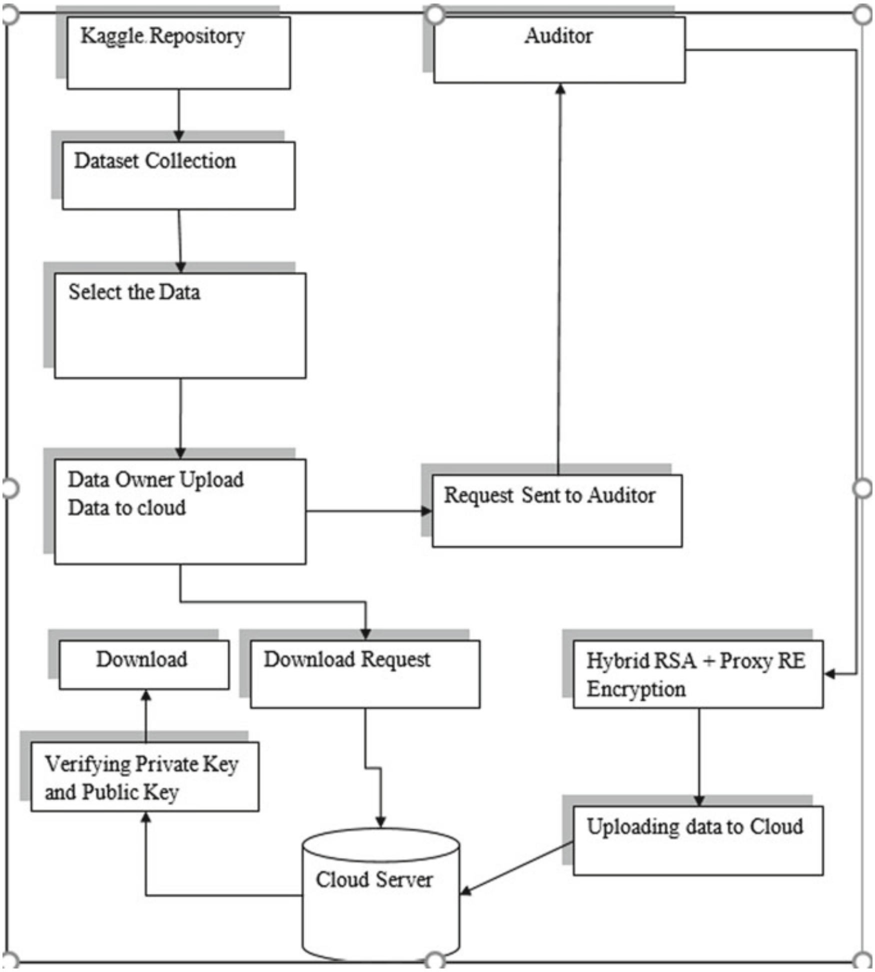


Fig. 1 Proposed architecture

4 Proxy Re-encryption

A total of five algorithms are used in the proposed PRE system. Each algorithm is described in terms of the inputs and outputs it expects and produces:

1. **System Setup (*Setup*)**
- For input, please use the nth security parameter.
- Exit Message: SP System Parameters = (E, q, p, e, G, s).
- Syntax: ***Setup*** $(1^n) \rightarrow SP$.
2. **Key Generation (*KeyGen*)**
- Input: Parameters of the system.

- Output: Secret keys sk_A, sk_B. For users A and B, a re-encryption key *rk* for a proxy server.
- Syntax: ***KeyGen*** $(SP) \rightarrow sk_A, sk_B, rk.$

3. **Data Owner Encryption *(OwnerEnc)***

- Input: Transmission parameters (m), public key, and system parameters (SP) pk_A of a party *A*.
- Output: The ciphertext CT_A and ciphered key CK_A.
- Syntax: ***OwnerEn.***$\left(m, SP, pk_A\right) \rightarrow CT_A, CK_A$

4. **Cloud Re-encryption (*CloudReEnc*)**

- Input: The ciphered key CK_A, system parameters *SP*, and the re-encryption key *rk*.
- Output: The ciphered key $CK_{B.}$
- Syntax: ***CloudReEnc*** $(CK_A, SP, rk) \rightarrow CK_{B.}$

5. **User Decryption (*UserDec*)**

- Input: The ciphertext CT_A, ciphered key CK_B, system parameters *SP*, and a secret key sk_B of user *B*.
- Output: The original message *m*.
- Syntax: ***UserDec*** $(CT_A, CK_{B.}, SP, sk_B) \rightarrow m.$

5 RSA Algorithm

RSA algorithm involves three steps:

- **Key generation.**
- **Encryption.**
- **Decryption**.

Key generation: Key generation must occur before any data can be encrypted. This is an interaction between the user and the cloud service provider (Manjula et al. 2022).

6 Key generation algorithm steps:

1. Pick *a* and *b*, *a* and *b* being prime numbers, respectively. Choose *a* and *b* at random, and make sure they have identical bit lengths for security.
2. Compute n = *a* * *b*.
3. Get the quotient function of Euler, Ø(*n*) = (*a*-1) * (*b*-1).
4. Find the largest common divisor of e and an integer n such that 1 e (*n*), Ø(*n*) is 1.

The exponent e in public-key cryptography has just been made public.

5. Here is how to calculate *d*: *d* = e*P*-1*P*(mod Ø(n)), i.e., *d* is multiplicative inverse of *e* mod Ø(*n*).

6. A secret value, d, is stored as part of the private key $d * e = 1 \bmod Ø(n)$.
7. The public key comprises the modulus n and the public exponent, e, i.e., (e, n).
8. The private key is a secret combination of the modulus n and a private exponent d, i.e., (d, n).

Encryption algorithm:

Encryption is converting data from its plaintext form into an unbreakable code. Steps are the following:

- If a user wants to save data with the cloud service provider, the provider must issue or provide the user the Public- Key (n, e).
- A reversible protocol known as a padding scheme has been implemented to ensure that user information stays within integer bounds.
- A ciphertext is the result of encrypting data (data) C is $C = m^e$ (mod n).
- Now that the encrypted text or data have been uploaded to the cloud storage service, only the service provider can read it.

Decryption algorithm:

According to the definition, decryption is turning cipher text (data) back to its original plaintext. Steps are the following:

- The user makes a data request to the cloud provider.
- As soon as the cloud provider is sure, the user is who they say they are, they provide over the encrypted data
- Cloud users can encrypt their data and then use their computers to decrypt it, $m = c^d$ (mod n).
- Once the user has m, they may revert to the original data by using the padding technique in reverse.

7 Blockchain Technology

The term "blockchain" refers to a data structure in which linked blocks of information are stored linearly. Blocks are used to store information irreversibly in blockchain technology. The chain structure is responsible for the sequential generation of the blocks and their subsequent connection to one another in the form of a block chain. From the blockchain's inception to when this block was generated, all transaction information is recorded inside its body. Blockchain is another example of a distributed, decentralized database. All of the nodes will replicate the backup. Data saved in the blockchain cannot be altered or lost even if a single node fails.

8 Hybrid Encryption Methodology

Hybrid encryption is a system for converting double ciphertexts, meaning that it can take a ciphertext that has been decrypted using one key and use it to translate another ciphertext using the same ciphertext state and a different legend. For this system, Alice and Bob each have their key. During the ciphertext conversion procedure, the plaintext and Alice and Bob's private keys will remain hidden from the proxy server. In this way, under the premise of cryptography, the degree of trust and dependence on the intermediate conversion server is reduced, the information security protection of plaintext and bilateral users is also guaranteed, and the process of entrusting or redistributing ciphertext is relatively convenient.

Taking financial transactions as an example, user Alice stores encrypted files on an untrusted server and allows users to access them but cannot perform ciphertext conversion online whenever data access occurs. If hybrid encryption technology is used, Alice can calculate a re-encryption key based on her private and public keys and set it on the proxy server. Encrypted so that only the delegate's private key may read it. Hybrid encryption ensures that neither the plaintext nor the private key is accessible to the proxy server. To gain even the barest measure of confidence in the server, it is sufficient that the re-encryption algorithm is correctly implemented. Therefore, in financial transactions, the delegator can conveniently and effectively realize access control to its ciphertext files through proxy re-encryption.

9 Model Design

The model is improved from the peer-to-peer model proposed. As shown in Fig. 2, this research work model comprises four roles: a data holder, a data consumer, a blockchain member, and a proxy service provider. Each node of the solution can be one or more roles, but it cannot be block chain member and proxy service provider simultaneously.

- Data Holder: Data holder has sensitive financial data, such as the balance of the account, the id number of the account holder, password, transaction history. Data holders can authorize data consumers to access their data. Sensitive information is encrypted and stored in a separate distributed database. The hash value of data, the storage address, and the access strategy are uploaded by the data holders and updated in the blockchain system. Through blockchain technology, data holders can shield their financial records from being tampered with in malicious hacking attempts. Before any information can be shared between a data holder and a proxy service provider, the holder must generate and broadcast a hybrid encryption key.
- Information Requester (Data Consumer): A data requester (data consumer) submits an authentication request (authentication request) to the data holder (data owner) to request access to the data holder's data. Following authorization, the

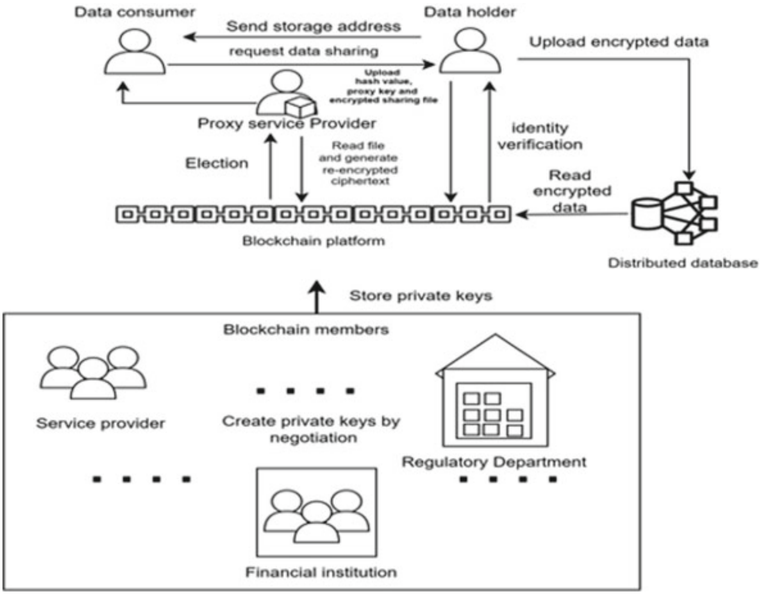


Fig. 2 Blockchain-based sample secure financial data-sharing model

data consumer can use his private key to decrypt the ciphertext and access the protected financial data.

- Blockchain Member: Blockchain members include banks, government regulatory departments, and related service providers. These members maintain the function of the blockchain platform. They also negotiate and generate their private key by calculating and broadcasting a secret value.
- Proxy Service Provider: A node can be selected by delegated proof of stake (DPOS) to play as a proxy service provider. At this node, an algorithm for hybrid encryption is carried out. The proxy service provider re-encrypts the ciphertext of private financial data using the re-encryption key sent by the data holders.

In this model, sensitive financial data were stored both in the distributed database and blockchain platform. Distributed databases hold ciphertext of sensitive financial data, and the hash value of data, storage address of data, and access control rules are stored on the blockchain platform.

10 Protocol

The article suggested using multi-center identity-based hybrid encryption for data exchange. The protocol is improved with the help of Matthew Green's IBPRE patch. The PKG method creates the master key, which the user then uses to make their private key in the first patch. This method is not bulletproof, and it does not safeguard the PKG from malicious attacks or guarantee that it is genuine. If the master key is compromised, the user's private key is also at risk. Using a distributed key generation (DKG) algorithm, this paper enhances the IBPRE solution better to protect the privacy of its users during key generation. Since the private key for each user is generated using the secret value and agreed upon by all users, the key's security is ensured even if a user is maliciously attacked. There are five stages to the data-sharing protocol: setup, key generation, data storage, data transfer, and decryption.

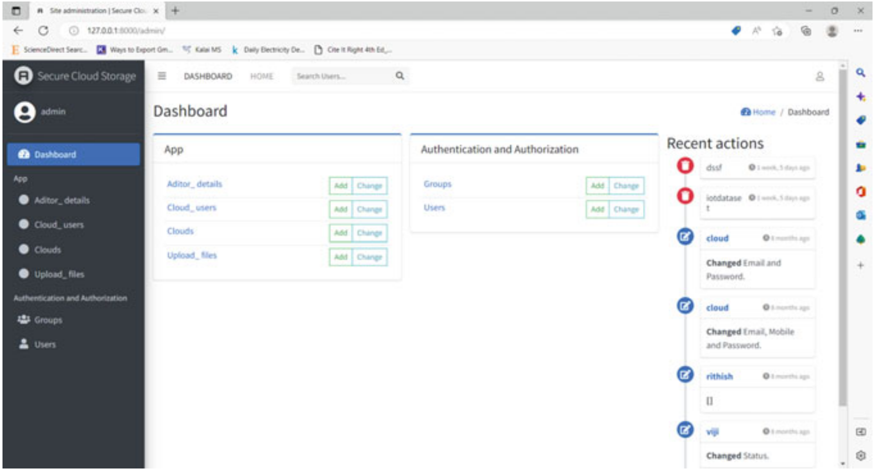
- First, enter the server's security settings, and then have the server build the system's public settings. Each participant in the blockchain is responsible for establishing its polynomial algorithm after acquiring the generic parameters. Then, create the master key with the help of the DKG algorithm. Each participant in the blockchain then chooses a polynomial algorithm at random and broadcasts the result of that algorithm's calculation. Each node in the blockchain simultaneously transmits a secret value to all other nodes, and the node receiving the value checks its authenticity. If the remote value verification fails, an error prompt will be sent back to the sender, who will then retry sending the secret value until it is accepted. The master key will be generated once all individual blockchain participants have been verified trustworthy.
- In the critical generation process, each participant in the blockchain provides a private key based on the parameters and identifier given.
- Whenever sensitive information is generated, the data's keeper encrypts it with the public key, combines the identification id and plaintext to form the first-layer ciphertext, and then saves it in a distributed database. The data owner then broadcasts the transaction after digitally signing the sensitive data and saving it to a file with its hash value, storage location, access control strategy, etc. Once the relevant nodes confirm the transaction, it will be added to the blockchain.
- When data users need access to private information, they must submit a signed request to the data owner. The message of request lets the shop owner verify the customer's identity. Second, the data holder will use the consumer's identity id and private key to generate a hybrid encryption key following the file access control strategy if the consumer's identity is valid and has read rights. Sharing the re-encryption key and storage location with the proxy service provider is essential. The proxy service provider encrypts the ciphertext again using the re-encryption key. This fresh ciphertext represents the second-layer ciphertext. Finally, the second layer of ciphertext is delivered to the recipient.
- Data Decryption: When a data consumer receives ciphertext encrypted with a second layer, he may use his private key to decode the ciphertext and recover the original plaintext file.

Creating Registration Form/ Cloud User Login Form:

Make a web-based signup sheet with HTML and CSS, and include fields for a user's name, email address, physical address, and phone number. A user's login credentials for the cloud service may be entered. The server will then enter its credentials to access the cloud-based resource.

In Fig. 3, a cloud user page has been added to the control panel. The authentication and authorization information for a given user is displayed here for perusal by the administrator. This page allows the administrator to manage cloud users, including adding and removing users (Fig. 4).

In Fig. 5, the user's browser will see the cloud files' download page. Selecting Download Decrypt File from the cloud files' menu decrypts the file after the user has obtained the document's private and public keys. This window will take you to the file's download page.



Dashboard

App		
Aditor_ details	Add	Change
Cloud_ users	Add	Change
Clouds	Add	Change
Upload_ files	Add	Change

Authentication and Authorization		
Groups	Add	Change
Users	Add	Change

Recent actions

dssf

iotdatase t

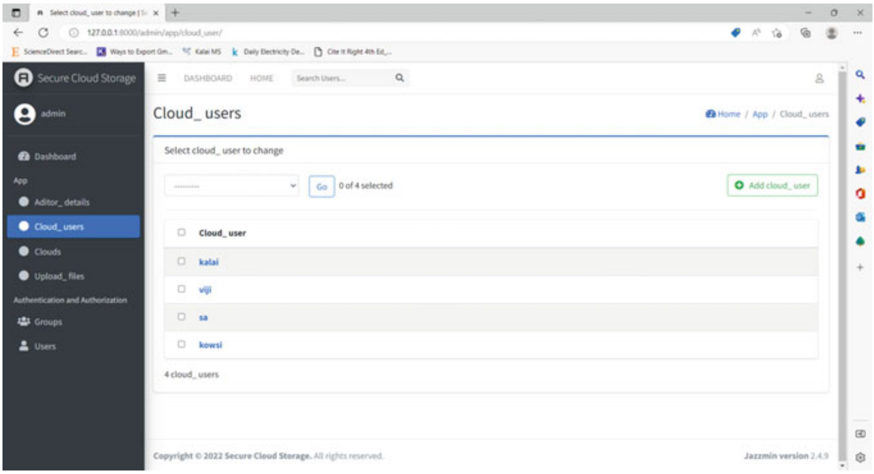
cloud
Changed Email and Password.

cloud
Changed Email, Mobile and Password.

rithish
[]

viji
Changed Status.

Fig. 3 Safe online data storage control panel



Cloud_ users

Select cloud_ user to change

0 of 4 selected

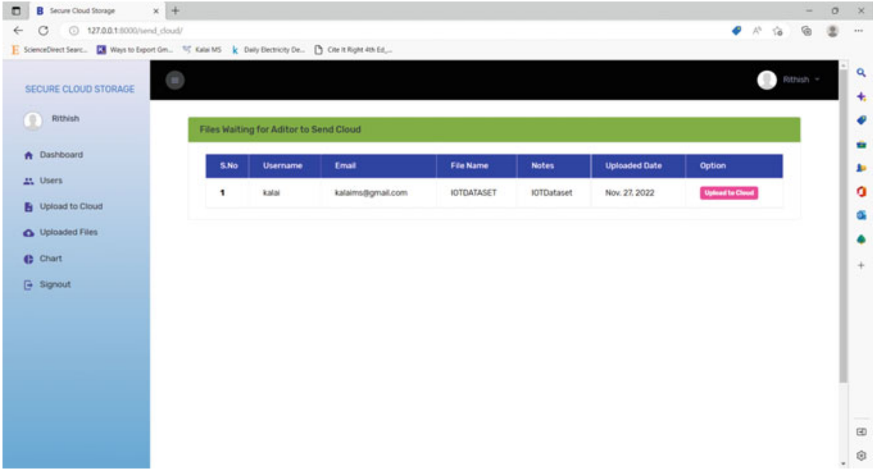
□	Cloud_ user
□	kalai
□	viji
□	sa
□	kowsi

4 cloud_ users

Copyright © 2022 Secure Cloud Storage. All rights reserved.

Jazzmin version 2.4.9

Fig. 4 Cloud administration console user interface



Files Waiting for Aditor to Send Cloud

S.No	Username	Email	File Name	Notes	Uploaded Date	Option
1	kalai	kalaims@gmail.com	IOTDATASET	IOTDataset	Nov. 27, 2022	Upload to Cloud

Fig. 5 Cloud files page in auditor

11 Conclusion

The fundamental goal of this research is to protect data privacy and cloud storage data in outsourced ways. Propose an unbiased auditor to audit secure data to safeguard the data. All CS security requirements need access management to prevent unauthorized access to the cloud server system and to maintain data confidentiality. A hybrid framework architecture for data security is proposed in this research. By utilizing CS storage solutions, customers can transfer their data centers and applications to colossal central data centers, eliminating the need for recurring maintenance and repairs on their local data storage devices. Nonetheless, this creates significant problems protecting and maintaining file storage dependability compared to untrustworthy providers.

References

Anil, Mannepalli PK (2021) Achieving effective secrecy based on blockchain and data sharing in cloud computing. In: 2021 10th IEEE international conference on communication systems and network technologies (CSNT), pp 709–716. https://doi.org/10.1109/CSNT51715.2021.9509663

Apirajitha PS, Renuka DR (2021) A survey on digital forensics in cloud environment using blockchain technology. In: 2021 4th international conference on computing and communications technologies (ICCCT), pp 160–163. https://doi.org/10.1109/ICCCT53315.2021.9711768

Awad AS, Yousif A, Kadoda G (2019) Enhanced model for cloud data security based on searchable encryption and hybrid fragmentation. In: 2019 international conference on computer, control, electrical, and electronics engineering (ICCCEEE), pp 1–4. https://doi.org/10.1109/ICCCEEE46830.2019.9070918

Awadallah, Samsudin A (2021) Using blockchain in cloud computing to enhance relational database security. IEEE Access 9:137353–137366. https://doi.org/10.1109/ACCESS.2021.3117733

Deepthi A, Ramani G, Deepika R, Shabbeer M (2021) Hybrid secure cloud storage data based on improved encryption scheme. In: 2021 international conference on emerging smart computing and informatics (ESCI), pp 776–779. https://doi.org/10.1109/ESCI50559.2021.9396842

Manjula GS, Meyyappan T (2022) DPMM: data privacy and memory management in big data server using hybrid hashing method. In: 2022 international conference on automation, computing and renewable systems ICACRS-2022. https://doi.org/10.1109/ICACRS55517.2022.10029073

A Dataset on Digital Image Forgery Detection

Priyanka Kumari and Kishore Kumar Senapati

1 Introduction

Forensic digital technology may be an advocacy branch called cybercrime. This involves essentially detection, retrieval, and research of the fabric found in digital devices. In general, digital forensics are called forensic computers. Today, this is not only connected to the pc device as the forensic company has expended on the investigation of all the digital data storage ready devices. Digital forensics means that electronic data are detected and interpreted during the courts. The aim of this method is to maintain all proofs in its original form while conducting a structured study by collecting, identifying, and validating digital information in order to reconstruct past occurrence (Fig. 1).

Forensic research on digital media devices, detection and investigation of theft of intellectual property, fraud detection, e-discovery of digital evidence confirms alibis or statements and intentionally identify sources, e.g. in the case of copyright, and authenticate documents (Kumari et al. 2021).

Digital images are classified as passive or active in searching for falsification or artificial distortions. While the active falsification detection method uses the digital watermarks/signatures, the passive methods do not need any extra information to authenticate the images using image statistics. In an active approach, certain changes have been made in the process of image building where the digitally verified information is inserted into the image. This information helps to verify whether or not some portion of the picture is copied. Passive image forgery often involves two types of forgery: copy–move, where a certain part of the image is copied to another position within the same imagery, and splicing, where some images are copied to a different location inside the different images (Battiato et al. 2022; Fig. 2).

P. Kumari (✉) · K. K. Senapati
Birla Institute of Technology, Mesra, Ranchi, India
e-mail: priyanka.kmri1993@gmail.com

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_30

CG
CG
REAL
CG
REAL
CG
REAL
CG
REAL
REAL
"Seeing is not believing"
First manipulated image.[1840]
thispersondoesnotexist.com
FAKE

Fig. 1 Example of a digital forgery

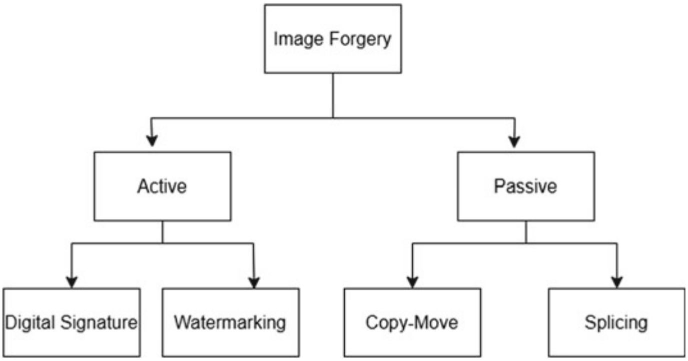


Fig. 2 Types of image forgery

A. Active Approach

The active approach involves adding preprocessing elements like embedded watermarks or generated signatures during image creation. However, this practical implementation limits their use. Digital watermarking and signatures are primary methods where data are added during image acquisition. Tampering is detectable when specific data cannot be extracted from the image. Watermarking detects tampering by embedding a security structure in the image. Yet, modern imaging devices often lack these features. New methods resembling watermarking, such as message authentication codes, image hashing, checksums, and image shielding, have emerged for image security.

B. **Passive Approach**

Passive approach in image forgery detection refers to a method where manipulation or tampering within an image is identified without altering the image itself. Passive techniques do not modify the image to include any specific markers or hidden information; instead, they rely on patterns, inconsistencies, or anomalies that might arise due to the manipulation process. These anomalies could include variations in noise patterns, inconsistencies in lighting and shadows, or irregularities in the distribution of certain image features. The advantage of passive approaches is that they do not disrupt the image content and can identify a broader range of manipulations, including those that might be designed to bypass active detection methods. However, passive methods might be limited in their ability to detect very subtle or sophisticated forgeries, as they rely on observable patterns rather than explicit embedded information.

Here are a few scenarios where passive analysis could come into play:

Copy–Paste Forgery: An attacker might have copied an object from one part of the image and pasted it into another, to create a misleading impression. In a passive approach, you could analyze the lighting and shadows on the object and the background to identify any inconsistencies. If the lighting angles and shadows do not match, it could indicate manipulation.

Splicing: If two different images were combined by splicing them together, there might be subtle colour variations or noise patterns at the splice boundary. Passive analysis would involve examining the image for abrupt changes that might not naturally occur in an unaltered photograph.

The detection of copy–move forgery (CMFD) is often subdivided into two main categories, block-based and keypoint-based. The block method divides the image in blocks of fixed size and extract overlaps or does not overlap and matches the features of the blocks. The methods of blocking generally utilize the techniques of feature extraction. There is no subdivision of blocks in keypoint-based methods. It is based on the extraction of photo features in certain regions (Battiato et al. 2022). Copying–moving falsification, which can be a passive method, is often made without the use of any Photoshop expertise.

2 Existing Research Methodologies for Digital Image Forgery Detection

A digital imaging fiction detection was proposed based on the Expectation Maximizing Algorithm by Mohammed Hazim Alkawaz. In this paper, the EM mismanagement rule analyzes a digital image. The EM rule could be a two-step procedure. Estimate the probability of all samples of happiness for each model in the E-step.

The precise kind of correlations between samples will be evaluated within the M step and a random α can begin. Multiple EM versions are available. Because the result is a swish change from the original region between frequencies within the managed area (Alkawaz et al. 2020).

A keypoint copy–move forgery detection method based on the Efficient Keypoint has been proposed by Sunitha K. This paper shows an efficient CMFD technique using hybrid function extraction, detection, and hierarchical clusters. In this paper, you will find out more about the forgery detection technique. First, the picture is divided into the same patch or block size. This is then used to extract keypoints using an SURF detector and SIFT descriptor. For the matching characteristic points, the agglomerative hierarchical clustering method is then used. Finally, the RANSAC image transformation is used to eliminate mistreats (i.e. outliers) (Thakur and Rohilla 2020).

Based on statistical features, SB G Tilak Babu proposed an Optimized Copy–Move Forgery Detection Process. Steerable Pyramid Transform (SPT), Grey-Level Co-occurrence Matrix (GLCM), and Optimized Naive–Bayes Classifier (ONBC) are used to propose the CMFD method in this paper. The suspected picture is given to SPT to get different directions, and GLCM features are extracted from all thought picture directions. These features are also required for the ONBC classification. The proposed algorithm's performance is measured in terms of true-positive TPR and false-positive FNR across a wide range of CoMoFoD tests, MICC F, and CASIA V1.0 databases. These tests, as well as the cast image, which has been the target of numerical postprocessing attacks, have demonstrated their superiority over current CMFD techniques (Tilak Babu and Rao 2020).

Navdeep Kanwal proposed the use of fast Fourier transformation and local features for the detection of digital image forgery. The paper presents a thorough comparative analysis of the use of local texture descriptors for the detection of falsification, e.g. the image's local ternary pattern (LTP) and binary pattern (LBP). Using the existing block-based approach, the paper also proposes a method for combining the fast Fourier transform (FFT) and local descriptors to detect false images. Testing the performance of the CASIA v1.0 bench marked dataset for the technology(s) and descriptor(s). The results shall be assessed with the accuracy and recall of standard detection metrics (Sridevi and Ramya Krishna 2018).

Anuja Dixit has proposed a multi-copy-moved image forgery detection application for HOG-SVD-based features with Connected Component Labelling. During this paper, he proposes a block approach for the detection of copy-moving images to secure the information transmitted through the image by identifying the images casted and by stopping the propagation of five manipulated materials. The proposed model divides suspect image into blocks that overlap. It has extracted block characteristics using histograms of oriented gradients (HOG), (SVD). Lexicographic sorting is carried out over function matrix followed by Euclidean distance calculation for similar characteristics. Connected component labelling is used to get rid of the wrong match detection. In the system, it achieved the highest F measure than in the past, multiple move–copies with the contrast adjustment, colour reduction, and the fluttering attacks of the image when forged images sustain multiple copy–move (Maji et al. 2020).

Noor S. Farhan proposed an Image Plagiarism System for Forgery Detection in Maps Design (Soni and Biswas 2018). In this paper, he proposes an adaptable, scalable and extendable, robust imaging plagiarism method that has been tested in architectural/technology department designs, The information (design images) entered into the system is primarily compared using this method to datasets stored in the database mainly as a feature that is made through the Invariant Feature Transformation Algorithm (SIFT) which is one in synthesized intelligence algorithms and which matches the K-mean cluster and Brute Force L1 match. We are able to construct a robust system that does not discard any extracted images thanks to the utilization of the k-mean clustering algorithm, which is an effective AI clustering algorithm. In this article, 45 sample sets are saved and used in the system as training frames as a result of the system database and the use of 48 samples as test frames containing original and falsified designs. These images with an equivalent rate of 90% and a corresponding accuracy rate of 81% were evaluated (Soni and Biswas 2018).

The Modern Approach for Image Forgery Detection based on the Normed Mean and Standard Deviation (Alkawaz et al. 2020) is proposed by G. Nirmala and K.K. Thyagharajan. This paper is used for the detection of falsification and the classification of associated forged images by the BIRCH algorithm as clusters based on a hierarchical procedure (Nirmala and Thyagharajan 2019).

3 Comparative Analysis of Digital Image Forgery Detection Methodology

The varied techniques/study parameters (such as features and a classifier) and algorithms used by various researchers are indexed. We have talked about various authors' suggestions for image forgery detection methods. The intention of the multitude of strategies is to identify the forgery inside the picture; however, the methods are different.

In this paper, we first introduce the objectives, challenges, and approaches to building altered datasets.

With the goal of verifying image content, authenticity detection was called for and much work has been done in this area (Battiato et al. 2022; Hiremani and Senapati 2022). From year 2011 to year 2021, over 350 papers about digital forensics are published according to our records. From Fig. 3, we can notice that much work has been done on image tampering detection.

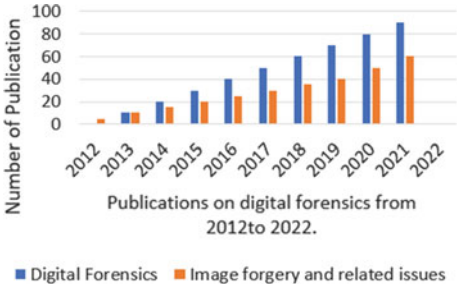


Fig. 3 Statistics of publications on digital forensics from 2011 to 2022

4 Suggested Methodology

Many methods are used to forge images, a new IFD process pipeline is introduced. Unlike the existing framework, the techniques are integrated into a single framework. It gives the explanation of the proposed method for image forgery using copy–move and splicing techniques. The architecture explanation is as follows (Fig. 4):

This step is used to remove noise by preparing the image for the next step after that image is divided into square and circle blocks, after feature extraction each block are computed. This feature will later be compared with the same features from the set of the earlier stored feature in the feature database. It is used to compute the distance measure between the feature from the query image and each feature vector of the database images. Few methods classified to the selected feature after classification user can get manipulated or authentic image output.

Zernike moment: Over a unit circle, Zernike moments can be defined as a collection of intricate orthogonal polynomials. A global shape descriptor is the Zernike

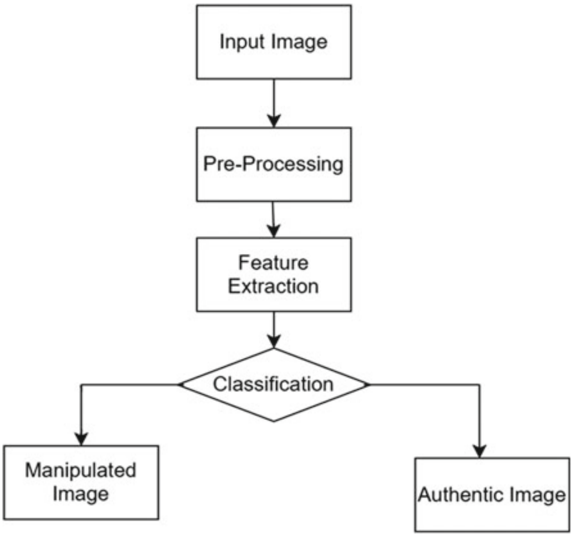


Fig. 4 Architecture of image forgery detection

moment (Dixit and Bag 2019). SURF: A patented local feature detector and descriptor are called speeded-up robust features (SURFs). The scale-invariant feature transform (SIFT) descriptor serves as a partial source of inspiration. SURF employs an integer approximation of the determinant of the Hessian blob detector to identify interest points. This approximation can be calculated in three integer operations using a precomputed integral image.

4.1 Zernike Moments-Based Method

The complex ZM is derived from Zernike polynomials, which are a set of complex, orthogonal polynomials defined over the interior of a unit circle (Dixit and Bag 2019). Figure 5 depicts the Zernike polynomials. Zernike moments have many desirable properties, including rotation invariance, noise resistance, and expression efficiency. $x2 + y2 = 1$.

Where n can be zero or positive integer such that $(n \geq 0)$ and m can be positive or negative satisfying the constraints. Zernike moments in the context of image forgery detection provide a mathematical framework for quantifying spatial features in an image. By comparing these features with those of authentic images, it is possible to identify potential forgeries or alterations. Keep in mind that while Zernike moments can be a useful tool, they are typically used as part of a broader arsenal of techniques for image forgery detection.

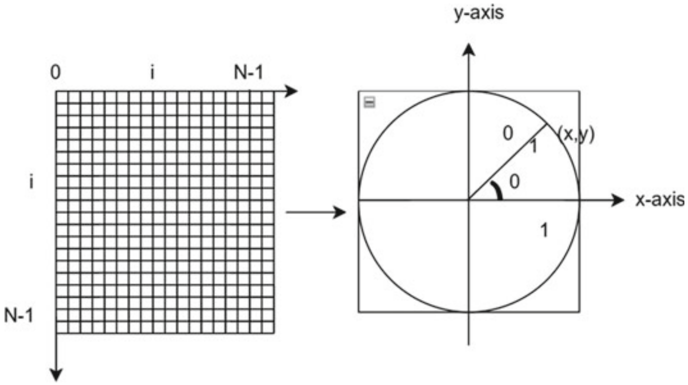


Fig. 5 Square-to-circular image transformation

5 Dataset Description

The altered dataset has a total of 1000 images, with 100 images belonging to each of ten distinct categories: Buildings, Food, Dishes, People, Historical Buildings, Cats, Shadows, Rooms, Roses, and Mountains.

5.1 Data Collection Process

The altered images are collected through offline mode. I have selected a few camera devices through which we processed the collected images for creating a novel dataset. IPHONE 5S, OPPO RENO 6 PRO, XIOMI 12 PRO, IPHONE 12 MINI. Standard tools used for manipulating images include Photoshop (GIMP, ImageJ), Adobe Photoshop, and Paint Shop Pro.

Image Splicing: In this procedure, a forged picture is produced by copying and pasting a portion of another picture into another picture. Pipeline and example of this technique are given in Figs. 6 and 7.

- **Copy–Move Image Forgery:** It refers to the copying and pasting of an object from one region of an image to another. Copy–move images are modified after

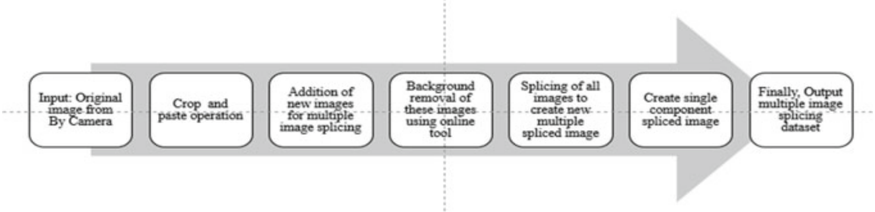
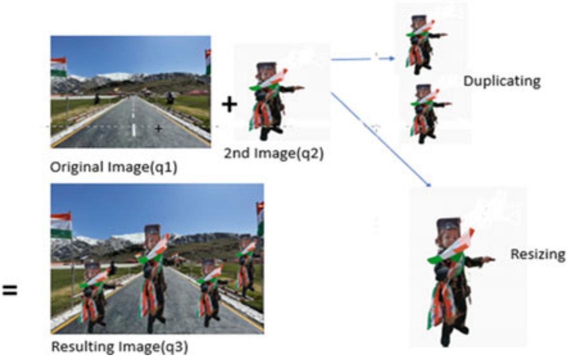

Input: Original image from By Camera
Crop and paste operation
Addition of new images for multiple image splicing
Background removal of these images using online tool
Splicing of all images to create new multiple spliced image
Create single component spliced image
Finally, Output multiple image splicing dataset

Fig. 6 Pipeline of image splicing

Fig. 7 Example of image splicing



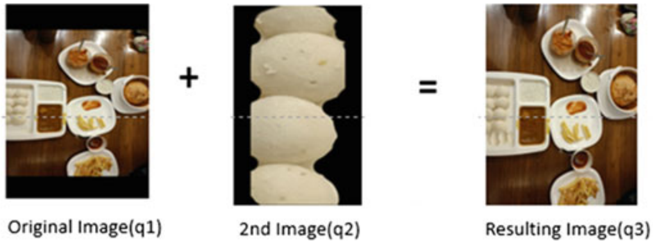


Fig. 8 Example of copy–move image forgery

Fig. 9 Example of dataset

being copied, rotated/flipped, and scaled before being pasted somewhere else. This can be used to hide details or take them out entirely (Fig. 8).

This database consists of 2000 multiclass images of .jpg format in ten groups each of 200 images of the size 256 × 384 or 384 × 256. Images in same group are thought to be similar. If the query is from one of the ten classes that the images are divided into, it is almost certain that the user wants to find the other images in that class. Due to its size and availability of class information, this database was extensively utilized for feature testing. Figure 9 demonstrates the altered dataset example.

6 Result and Discussion

How to authenticate an image?

We can authenticate an image by (1) Visual inspection, (2) File analysis, (3) Global analysis, (4) Local analysis. We discuss the source information result of different processes of alterations in images which is as under: Image Filename Alterations, Image Size Alterations, Metadata Alterations. A novel result by using copy–move, splicing forgery techniques. The combination of colour, texture and shape features. As shown below the Result for retrieval to a selected query image.

1. Visual inspection results: enhancement/filtering or other image processing techniques can help the analysis (Fig. 10).
2. File analysis results: camera, file format and structures, metadata (EXIF), compression parameters (quantization tables; Figs. 11, 12, and 13).

Fig. 10 Example of VI

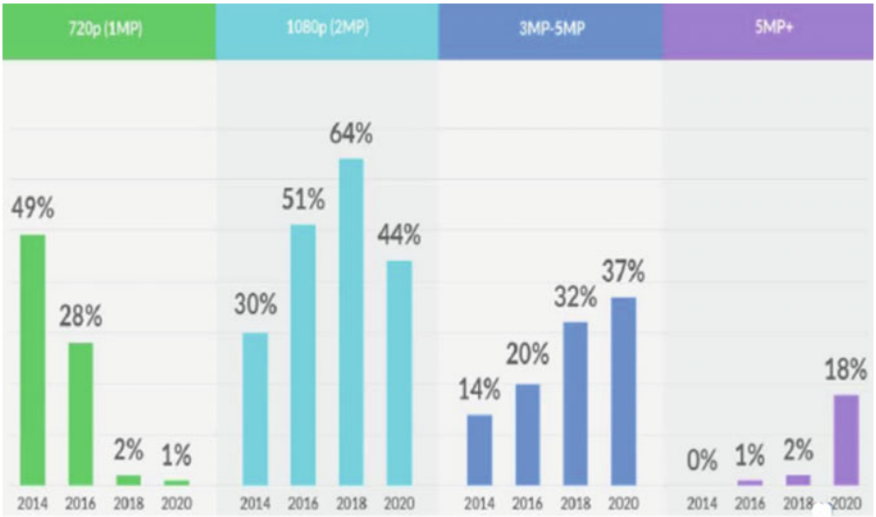
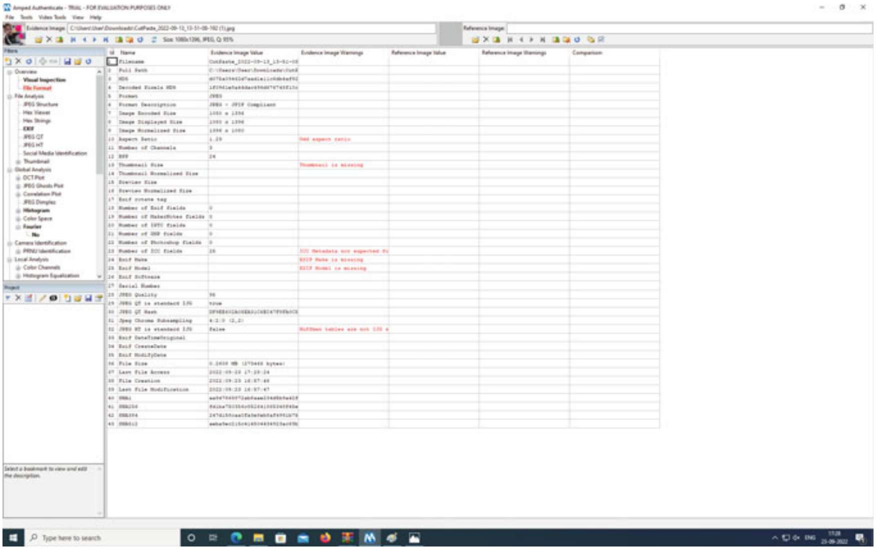
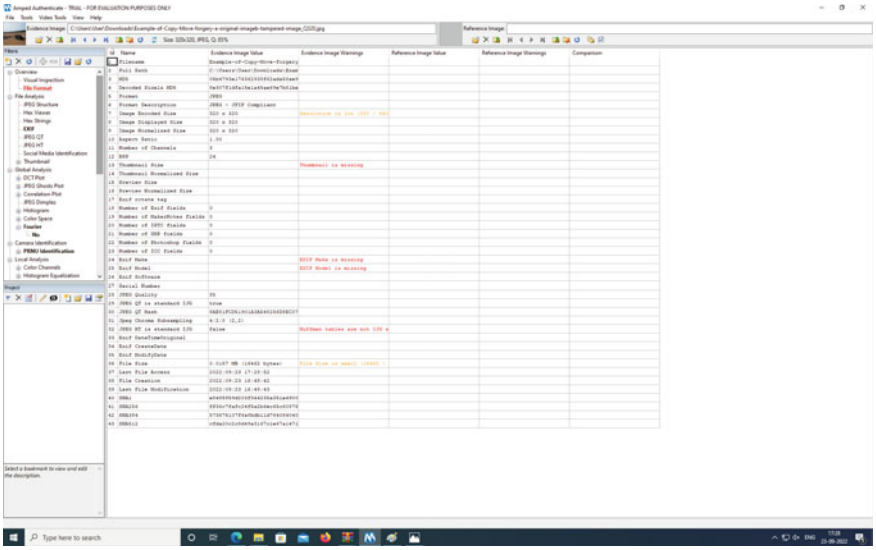


Fig. 11 Camera resolution most commonly used



Id	Name	Evidence Image Value	Evidence Image Warnings	Reference Image Value	Reference Image Warnings	Comparison
1	Filename	CutPaste_2022-09-13_13-51-08				
2	Full Path	C:\Users\User\Downloads\CutP				
3	MD5	[illegible]				
4	Decoded Pixels MD5	[illegible]				
5	Format	JPEG				
6	Format Description	JPEG - JFIF Compliant				
7	Image Encoded Size	1080 x 1396				
8	Image Displayed Size	1080 x 1396				
9	Image Normalized Size	1396 x 1080				
10	Aspect Ratio	1.29	Odd aspect ratio			
11	Number of Channels	3				
12	BPP	24				
13	Thumbnail Size		Thumbnail is missing			
14	Thumbnail Normalized Size					
15	Preview Size					
16	Preview Normalized Size					
17	Exif rotate tag					
18	Number of Exif fields	0				
19	Number of MakerNotes fields	0				
20	Number of IPTC fields	0				
21	Number of XMP fields	0				
22	Number of Photoshop fields	0				
23	Number of ICC fields	18	ICC Metadata not expected fo			
24	Exif Make		EXIF Make is missing			
25	Exif Model		EXIF Model is missing			
26	Exif Software					
27	Serial Number					
28	JPEG Quality	95				
29	JPEG QT is standard IJG	true				
30	JPEG QT Hash	[illegible]				
31	Jpeg Chroma Subsampling	4:2:0 (2,2)				
32	JPEG HT is standard IJG	false	Huffman tables are not IJG s			
33	Exif DateTimeOriginal					
34	Exif CreateDate					
35	Exif ModifyDate					
36	File Size	0.2430 MB (254848 bytes)				
37	Last File Access	2022:09:23 17:28:24				
38	File Creation	2022:09:23 16:57:46				
39	Last File Modification	2022:09:23 16:57:47				
40	SHA1	[illegible]				
41	SHA256	[illegible]				
42	SHA384	[illegible]				
43	SHA512	[illegible]				

Fig. 12 Result snapshot for metadata to a selected query image



Id	Name	Evidence Image Value	Evidence Image Warnings	Reference Image Value	Reference Image Warnings	Comparison
1	Filename	Example-of-Copy-Move-Forgery				
2	Full Path	C:\Users\User\Downloads\Exam				
3	MD5	[illegible]				
4	Decoded Pixels MD5	[illegible]				
5	Format	JPEG				
6	Format Description	JPEG - JFIF Compliant				
7	Image Encoded Size	320 x 320	Resolution is low (320 * 320			
8	Image Displayed Size	320 x 320				
9	Image Normalized Size	320 x 320				
10	Aspect Ratio	1.00				
11	Number of Channels	3				
12	BPP	24				
13	Thumbnail Size		Thumbnail is missing			
14	Thumbnail Normalized Size					
15	Preview Size					
16	Preview Normalized Size					
17	Exif rotate tag					
18	Number of Exif fields	0				
19	Number of MakerNotes fields	0				
20	Number of IPTC fields	0				
21	Number of XMP fields	0				
22	Number of Photoshop fields	0				
23	Number of ICC fields	0				
24	Exif Make		EXIF Make is missing			
25	Exif Model		EXIF Model is missing			
26	Exif Software					
27	Serial Number					
28	JPEG Quality	85				
29	JPEG QT is standard IJG	true				
30	JPEG QT Hash	[illegible]				
31	Jpeg Chroma Subsampling	4:2:0 (2,2)				
32	JPEG HT is standard IJG	false	Huffman tables are not IJG s			
33	Exif DateTimeOriginal					
34	Exif CreateDate					
35	Exif ModifyDate					
36	File Size	0.0137 MB (14462 bytes)	File Size is small (14462)			
37	Last File Access	2022:09:23 17:28:42				
38	File Creation	2022:09:23 16:49:42				
39	Last File Modification	2022:09:23 16:49:43				
40	SHA1	[illegible]				
41	SHA256	[illegible]				
42	SHA384	[illegible]				
43	SHA512	[illegible]				

Fig. 13 Result snapshot for metadata to a selected query image

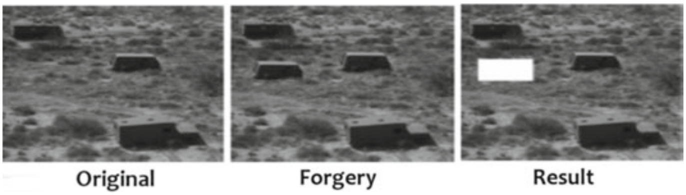


Fig. 14 Two examples of generating the copy–move tampered images

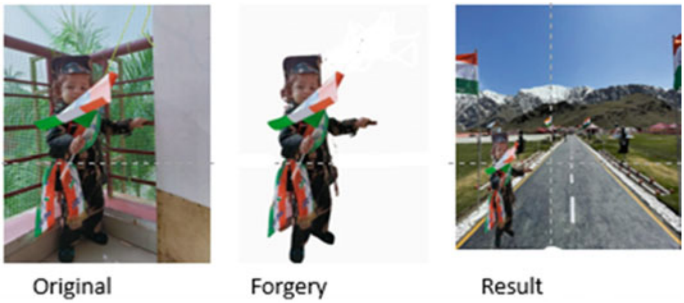


Fig. 15 Two examples of generating the splicing tampered images

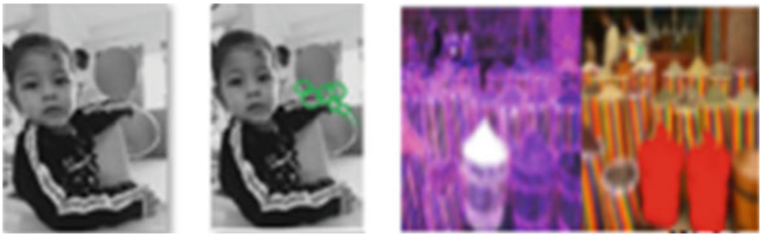


Fig. 16 Example of local analysis

3. Global Analysis Results: Pixel and compressed data statistics (Figs. 14 and 15).
4. Local Analysis Results: Finding inconsistencies of pixel statistics across the image (Fig. 16).

7 Conclusion and Future Scope

The goal of the novel work is creation of novel dataset with the varied aspects of image forgery detection. Exploration and Creation of a Dataset for the Carried Research. There is no single standard method for evaluating the authenticity and integrity of the image.

Further, the proposed method is compared with existing techniques in terms of performance. To understanding the feature extraction techniques, the tool/method needs an original image in order to detect tampering and exploring the possibility to modify the script/technique/procedure to run the detection algorithm without having the original image.

References

Alkawaz M, Veeran M, Bachok R (2020) Digital image forgery detection based on expectation maximization algorithm. In: 2020 16th IEEE international colloquium on signal processing & its applications (CSPA 2020), 28–29 Feb 2020, Langkawi, Malaysia

Battiato S, Giudice O, Guarnera F, Puglisi G (2021) First quantization estimation by a robust data exploitation strategy of DCT coefficients. IEEE Access 9:73110–73120

Battiato S, Giudice O, Guarnera F, Puglisi G (2022) CNN-based estimation of the first quantization matrix from double compressed JPEG images. J Vis Commun Image Represent

Dixit A, Bag S (2019) Utilization of HOG-SVD based features with connected component labeling for multiple copy-move image forgery detection. In: 2019 IEEE 5th international conference on identity, security, and behavior analysis (ISBA)

Farhan NS, Abdulmunem ME, Abid-Ali M (2019) Image plagiarism system for forgery detection in maps design. In: 2019 2nd scientific conference of computer sciences (SCCS), University of Technology—Iraq

Farid H (2006) Digital doctoring: how to tell the real from the fake. Significance 3(4):2–4

Hiremani VA, Senapati KK (2022) Developing real-time training dataset for human racial classification. Multimed Tools Appl 1–25

Johnson MK, Farid H (2007) Detecting photographic composites of people. In: Proceedings of the 6th international workshop on digital watermarking, Guangzhou, China, 2007

Kanwal N, Girdhar A, Kaur L (2019) Detection of digital image forgery using fast Fourier transform and local features. In: 2019 international conference on automation, computational and technology management (ICACTM), Amity University

Kim D-H, Lee H-Y (2017) Image manipulation detection using convolutional neural network. Int J Appl Eng Res 12(21):11640–11646

Kumari P, Senapati KK (2021) Analysis of image forgery detection using convolutional neural network (CNN). NCCS-2021

Maji P, Pal M, Ray R, Shil R (2020) Image tampering issues in social media with proper detection. In: 2020 8th international conference on reliability, infocom technologies and optimization (trends and future directions) (ICRITO), Amity University, Noida, India, 4–5 June 2020

Muzaffer G, Ulutas G (2019) A new deep learning-based method to detection of copy-move forgery in digital images. 978-1-7281-1013-4/19/$31.00 ©2019 IEEE

Nirmala G, Thyagharajan KK (2019) A modern approach for image forgery detection using BRICH clustering based on normalised mean and standard deviation. In: International conference on communication and signal processing, 4–6 Apr 2019, India

Shiva Kumar BL, Santosh S (2010) Detecting copy move forgery in digital images: a survey and analysis of current methods. Global J Comput Sci Technol 10

Soni B, Biswas D (2018) Image forensic using block-based copy-move forgery detection. In: 2018 5th international conference on signal processing and integrated networks (SPIN)

Sridevi T, Ramya Krishna B (2018) A novel approach for detection of copy-move forgery detection using transform domain. In: Proceedings of the 2nd international conference on inventive communication and computational technologies (ICICCT 2018)

Sunitha K, Krishna AN (2020) Efficient keypoint based copy move forgery detection method using hybrid feature extraction. In: Proceedings of the second international conference on innovative mechanisms for industry applications (ICIMIA 2020)

Thakur R, Rohilla R (2020) Recent advances in digital image manipulation detection techniques: a brief review. In: Proceedings of the forensic science international, ECE Department, Delhi Technological University, Delhi 110042, India, 3 Apr 2020

Tilak Babu SBG, Srinivasa Rao Ch (2020) Statistical features based optimized technique for copy move forgery detection. In: 11th ICCCNT 2020, 1–3 July, 2020—IIT—Kharagpur

Detection of Pneumonia from Chest X-ray Using Deep Learning

K. N. Chaithra, Shreyan P. Shetty, P. Raji, Aditya Datta, K. S. Sandeep, and Anikait Targolli

1 Introduction

Infections that are not treated lead to pneumonia. The major signs of pneumonia are cough and shortness of breath. Pneumonia is one of the main signs of COVID-19, making early detection of the condition crucial (Chowdhury et al. 2020). The MRI, CT scan, and X-ray are the three most popular medical imaging techniques. However, X-ray is the measurement technique that is the most economical. Radiologists have an extremely tough time using X-rays to identify pneumonia. Consequently, a deep learning algorithm is required to correctly detect the illness. There are two types of pneumonia: bacterial and viral. White inflammatory fluids fill the alveoli during bacterial pneumonia. In viral pneumonia, the leftmost portion of white dots is filled. The tell-tale signs of COVID-19 infection are bilateral hazy lung opacities and air space consolidation (Raji and Prashantha 2021).

Medical imaging benefits from artificial intelligence, machine learning, and deep learning. With fewer false positives, deep learning has been demonstrated to be particularly effective at detecting pneumonia (Khan et al. 2021). The availability of a big training dataset is the primary determinant for improved accuracy in deep learning. In order to improve data augmentation because there aren't enough chest X-ray pictures available, we use a generative adversarial network (GAN) (Goodfellow et al. 2014). For data augmentation, images are created using a GAN, which then assesses the results (Gonog and Zhou 2019). Five distinct models were deployed like VGG19, Xception, Densenet121, and InceptionV3 (Ayan and Ünver 2019; Hammoudi et al. 2021; Victor Ikechukwu et al. 2021). Their performance metrics were assessed. Accuracy, precision, recall, and $F1$ score are among the performance indicators

K. N. Chaithra · S. P. Shetty (✉) · A. Datta · K. S. Sandeep · A. Targolli
Nitte Meenakshi Institute of Technology, Bangalore, India
e-mail: shreyanshetty25@gmail.com

P. Raji
AI/ML, CloudThat, Bangalore, India

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_31

Table 1 Dataset

	Training set	Test set
Bacterial pneumonia	2530	242
Viral pneumonia	1344	148
Normal	1341	234
Total	5215	624

taken into account. Then we classify our X-ray images into three categories, i.e., normal, bacterial, and viral pneumonia (Kermany et al. 2018).

1.1 Motivation

One or both lungs may experience inflammation due to pneumonia, which is typically brought on by a bacterial infection. X-rays have been useful in the early phases of diagnosis and screening patients with COVID-19 and have helped save many lives. To avoid some complications that result in death from pneumonia, it's critical to diagnose and treat the condition quickly. Artificial intelligence's primary technology is now deep learning (AI). In our work, we seek to provide a framework for categorizing patient chest X-ray pictures.

2 Dataset

We have used the dataset which was available open-source and was provided by Kermany et al. (2018) who gathered and labeled CXRs from the Women and Children's Medical Center in Guangzhou (Guangzhou, China). The dataset has three folders (train, test, val) and the train and test subfolders have two classes (bacterial pneumonia/viral pneumonia/normal). More detail about dataset is given in Table 1 and snapshot of the dataset is given in Fig. 1.

3 Methodology

In our proposed model shown in Fig. 2, the dataset containing unprocessed images is processed to a uniform size by data processing. In order to boost the dataset's variety, data augmentation techniques are applied. Then DCGAN technique is used to produce new data samples. The pretrained networks are trained using the dataset which has been processed. The performances of networks are evaluated and the best performing network is used for classification.

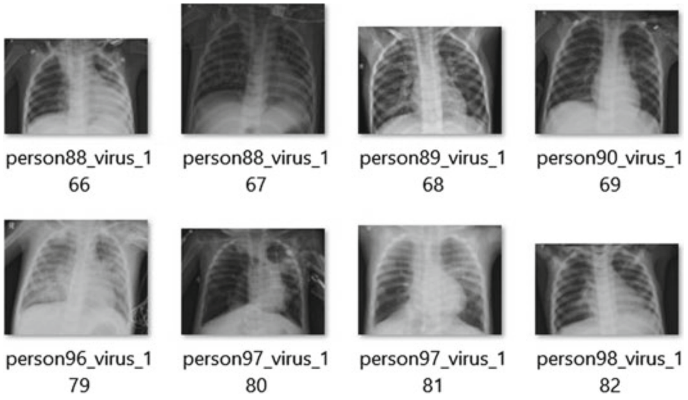


Fig. 1 Sample of dataset

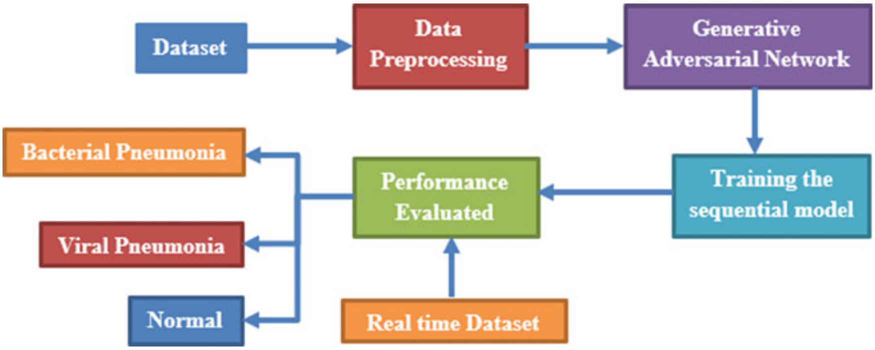


Fig. 2 Proposed methodology

3.1 Preprocessing Our Dataset

We normalize our images [0–1] and resize them to [224 × 224] so that we can suppress undesired distortion and enhance some features which are necessary for particular applications we are working for.

3.2 Generative Adversarial Network

Due to the high degree of imbalance in our data, there is a risk of overfitting. As a result, most machine learning algorithms will perform badly and need to be modified to prevent always predicting the majority class. So we are using DCGAN to generate

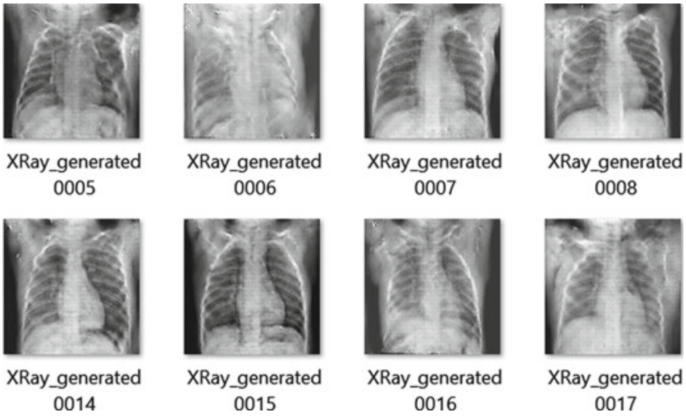


Fig. 3 Normal chest X-ray generated images

Table 2 Parameters for image augmentation

Method	Default	Adjusted
Horizontal flip	None	True
Shear range	0	0.2
Zoom range	0	0.2
Resize	–	224 × 224

1000 normal and viral chest X-ray images. Figure 3 shows the generated normal CXR images using DCGAN.

3.3 Image Augmentation

To eliminate further possibility of overfitting we are adjusting the image parameters as given in Table 2.

3.4 Selection of Pretrained Models

We selected four pretrained models VGG19, Xception, InceptionV3, and Densenet121. Firstly, we executed these four models without including the generated images in our training dataset. Then we have included the generated images in our training dataset. A comparison was done among them. Then the best three models are selected for further training.

3.5 Performance Evaluation

Accuracy, recall, precision, and F score are just a few of the performance indicators that are used to evaluate models. The suspected patients who are accurately identified as having a condition are called true positives (TP). The percentage of patients with normal lungs is known as true negatives (TN). False negative (FN) patients are those who have lung illness yet go undiagnosed. False positive (FP) refers to all patients who have been incorrectly identified as having lung conditions.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$F1\text{ - score} = 2 * \left[\frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}\right] \tag{4}$$

3.6 Classification

The best performing model was selected for classifying our images into three classes: normal, bacterial, and viral pneumonia. A graphical user interface was created using streamlit.

4 Result and Analysis

4.1 DCGAN Results

Based on the conducted experiments and results, we found that at the end of 200th epoch we found GAN accuracy to be 95% for normal image and 89% for viral images (Figs. 4, 5, 6 and 7).

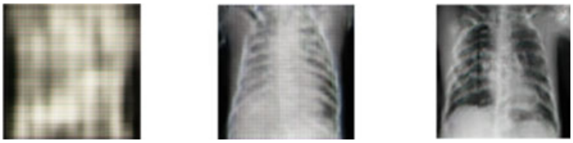


Fig. 4 Generation of viral chest X-ray images at 20th, 100th, and 200th epoch, respectively

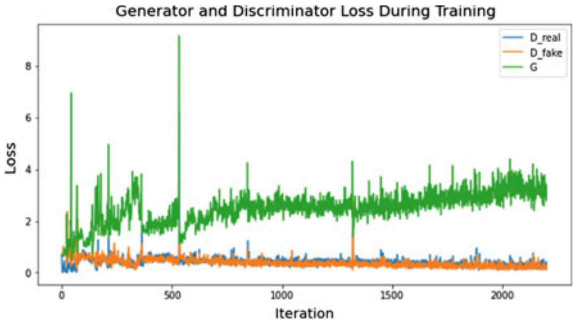


Fig. 5 Plot of generator and discriminator loss during training for viral generated chest X-ray

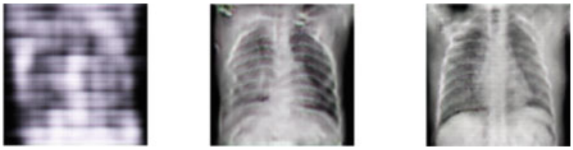


Fig. 6 Generation of normal chest X-ray images at 20th, 100th, and 200th epoch, respectively

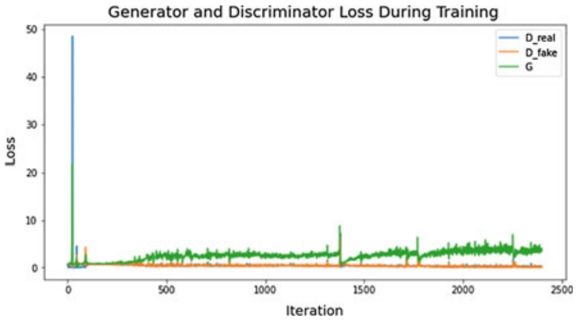


Fig. 7 Plot of generator and discriminator loss during training for normal generated chest X-ray

4.2 *Pretrained Models Results*

We have evaluated the performance of four CNN models, namely VGG19, Xception, InceptionV3, and Densenet121. Firstly, we trained our models without adding the generated images by GAN into the dataset. The models achieved an accuracy rate ranging from 57 to 74% after the 20th epoch as given in Table 3. Then after adding the generated images to the dataset, we have seen an improvement in our model accuracy rate, ranging from 70 to 83% after the 20th epoch as given in Table 4. So, we selected the best three models, i.e., VGG19, Xception, and Densenet121 and executed them for the 100th epoch as given in Table 5. From the experimental outcomes, we can observe that Xception has achieved the highest accuracy of 83% with a precision of 85%, recall of 83%, and $F1$ score of 83% (Figs. 8, 9, 10 and 11).

Table 3 Performance metrics results for pretrained models at the end of 20th epoch without GAN

	VGG19	Xception	Densenet121	InceptionV3
Accuracy	0.72	0.74	0.61	0.57
Precision	0.76	0.80	0.81	0.79
Recall	0.72	0.74	0.61	0.57
$F1$ score	0.73	0.74	0.63	0.58

Table 4 Performance metrics results for pretrained models at the end of 20th epoch with GAN

	VGG19	Xception	Densenet121	InceptionV3
Accuracy	0.83	0.80	0.78	0.70
Precision	0.83	0.81	0.81	0.76
Recall	0.83	0.80	0.78	0.70
$F1$ score	0.83	0.80	0.78	0.68

Table 5 Performance metrics results for pretrained models at the end of 100th epoch with GAN

	VGG19	Xception	Densenet121
Accuracy	0.80	0.83	0.80
Precision	0.83	0.85	0.81
Recall	0.80	0.83	0.81
$F1$ score	0.80	0.83	0.80

Fig. 8 Accuracy comparison between the pretrained models with and without using the generated images at the end of 20th epoch

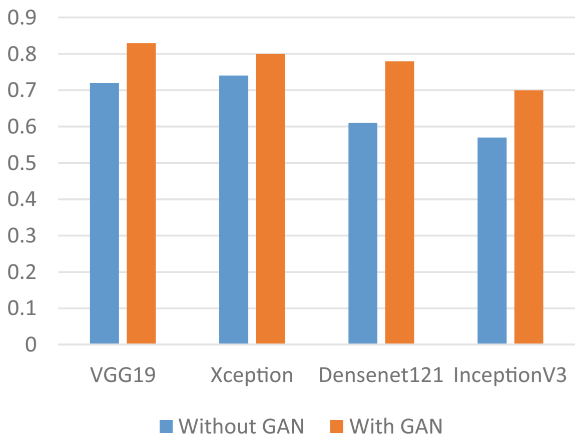


Fig. 9 Plot of performance metrics results for pretrained models at the end of 100th epoch

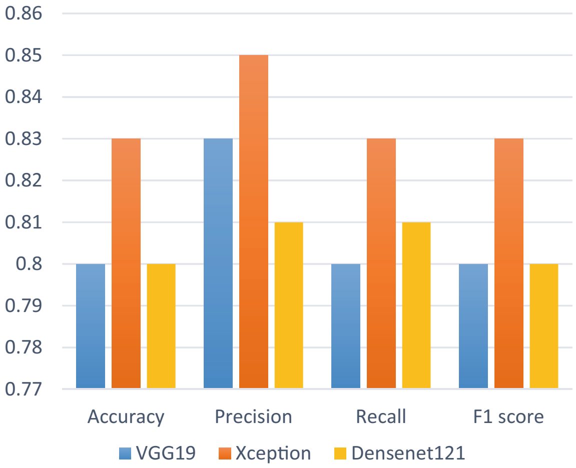


Fig. 10 Confusion matrix for Densenet121

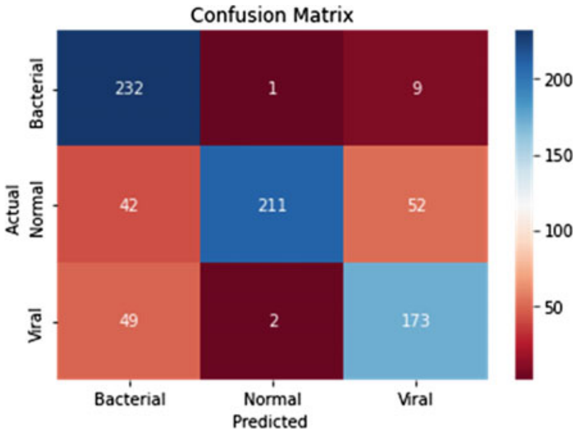


Fig. 11 Confusion matrix for VGG19

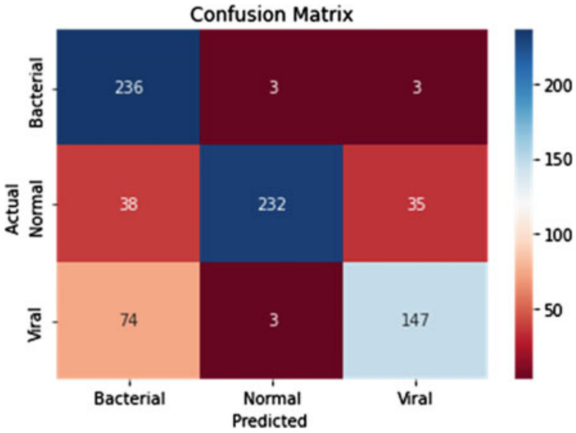
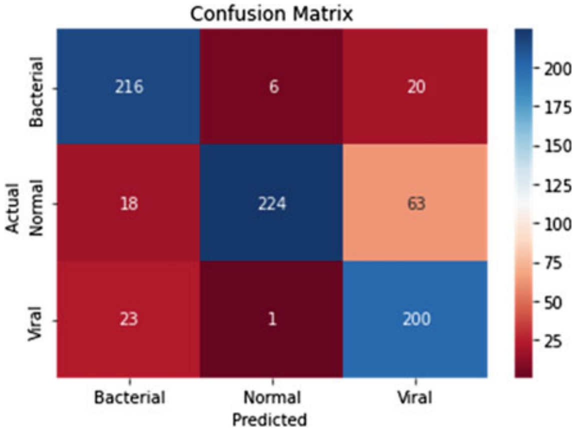


Fig. 12 Confusion matrix for Xception



5 Conclusion

In this project, we successfully classified the chest X-ray images. Also, we have created synthetic images using DCGAN (a transposed convolution technique) for data augmentation in order to eliminate imbalanced data due to overfitting. DCGAN was trained using 200 training epochs with NVidia Tesla T4 GPU and TensorFlow framework. It achieved a GAN accuracy of 95% for normal images and 89% for viral images (Fig. 12). Various transfer learning models, like VGG19, Xception, DenseNet121, and InceptionV3 are implemented for classifying our chest X-ray. This deep convolutional neural network was trained using 100 training epochs with NVidia Tesla P100 GPU and TensorFlow framework. We observed that Xception achieved the highest accuracy of 83% with a precision of 85%, recall of 83%, and $F1$ score of 83% at the end of 100th epoch.

6 Future Scope

The concept of image segmentation and localization has not been used in this work which can be applied in the future for better results. In addition, we will include various types of lung diseases in this model using X-ray images. We can also increase the performance of our deep learning model with additional layers and parameters. This model will allow doctors to examine infections in the lung from chest X-ray images at an earlier stage.

References

Almezhghwi K, Serte S, Al-Turjman F (2021) Convolutional neural networks for the classification of chest X-rays in the IoT era. Multimed Tools Appl 80:29051–29065. https://doi.org/10.1007/s11042-021-10907-y. Epub ahead of print. PMID: 34155434

Ayan E, Ünver HM (2019) Diagnosis of pneumonia from chest X-ray images using deep learning. In: 2019 scientific meeting on electrical-electronics biomedical engineering and computer science (EBBT)

Bharati S et al (2020) Hybrid deep learning for detecting lung diseases from X-ray images. Inform Med Unlock 20(20):100391

Chowdhury MEH et al (2020) Can AI help in screening viral and COVID-19 pneumonia? IEEE Access 8:132665–132676. https://doi.org/10.1109/ACCESS.2020.3010287

Ciano G, Andreini P, Mazzierli T, Bianchini M, Scarselli F (2021) A multi-stage GAN for multi-organ chest X-ray image generation and segmentation. https://arxiv.org/abs/2106.05132

Gonog L, Zhou Y (2019) A review: generative adversarial networks. In: 2019 14th IEEE conference on industrial electronics and applications (ICIEA), pp 505–510. https://doi.org/10.1109/ICIEA.2019.8833686

Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial networks. https://arxiv.org/abs/1406.2661

Hammoudi K, Benhabiles H, Melkemi M et al (2021) Deep learning on chest X-ray images to detect and evaluate pneumonia cases at the era of COVID-19. J Med Syst 45:75. https://doi.org/10.1007/s10916-021-01745-4

Ibrahim AU, Ozsoz M, Serte S et al (2021) Pneumonia classification using deep learning from chest X-ray images during COVID-19. Cogn Comput. https://doi.org/10.1007/s12559-020-09787-5

Kermany DS et al (2018) Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell 172:1122-1131.e9. https://doi.org/10.1016/j.cell.2018.02.010

Khan W, Zaki N, Ali L (2021) Intelligent pneumonia identification from chest X-rays: a systematic literature review. IEEE Access 9:51747–51771. https://doi.org/10.1109/ACCESS.2021.3069937

Militante SV, Sibbaluca BG (2020) Pneumonia detection using convolutional neural networks. Int J Sci Technol Res 9(04). ISSN: 2277-8616

Mishra M, Parashar V, Shimpi R (2020) Development and evaluation of an AI system for early detection of Covid-19 pneumonia using X-ray (student consortium). In: 2020 IEEE sixth international conference on multimedia big data (BigMM)

Rajasenbagam T, Jeyanthi S, Pandian JA (2021) Detection of pneumonia infection in lungs from chest X-ray images using deep convolutional neural network and content-based image retrieval techniques. J Ambient Intell Hum Comput. https://doi.org/10.1007/s12652-021-03075-2

Raji P, Prashantha HS (2021) Review on pneumonia detection using chest X rays. JCR 8(2):1801–1810. https://doi.org/10.31838/jcr.08.02.182

Rajpurkar P et al (2017) CheXNet: radiologist-level pneumonia detection on chest X-rays with deep learning. arXiv preprint arXiv:1711.05225

Salehinejad H, Colak E, Dowdell T, Barfett J, Valaee S (2019) Synthesizing chest X-ray pathology for training deep convolutional neural networks. IEEE Trans Med Imaging 38(5):1197–1206. https://doi.org/10.1109/TMI.2018.2881415

Shibly KH, Dey SK, Islam MT, Rahman MM (2020) COVID faster R-CNN: a novel framework to diagnose novel coronavirus disease (COVID-19) in X-ray images. Inform Med Unlock 20:100405. ISSN: 2352-9148

Verma G, Prakash S (2020) Pneumonia classification using deep learning in healthcare. Int J Innov Technol Expl Eng (IJITEE) 9(4). ISSN: 2278-3075

Victor Ikechukwu A, Murali S, Deepu R, Shivamurthy RC (2021) ResNet-50 vs VGG-19 vs training from scratch: a comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images. Global Transit Proc 2(2):375–381. https://doi.org/10.1016/j.gltp.2021.08.027

Wang L, Lin ZQ, Wong A (2020) COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. Sci Rep 10:19549. https://doi.org/10.1038/s41598-020-76550-z

Image Captioning Using CNN-LSTM

Akshay Joshi, Kartik Kalal, Dhiraj Bhandare, Vaishnavi Patil, Uday Kulkarni, and S. M. Meena

1 Introduction

Every picture tells a story. Countless details about various objects, their attributes, and relationships to other objects can be found in images. Humans have the capacity to identify key details and provide a concise summary of an image using just one or more captions.

When we ask any human to describe Fig. 1, they will describe it as: "A dog sitting in the grass" or "A white dog playing with a yellow ball". We do this by seeing the image and creating a meaningful sequence of words simultaneously. In the recent years, computer vision (Aker and Gaizauskas 2010) has made significant advance in performing tasks such as image classification and object detection (Liu 2018). Benefitting from such progresses, it is possible to automatically generate one or more captions for an image. This task can be achieved with the help of deep neural networks (Liu 2018).

A. Joshi (✉) · K. Kalal · D. Bhandare · V. Patil · U. Kulkarni · S. M. Meena
KLE Technological University, Hubballi, Karnataka, India
e-mail: akshayjoshi1811@gmail.com

K. Kalal
e-mail: kartikkalalst10@gmail.com

D. Bhandare
e-mail: dbhandare841@gmail.com

V. Patil
e-mail: vaishnavipatil0248@gmail.com

U. Kulkarni
e-mail: uday_kulkarni@kletech.ac.in

S. M. Meena
e-mail: msm@kletech.ac.in

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_32

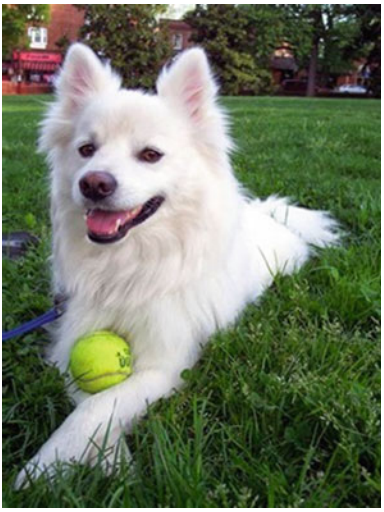


Fig. 1 Sample image shown to a human

Deep learning (Liu 2018) is a machine learning technique that teaches computers to perform tasks that are second nature to people. It is one of the fundamental technologies that enables driverless cars to spot a stop sign, or to differentiate between a pedestrian and a lamppost. A computer model can directly learn to do categorization tasks from images, text, or sound through deep learning. Deep learning models can sometimes beat human performance in terms of accuracy. Models are trained with a vast collection of labelled data and multi-layered neural network architectures.

Image captioning is the process of writing a textual description for a given picture. It has been a key task in the deep learning sector. As it converts images, which are conceived as a sequence of pixels to a sequence of words, image captioning (Aker and Gaizauskas 2010) can be seen as an end-to-end Sequence to Sequence problem. For this purpose, we need to process both the language or statements and the images. In this paper, we are generating a model that will be able to generate a caption for an image. Our work is largely inspired by recent developments in machine translation, where the goal is to translate a sentence S written in a source language into its translation T in a target language by maximizing P(T|S).

To achieve this task, we require both the language component and the image component. For the language component, we will be using Long Short-Term Memory (LSTM) (Staudemeyer and Morris 2019) and for the Image component, we will be using Convolution Neural Network (CNN) (Vedantam et al. 2015) to obtain the feature vectors. In order to increase the likelihood of the desired description sentence of the given training image, the model will be trained.

CNNs' are one kind of Artificial Neural Network (ANN) that are often employed in deep learning to analyse visual input. CNNs are regularized versions of multi-layer perceptrons. Fully linked networks, also known as multi-layer perceptrons, have connections between every neuron in the layer above it and every other neuron in the network. In comparison with other image classification methods, CNNs employ comparatively less pre-processing, that is, the network automatically learns how to improve the filters (or kernels). The main benefit is the lack of dependence on past knowledge or human assistance in feature extraction. LSTM is an advanced Recurrent Neural Network (RNN) that retains information for a longer period of time. On the basis of time-series data, it is employed for processing, predicting, and categorization.

The paper is organized into the following sections: Sect. 2 focuses on the comparison of available image captioning models and background study of CNN and LSTM architectures along with their working. In Sect. 3, we analyse the issues contributing to the decrease in the BLEU score of the baseline VGG-LSTM model and propose a CNN-LSTM model to increase the BLEU score for better generation of caption to images. Section 4 discusses how the proposed model overcomes the issues addressed in the baseline VGG-LSTM model. Section 5 concludes the proposed model.

2 Literature Review

The authors of Herdade and Kappeler (2019) have introduced an approach that incorporated information about the spatial relationship between the identified objects in the given image through the geometric attention which is built by the use of Object Relation Transformer. The algorithm proposed in this work uses feature vectors which are extracted from the region proposals obtained from an object detector. It uses an abstract image feature vector which is sent as input to the encoder in encoder-decoder architecture and the architecture is trained on the ImageNet dataset.

In paper LSTM-VGG-16 (2023), the authors have proposed the approach of CNN-LSTM-based model to solve the challenge of image captioning. The dataset used was Flikr8k. Here, the authors have used Visual Geometry Group (VGG)-16 to extract the features of the images which are than fed along with their captions to LSTM-based model for training. This involves both deep learning and natural language processing technologies. For caption generation the authors have use RNN model that is LSTM which can retain important information over time using its respective gates. The evaluation is done for all the test dataset using standard cost function. Bilingual Evaluation Understudy was used for evaluating the generated captions against different actual captions which yielded a best BLEU score of 0.128 (Fig. 2).

In order to create natural phrases that describe a picture, the authors of paper Vinyals et al. (2015) have suggested a generative model built on a deep recurrent architecture. This model incorporated the recent developments in computer vision and machine translation. They have replaced the encoder Recurrent Neural Network (RNN) by a deep Convolution Neural Network (CNN). This model yielded a BLEU score 28 on the Flickr30k dataset (Fig. 3).

In paper Liu (2018), the authors have described three image captioning techniques using deep neural networks, namely CNN-RNN-based, CNN-CNN–based, and Reinforcement-based framework. In CNN-CNN-based framework, both encoder and decoder are made up of CNN. These networks, unlike the RNN, are feed-forward

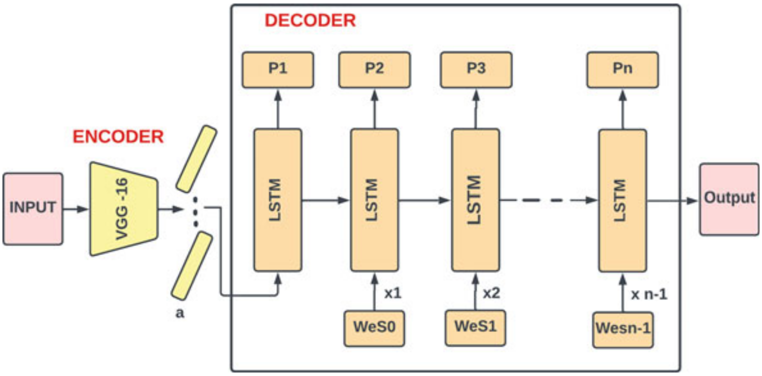


Fig. 2 Neural network architecture used in LSTM-VGG-16 (2023)

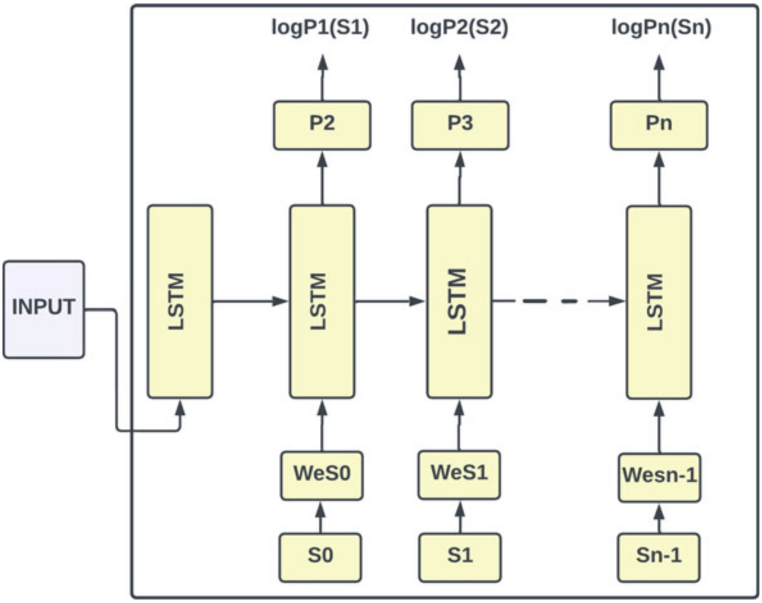


Fig. 3 Neural network architecture used in Vinyals et al. (2015)

without any recurrent function. In CNN-RNN-based framework, CNN enables embedding a fixed-length vector which will help to produce rich representation of input image which can be used for various applications like object detection, segmentation, and recognition. Hence, CNN will be used usually for encoder-decoder-based image captioning techniques. By continuously circling the hidden layer, RNN can be utilized to save historical data, it performs better in case of linguistic knowledge like syntax and semantic information in word sequence and has better training capability. The used Multimodal Recurrent Neural Network, usually know as m-RNN, creatively

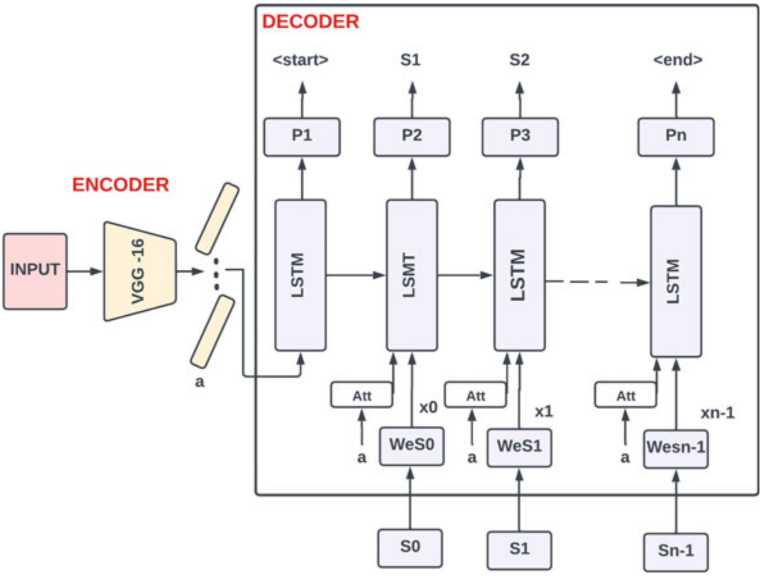


Fig. 4 Neural network architecture used in Liu (2018)

combines CNN and RNN which solves image captioning problem. By applying reinforcement learning into image captioning, the generative model (RNN) will interact with the external environment. Predicting the subsequent word in the sequence can be achieved by the action of sequence generation. This action updates its internal state. When the agent reaches the end of the sequence it receives a reward. The RNN decoder behaves as a stochastic policy which will choose an action corresponding to yielding the following word (Fig. 4).

3 Methodology

In this proposed work, we are implementing a model for image captioning using Convolutional Neural Network (CNN) Vedantam et al. (2015) and Long Short-Term Memory (LSTM) (Staudemeyer and Morris 2019) (Hochreiter and Schmidhuber 1997). This is called the CNN-LSTM model, designed to address the caption generation problem with spatial inputs like images. Given an input image, the model should output a sequence of words or sentence as a caption to the image. The model requires both CNN and LSTM for prediction of the sequences. The CNN extracts the features and nuances of the input image and sends it to the linear layer. Utilizing a weight matrix, linear layers multiply matrices to convert input data into output features. A linear layer receives input information in the form of a one-dimensional tensor that

Table 1 Dataset description

Dataset name	Size		
	Train	Valid	Test
Flickr8k	6000	1000	1000
Flickr30k	28,000	1000	1000

has been flattened, which is then multiplied by the weight matrix. Finally, the LSTM translates the features and objects given by the image into a natural sentence.

A. *Dataset Description*

Flickr30k.token.txt contains the name of each image along with its associated 5 captions. The 5 captions for each image are listed as values in a dictionary that is generated using the image name (without the.jpg suffix) as keys (Table 1).

B. *Flow Chart*

The flow chart of the model is as shown in Fig. 5.

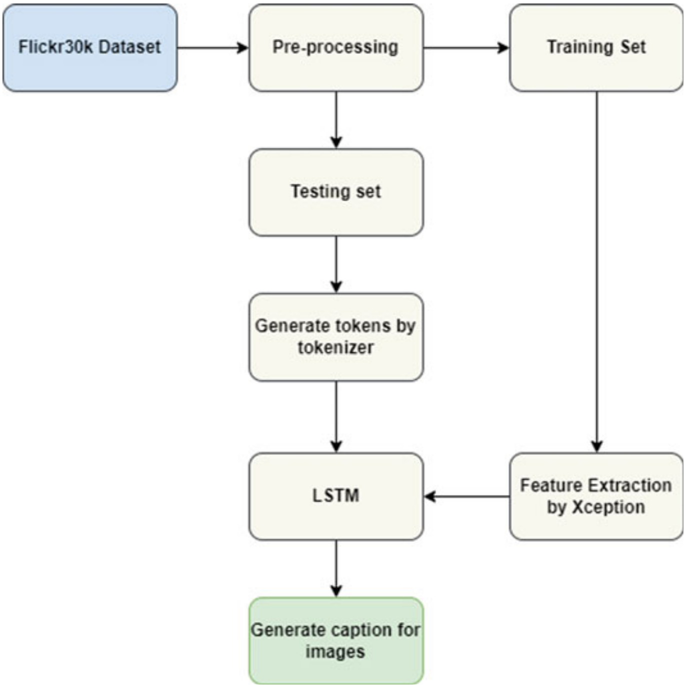

Flickr30k Dataset
Pre-processing
Training Set
Testing set
Generate tokens by tokenizer
LSTM
Feature Extraction by Xception
Generate caption for images

Fig. 5 Flow chart of the proposed work

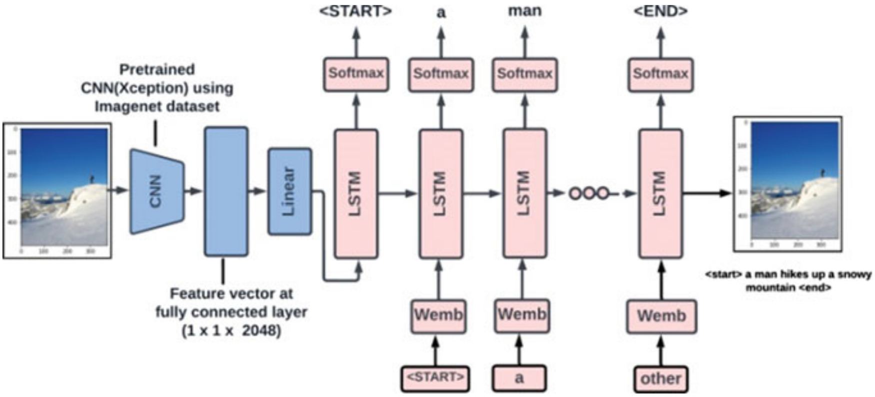


Fig. 6 Architecture of the proposed work

Feature Extractor: This is a 71-layer Xception model trained on the Flickr30k dataset. The dataset has been pre-processed and is sent to the Xception model which will generate features that is further sent to the LSTM.

LSTM: The LSTM will take the extracted features as input and generate the captions of the images by maximizing the likelihood of the sentences. The combination of Xception and LSTM is shown in Fig. 6.

C. *Data Pre-processing*

Since we are dealing with text, we have performed basic data cleaning like converting all the text to lowercase, removing special tokens and eliminating words that contain numbers.

The key of the photos that make up the training set are listed in the text file. These names are therefore added to a train list. In the Python dictionary clean descriptions, these image descriptions are stored. However, each caption will have two additional tokens added when they are loaded; start—this is an initial sequence token that will be put at the beginning of each caption; end—this end sequence token will be added at the conclusion of each caption. The training set is ready to be sent to the CNN for feature extraction.

D. *Convolutional Neural Network*

The significant elements of an image can be extracted using the feature extraction model, a neural network, which frequently produces fixed-length vectors as a result. A deep-CNN, or CNN, is used as the feature extraction sub-model.

Our proposed work involves Xception Network as CNN. This network is trained on the ImageNet dataset.

(1) *Xception Model:* Xception CNN (Chollet 2017) employs depthwise separable convolutions. 36 convolutional layers make up the network's feature extraction base in the Xception architecture. With the exception of the first and last modules, all of the 14 modules made up of the 36 convolutional layers contain

linear residual connections surrounding them. As shown in Fig. 7, it comprises three parts, namely entry flow, middle flow, and exit flow; the data passes through these in order and the middle flow is repeated eight times. After every Convolution and Separable Convolution layer Batch Normalization is applied. The depthwise convolution followed by pointwise convolutions helps in reducing the size to $K \times K \times C$ where as normal convolution would give $K \times K \times N$ (Table 2).

Separable Convolution Layer

- Convolutions known as "depthwise separable convolutions" (Kaiser et al. 2017) are alternatives to "classical" convolutions that are significantly faster to compute.

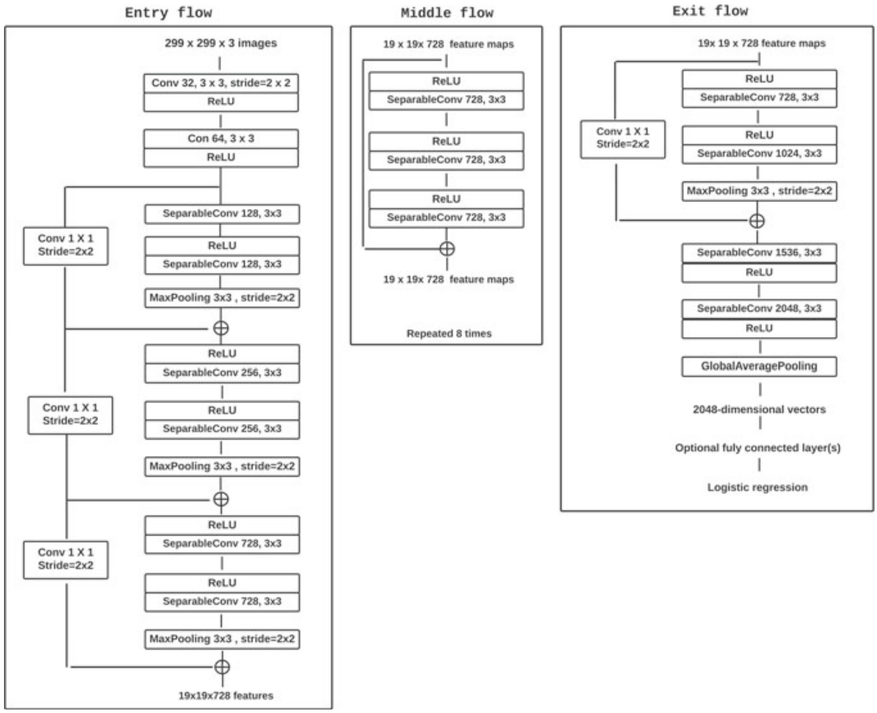


Fig. 7 Xception architecture

Table 2 Comparison of various CNN models

Model	Top-1 accuracy	Top-5 accuracy
VGG-16	0.715	0.9010
Resnet-152	0.770	0.933
Inception-V3	0.7820	0.941
Xception	0.790	0.945

- Number of operations 1 kernel makes is given by, $K^2 \times d^2 \times C$.
- Number of operations for N kernels is given by, $K^2 \times d^2 \times C \times N$.
- To overcome the cost of such operations, depthwise separable convolutions is used which has two parts, namely pointwise convolution and depthwise convolution.
- Depthwise convolution (Kaiser et al. 2017) is the channel-wise $N \times N$ convolution.
- Pointwise convolution (Kaiser et al. 2017) is the 1 × 1 convolution to change the dimension.

As shown in Fig. 8, the pointwise convolution (Kaiser et al. 2017) is followed by the depthwise convolution (Kaiser et al. 2017). The agenda here is that 1 × 1 convolution is done first before any $N \times N$ convolution. This CNN model will extract the features of the image and will save the features in a.pkl file. This file is passed as an input to the LSTM for decoding and prediction of the sequences.

E. *Long Short-Term memory*

In order to solve the issue of vanishing gradients and exploding gradients, we use LSTM. LSTM is an advanced RNN that has three components as observed in Fig. 9. The information from the preceding timestamp is either relevant and should be remembered, or it is irrelevant and can be ignored, is decided by the first component (Forget Gate).

The cell in the second component makes an effort to pick up additional information from the input (Input Gate). In the third component, the cell eventually communicates the revised data from the present timestamp to the following timestamp (Output Gate).

The core of the LSTM is its memory cell c that encodes the knowledge it needs to remember at every time step of the input. By carefully controlling the structures known as gates, the LSTM can either add or remove information from the cell. A

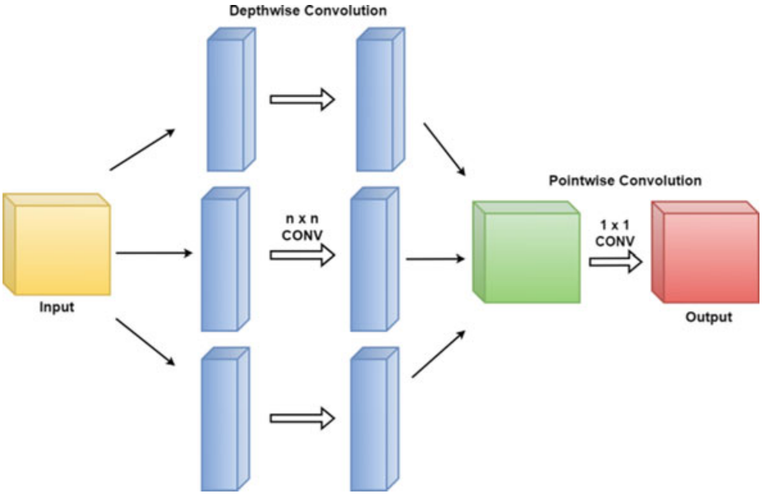


Fig. 8 Separable convolution operation in Xception model

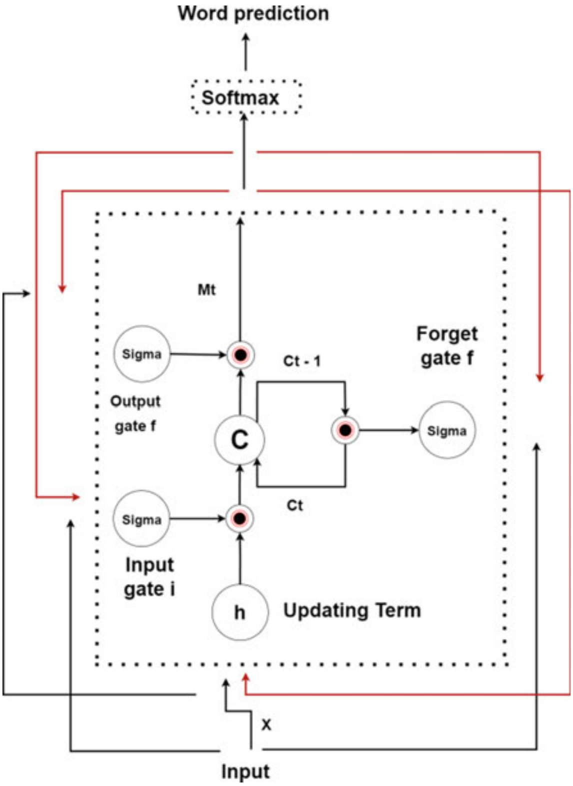


Fig. 9 LSTM used to take features as input and output captions to the image

pointwise multiplication operation and a layer of a sigmoid neural network make up the gates. Since sigmoid layer outputs a value of 0 or 1, a 0 means let nothing through, while a 1 denotes allow everything to pass. The next step is to choose the new data that will be kept in the memory cell. A sigmoid layer that determines which values need to be updated and a tanh layer that generates a vector of new candidate values are introduced in order to accomplish this. The cell state C_{t-1} is updated to the new cell state C_t by multiplying C_{t-1} with the forget gate f_t and adding the input i_t.

The following are the definition of the gates, cell updates, and output.

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1}), \tag{1}$$

$$f_t = \sigma\left(W_{fx}x_t + W_{fm}m_{t-1}\right), \tag{2}$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1}), \tag{3}$$

$$c_t = f_t.c_{t-1} + i_t.\mathrm{h}(W_{cx}x_t + W_{cm}m_{t-1} - 1), \tag{4}$$

$$m_t = o_t \cdot c_t, \tag{5}$$

$$p_{t+1} = SoftMax(m_t). \tag{6}$$

The decoder unit takes as input the encoded features extracted from the image by the CNN and uses them to generate a sequence of words that describe the content of the image. The output of the decoder unit at each time step is then fed into a fully connected layer, which maps the output of the LSTM cells to a probability distribution over the possible words in the vocabulary. This allows the decoder unit to choose the most likely next word in the description, based on the encoded features of the image and the previous words in the generated description.

4 Results

Bilingual Evaluation Understudy (BLEU) is an algorithm that has been used for evaluating the quality of a machine-translated text. BLEU is easy to understand and compute. BLEU is a language independent algorithm that lies between 0 and 1. BLEU tells how good our predicted caption is compared to the provided reference captions. Higher the BLEU score better the quality of the caption. To compute the BLEU score for predicted caption concerns for actual captions, convert the predicted caption and references to unigrams/bigrams. Finally, the modified n-gram precision is calculated from the following formula (Table 3):

$$p_{na} = \sum_{c\varepsilon C} \sum_{n\text{-}gram\varepsilon C} Count_{clip}(n\text{-}gram),$$

$$p_{nb} = \sum_{c'\varepsilon C} \sum_{n-gram'\varepsilon C'} Count\big(n-gram'\big),$$

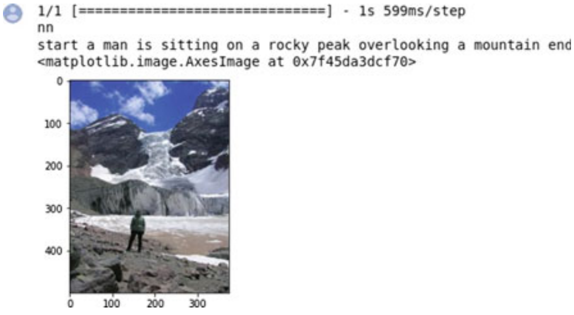
$$p_n = p_{na}/p_{nb}.$$

The proposed model was tested against various input images and the output captions for each image is shown in Figs. 10 and 11.

Table 3 BLEU scores

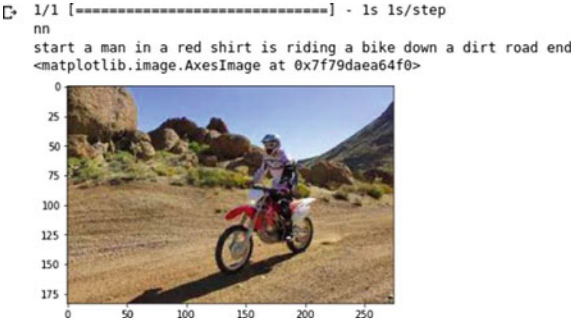
Model	Training Dataset	Test Size	BLEU-1 Score
VGG-LSTM	Flickr8k	1000	0.128326
Ours	Flickr8k	1000	**0.231245**
Ours	Flickr30k	1000	**0.318762**

Bold represent proposed solution and obtained results



```
1/1 [==============================] - 1s 599ms/step
nn
start a man is sitting on a rocky peak overlooking a mountain end
<matplotlib.image.AxesImage at 0x7f45da3dcf70>
```

Fig. 10 Sample input 1 with its respective output



```
1/1 [==============================] - 1s 1s/step
nn
start a man in a red shirt is riding a bike down a dirt road end
<matplotlib.image.AxesImage at 0x7f79daea64f0>
```

Fig. 11 Sample input 2 with its respective output

5 Conclusion

A neural network system that can view images automatically and provide a comprehensible English caption has been introduced in this work as the CNN-LSTM model. The model is built using a Convolutional Neural Network architecture called Xception, which encodes an image into a compact representation (feature extraction), then uses a Recurrent Neural Network called LSTM to create a caption given an image using the extracted features from the CNN. The model is trained to enhance the sentence's likelihood given an image.

References

Aker A, Gaizauskas R (2010) Generating image descriptions using dependency relational patterns. In: ACL

Chollet F (2017) Xception: deep learning with depthwise separable convolutions

Farhadi A, Hejrati M, Sadeghi MA, Young P, Rashtchian C, Hockenmaier J, Forsyth D (2010) Every picture tells a story: generating sentences from images. In: ECCV

Herdade S, Kappeler A, Boakye K, Soares J (2019) Image captioning: transforming objects into words

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8)

Kaiser L, Gomez AN, Chollet F (2014) Depthwise separable convolutions for neural machine translation

Liu S, Bai L, Hu Y, Wang H (2018) Image captioning based on deep neural networks

LSTM-VGG-16 (2023) A Novel and Modular Model for Image Captioning Using Deep Learning Approaches

Staudemeyer RC, Morris ER (2019) Understanding LSTM—a tutorial into long short-term memory recurrent neural networks

Vedantam R, Zitnick CL, Parikh D (2015) CIDEr: consensus-based image description evaluation. arXiv:1411.5726

Vinyals O, Toshev A, Bengio S, Erhan D (2015) Show and tell: a neural image caption generator

Future Sales Prediction Using Regression and Deep Learning Techniques

Uday Kulkarni, Apoorv Bagal, Sunil V. Gurlahosur, Sumedh Kulkarni, Siddhnt Saptasagar, Anant Alias Sudeep Suhas Pingulkar, and S. M. Meena

1 Introduction

Sales are the backbone of every company, and there is a must for the industry to analyze the demeanor of customers regarding product consumption in the market. It supports the industries in launching new products and being profitable (Kaunchi et al. 2021). The objective of sales forecasting is to help businesses develop better strategies according to the provided predictions, improve the cash flow and revenue generation, and inform them of potential threats and losses. Inaccuracy in sales forecasting can lead to massive losses and an increase in insignificant products in inventory or a decrease in significant products in the inventory (Gustriansyah et al. 2022). This can change the situation of a company completely. Decisions that are taken based on the sales forecast are also affected heavily. Accurate sales forecasting can optimize inventories and improve loyalty which ultimately leads to an increase in total profit.

Multiple techniques come under macroeconomics to generate a sales forecast. Macroeconomic changes can affect sales on a significant level. There exists a way to use these indicators to generate a tactical forecast (Verstraete et al. 2020). A framework that can forecast is established which automatically selects relevant variables and predicts. Further, the seasonal component is handled by the seasonal naive

U. Kulkarni (✉) · A. Bagal · S. V. Gurlahosur · S. Kulkarni · S. Saptasagar ·
A. A. S. S. Pingulkar · S. M. Meena
Department of CSE, KLE Technological University, Hubballi, Karnataka, India
e-mail: udaykulkarni@kletech.ac.in

S. V. Gurlahosur
e-mail: svgurlahosur@kletech.ac.in

S. M. Meena
e-mail: msm@kletech.ac.in

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_33

method, and the long-term trend is handled by Least Absolute Shrinkage and Selection Operator (LASSO) regression (Hans 2009; Ranstam and Cook 2018). This is all done while keeping the size of the indicator set as minimal as possible.

The sales forecast can have a major impact on many business decisions. There are many departments that are directly or indirectly affected by the sales forecast and the change in the budget it causes, for example, the marketing department. For processing and predicting long sequential intervals and delays in time series, the LSTM neural network (Hochreiter and Schmidhuber 1997) is adopted by many. Its performance is better in multi-feature regression (Hidalgo and Goodman 2013) by the virtue of its gating mechanism, which determines what data to recollect or erase to avoid overfitting (Hawkins 2004; Pan et al. 2022). Spending a considerable amount of time on determining the sales forecast can save the company from making some very expensive mistakes. Sales prediction is basically an approach to evaluate the product required to be sold to customers surrounded by a specific time period. It can also be used for business planning and helping a company in taking measures or future decisions (Sugiarto et al. 2016).

The most commonly used and conventional prediction models are moving averages and auto-regression (Kaunchi et al. 2021). Auto Regressive Integrated Moving Average with Exogenous Input (ARIMAX) (Peter and Silvia 2012) is a derivative of the traditional statistical model Autoregressive Integrated Moving Average (ARIMA) (Hannan and Kavalieris 1984). In addition to this, gated recurrent unit (GRU) (Dey and Salem 2017) can also be used. Both of these techniques are extremely useful because they can handle multivariable input. Product Life Cycle (PLC) (Levitt et al. 1965) shape-based methods can also be effective (Elalem et al. 2022). There are multiple approaches to predicting future trends in the market. Each of these methods has a certain set of prerequisites that need to be fulfilled for the successful application of the approach. In simple words, the forecasting depends on the previous data available related to their fields (Sharma et al. 2020). One of the recent methods to forecast sales was introduced by Facebook. However, the use of this has received mixed reviews. NeuralProphet (Jha and Pande 2021) is a deep learning technique (Kulkarni et al. 2021, 2020) combined with transfer learning (Kulkarni et al. 2019; Sreekanth et al. 2018) that can also forecast the impact of holidays on sales projections. It is built on an additive regression (AR) model (Stone 1985) that takes into account trends, holidays, and seasonality (Ensafi et al. 2022).

In this paper, our focus is on predicting the demands of the product using Light Gradient Boosting Machine (LGBM) (Ke et al. 2017), Long Short-Term Memory (LSTM), and Multi-Layer Perceptron with LSTM encoded features to help businesses be profitable. We will be comparing the applied models according to the Root Mean Square Error (RMSE) (Maiorov and Crippen 1994) metric to choose the finest of them. The paper follows the following pattern: Sect. 1 gives a concise introduction to the prediction of sales, approaches, methods, and the evaluation metric used. Section 2 outlines the literature survey and the work done in relation to our research topic. Section 3 covers a comprehensive survey of the methodology and model which is being proposed in our work. Section 4 displays the experimental study and resulting output. At long last, our paper comes to an end with future scope and conclusion in Sect. 5.

2 Related Work

In this section, some existing methods for sales forecasting have been presented, which include a CNN-LSTM based model, a stacking-based ensemble learning model, random forest, and an artificial neural network (ANN).

Convolutional neural network (CNN) (Bouvrie 2006) is usually used in image processing (2D), but 1D CNNs also exist and can be used for time-series processing. The structure of the prototype is made in such a way that it makes use of cells to gather data, interpret and evaluate it, and then give it to the next cell. Each cell has distinct gates that control whether the data must be saved, neglected, or transferred on to the forthcoming cell. An accuracy (maximum) degree of 97% was achieved using this methodology (Kaunchi et al. 2021).

Random forest regression (Svetnik et al. 2003) can be used for forecasting as it reduces the overfitting problem. The model uses the random forest algorithm for predicting sales of existing products as it reduces overfitting and is more accurate. Neural networks can work with incomplete knowledge; therefore, for predicting sales of new products, the artificial neural network (ANN) (Jain et al. 1996) model is applied to the data taken from the feedback of customers. The model works great with multidimensional data and dealing with outliers since random forest is used (Sharma et al. 2020).

The ensemble learning technique (Dietterich et al. 2002) known as “stacking” involves training multiple machine learning algorithms and combining the predictions of all the algorithms to improve the performance of the overall model. A two-level statistical model was designed where stacking was performed on two layers. Linear regression, support vector regression, and cubist acted as the bottom layers. Predictions from the bottom layer were used as inputs to the top layer to make the final prediction. A multi-level model approach performs better than a single model approach, as the former models provided more information to the subsequent which leads to better overall predictions (Li et al. 2022; Punam et al. 2018).

Every now and then, the Holt-Winters Additive Method and Holt-Winters Multiplicative Method (Chatfield 1978) are used in Enterprise Resource Planning (ERP). Implementing this method in Enterprise Resource Planning can further make the model versatile, error-free, and coherent in calculating the number of products demanded by customers. Three basic components are required for implementing this approach through an equation: Sales trend (the change in sales from month to month or season to season), level of sales (sales data without any influence from other variables), the periodic index for the month to carry out the predictions (Sugiarto et al. 2016).

3 The Proposed Model

The data flow diagram shown in Fig. 1 describes the sequence that the product sales dataset of 1C Company goes through in building the proposed model to predict future sales and produce accurate results. It consists of six steps, each playing an important part in the building of the proposed model. The first three steps are a pre-processing phase before building the LSTM network. After the model is built, it is trained and tested with the never-before-seen data.

LSTM stands for Long Short-Term Memory network, which is a class of Recurrent Neural Networks (RNN). This type of network is proven to be advantageous as it learns and remembers long sequences and does not depend on a predetermined window-lagging observation as input. An LSTM cell decides whether to preserve the past knowledge that the network has been administered to or to forget peripheral data.

3.1 LightGBM Model

Gradient Boosting Decision Tree (GBDT) is a widely used prediction algorithm and has popular predictive implementations such as XGBoost. These implementations work well with small-size data with lower dimensions but are not efficient with higher dimensional and large-size data because for each feature they need to scan every data instance hence reducing their efficiency and scalability. These algorithms scan all the information gained from all the split points to estimate the best one. Light Gradient Boosting Machine is faster than Simple Gradient Boosting Machine and XGBoost. It is achieved by using three algorithms Histogram Splitting, Gradient-Based One Side Sampling (GOSS), and Exclusive Feature Bundling(EFB). These

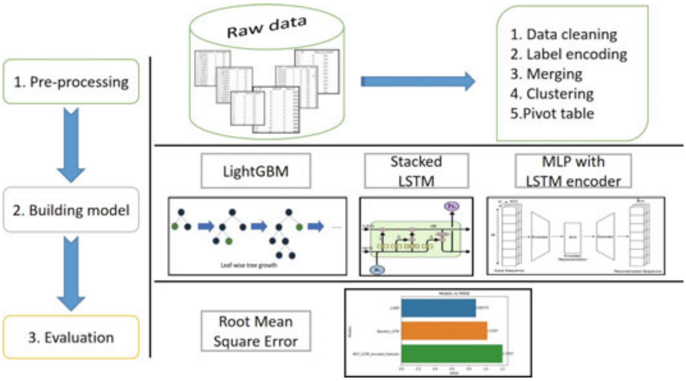


Fig. 1 Data flow diagram of proposed model

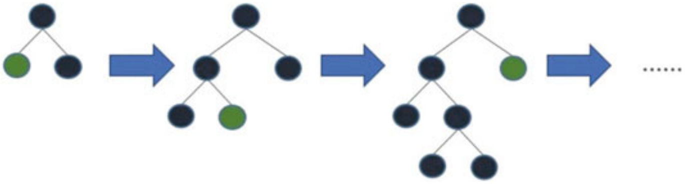


Fig. 2 Leaf-wise growth (Khandelwal 2017)

three algorithms help the model reduce the histogram complexity by downsampling the data and features, respectively.

Figure 2 describes how the above-stated three methods allow the model to grow the trees leaf-wise rather than level-wise (like in XGBoost). Since leaf-wise model grows the nodes on the basis of their global contribution rather than the local loss, it helps the tree learn faster than the level-wise growth. Unlike XGBoost and GBDT algorithm, LGBM scans and analyzes which leave reduces the loss to the maximum and splits that leaf instead of scanning all the leaves.

The Light Gradient Boosting Method is an open-source gradient boosting framework that utilizes decision tree-based methods to carry out ranking and classification undertakings. In simple words, it works on smart splitting optimization, smart feature engineering, and smart sampling of the input records using the following techniques to build a better and more efficient model than other boosting algorithms.

1. Histogram/Bin-wave splitting:

Gradient boosting generally iteratively builds weak learners with every new weak learner assigning more weights to the wrongly predicted features. Finally, it gives output as a form of a sum of all the weak learners F built iteratively as F_m. For each tree, the algorithm has to calculate (f, f) according to information gain where f is a feature and f is a threshold value for the respective feature. Histogram is used to make this algorithm more efficient.

Figure 3 depicts how the input data is transformed into histogram bins, which decreases the nodes of the tree structure, increasing efficiency and decreasing the time cost.

2. **EFB (Exclusive Feature Bundling)**:

This algorithm is used to reduce the number of features fed to the model using the Greedy algorithm.

It does so by bundling mutually exclusive features (features that never take nonzero values simultaneously), thus decreasing the dimensions through smart feature engineering.

Algorithm 1 Greedy bundling (Ke et al. 2017)

```
Input: $F$: Feature, $K$: max conflict count
Construct graph G
searchOrder ← $G$.sortbyDegree ()
bundles ← { }, bundlesConflict ← { }
for $i$ in $searchOrder$ do
    needNew ← True
    for $j = 1$ to $len(bundles)$ do
        cnt ← ConflictCnt(bundles[j],$F$ [i])
        if $cnt + bundlesConflict[i] \geq K$ then
            bundles[j].add($F$ [i]), needNew ← False
            break
    if $needNew$ then
        Add $F$ [$i$] as new bundle to $bundles$
Output: $bundles$
```

Algorithm 1 given below constructs a graph G with features as its vertices and the mutually exclusive features are connected with an edge between them. Then it sorts the graph G in descending order in order of their degree. In the next step, it creates two sets *bundles* and *bundlesConflict* each. For each feature F [i] in graph G, the algorithm determines if the features belong to the existing bundle, conflict bundle, or need a new bundle and append the feature to the appropriate bundle.

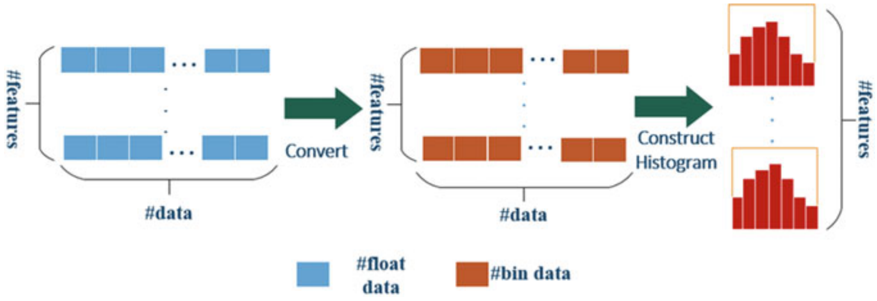


Fig. 3 Bin-wave splitting (Yin et al. 2021)

Algorithm 2 Merge exclusive features (Ke et al. 2017)

```
Input: numData: number of data
Input: F: one bundle of exclusive features
binRanges ← {0}, totalBin ← {0}
for f in F do
    totalBin + = f.numBin
    binRanges.append(totalBin)
newBin ← newBin(numData)
for i = 1 to numData do
    newBin[i] ← 0
    for j = 1 to len(F) do
        if F [j].bin[i] / = 0 then
            newBin[i] ← F [j].bin[i] + binRanges[j]
Output: bundles
```

Algorithm 2 merges the exclusive features. At first, it creates two sets with null values *binRanges* and *totalBin.* As the histogram algorithm creates discrete bins and the exclusive feature resides in different bins. To identify all the features, we add an offset to the original values of the features in the bundle before adding it to the bins. After adding the offset, the feature bundle is added to new bins. The values of *binRanges* and *totalBin* are updated simultaneously to the above procedure of adding offsets.

3. **GOSS (gradient-based one-sided sampling)**:

GOSS is a downsampling technique based on gradients. It elects samples with a large gradient and random samples with a smaller gradient.

$$V_j(d) = \frac{1}{n}(A + B) \tag{1}$$

In Eq. (1) for feature j, the algorithm calculates the maximum gain $V_j(d^*{}_j)$. At first, it sorts all the gradients in descending order.

$$A = \frac{\left(\sum_{x_i \in A_l} g_i + \frac{1-a}{b} \sum_{x_i \in B_l} g_i\right)^2}{n_l^j(d)} \tag{2}$$

$$B = \frac{\left(\sum_{x_i \in A_r} g_i + \frac{1-a}{b} \sum_{x_i \in B_r} g_i\right)^2}{n_r^j(d)} \tag{3}$$

where

$\mathrm{A}_l = \{x_i \in A : x_{ij} \leq d\}$, $A_r = \{x_i \in A : x_{ij} > d\}$,
$\mathrm{B}_l = \{x_i \in B : x_{ij} \leq d\}$, $B_r = \{x_i \in B : x_{ij} > d\}$

As in Eq. (2), it selects the top a*100% gradients (i.e., large undertrained gradients) and gets instance subset A. Then, for the remaining set A^c consisting $(1 - a)100\%$

instances with smaller gradients in Eq. (3). Further, we randomly sample a subset B. Minimizes the well-trained gradients by a factor of b. Therefore, in GOSS, we calculate estimated $\tilde{V}_j(d)$ over a subset instead of exact $\tilde{V}_j(d)$ over the entire dataset to determine the split point of the nodes, which reduces the computation cost. The coefficient $\frac{(1-a)}{b}$ in Eqs. (2) and (3) is used to normalize the sum of the gradients over B back to the size of A^c. It uses the instances that are not used well while training the model. Thus, the model overall increases the model's training speed.

3.2 Stacked LSTM

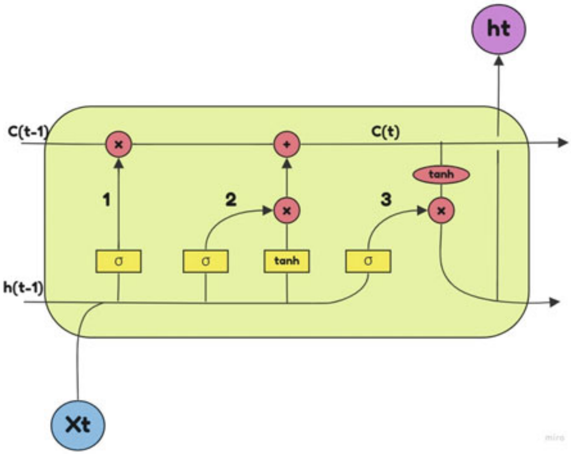
A specific kind of RNN called LSTM has a collection of cells with characteristics that allow it to memorize the data which is usually lost in a standard RNN model. It enables the categorization of sequence data according to certain time steps. Each LSTM cell has a forget gate, a cell state, and an output gate.

Figure 4 illustrates LSTM cell architecture. It can be represented by these equations:

$$i_t = \sigma\left(w_i\left[h_{t-1}, x_t\right] + b_i\right) \quad (4)$$

i_t represents input gate.
t represents timestep.
h_{t-1} represents previous hidden state.
w_i *represents weight matrix of sigmoid operator.*
x_t *input.*
b_i *connection bias at time 't'.*

Fig. 4 Data flow of stacked LSTM model (Soni 2018)



Equation (4) represents the input gate. The crucial decision of determining what part of the new data will be retained in the cell state.

A. Input Gate:

The input gate decides what information should be updated in the cell state. Two gates make up the input gate: one is a sigmoid layer gate, and the other is a tan *h* layer gate. The value that has to be updated will be determined by the sigmoid layer, and the value that needs to be added to the following step will be determined by the tan *h* layer.

$$f_t = \sigma\left(w_f\left[h_{t-1}, x_t\right] + b_f\right) \tag{5}$$

f_t *represents forget gate.*
w_f *weight matrix between forget gate and input gate*

Equation (5) depicts the working of forget gate. It specifies the information that shall be thrown away. Sigmoid function is added over it to make f_t between 0 and 1.

B. Forget Gate:

Which information needs to be neglected and which data must be reserved is decided by the forget gate. The hidden state h_{t-1} and input X_t data is allowed to pass through a function called as sigmoid, which generates a value between 0 and 1. If the output value is near to 1, then the old output is essential.

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \tag{6}$$

o_t *represents output gate.*
w_o *weight matrix of output gate.*

Equation (6) represents the output gate. This is done so as to provide activation to the final output of the LSTM block at a time "*t*."

C. Output Gate:

The hidden state of the next cell is determined by the output gate. First, the values of the current state and previous hidden state are passed through a function called sigmoid. The cell state is passed through a tan *h* function. To determine the output, we multiply the tan *h* output with the sigmoid output. This output is used as a hidden state for the next cell.

$$c_{t\prime} = \tan h(w_c[h_{t-1}, x_t] + b_c) \tag{7}$$

$$c_t = f_t * c_{t-1} + i_t * c_t \tag{8}$$

$$h_t = o_t * \tan h(c^t) \tag{9}$$

c_t *cell state* (*memory*) *at timestamp* (*t*).
$c_{t'}$ *candidate for cell state at timestamp* (*t*).

Note: Others are the same as stated before.

D. Cell State:

Now with the information provided, the cell state is calculated and, the cell state is multiplied by the forget gate's output. Then the cell state is administered with this output of the input gate, giving rise to a new cell state.

3.3 MLP with LSTM Encoded Features

The purpose of this work is to show one-way time-series data can be efficiently encoded into lower dimensions to be used in non-time-series models. LSTM networks can be constructed with a structure called encoder–decoder LSTM, which allows models to support flexible-length input sequences and predict flexible-length output sequences.

It can be represented by the following equations:

$$z = e(p) \in Rn \tag{10}$$

Equation (10) states that p is the input in Rm and compression of *p* takes place to get an encoded version.

$$l = 1/2 \sum_{p} ||p - \hat{p}||^2 \tag{11}$$

Equation (11) carry outs the reconstruction to give the output as $\widehat{p} = d(z)$ in Rm. The autoencoder is trained by making the error in reconstruction as merest as possible.

Equations (7)–(9): Represents the different cell states of LSTM.

Figure 5 represents the structure of the autoencoder. It is a type of neural network which tries to generate the output by copying the input.

4 Experiment and Results

4.1 Dataset

1C software company daily sales data has been used as the time-series dataset for our proposed work. We have gathered the dataset from the Kaggle competition—Predict future sales. The below table displays the CSV files contained in the dataset.

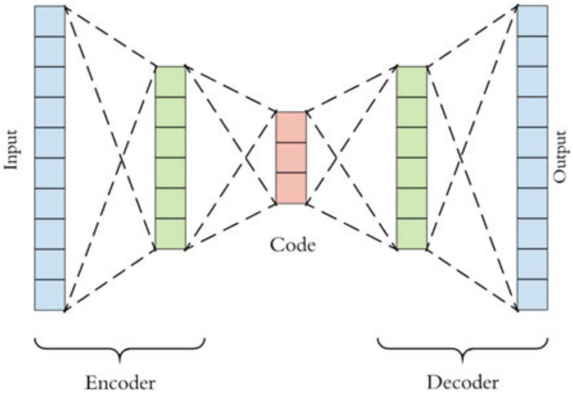


Fig. 5 Encoder–decoder architecture (Nguyen et al. 2021)

4.1.1 Description

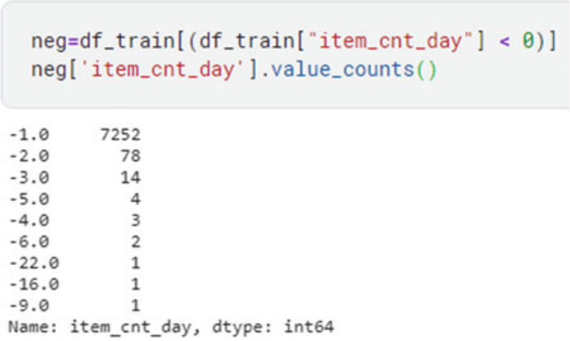
1. items.csv—Auxiliary data about the products.
2. item_categories.csv—Auxiliary data about the item categories.
3. shops.csv—Auxiliary data about the shops.
4. sales_train.csv—It represents the training block containing time-series data from Jan 2013 to Oct 2015.
5. test.csv—It represents the testing block. It is the data of shops for which we need to predict the sales for Nov 2015.
6. sample_submission.csv—It is a sample of the submission file in a specific format.

Data fields:

Following are the data fields present in the dataset files:

1. ID—id of a tuple within a dataset.
2. item_id—distinct id for each product.
3. shop_id—distinct id for each shop.
4. item_category_id—distinct id for item category.
5. item_name—the label of the item.
6. item_price—the present price of a particular item.
7. item_cnt_day—count of products sold monthly.
8. item_category_name—the label of a particular item category.
9. shop_name—the label of a particular shop.
10. date—date of DD/MM/YYYY form.
11. date_block_num—a successive number for months, e.g., September 2016 is 1, October 2016 is 2.

Fig. 6 Data cleaning for LightGBM model



```
neg=df_train[(df_train["item_cnt_day"] < 0)]
neg['item_cnt_day'].value_counts()
```

```
-1.0     7252
-2.0       78
-3.0       14
-5.0        4
-4.0        3
-6.0        2
-22.0       1
-16.0       1
-9.0        1
Name: item_cnt_day, dtype: int64
```

4.2 Pre-processing

4.2.1 LightGBM Pre-processing

The data pre-processing part consists of transforming the online shop names in Russian to “Internet” in English in shops.csv and merging city. pkl to add the city information to each shop. Left merging and label encoding the item category, creating Item class with item_id, item_name, and label encoded item_category_id for respective item_category.

Figure 6 represents the data cleaning process to remove all the negative item count values from the sales train dataset.

We also eliminate the shops with duplicate shop id from the shops.csv dataset. Build the indexed dataframe for sales train. We aggregate all the training datasets to form df work dataset to feed to the model to train. Adding item means, shop means and lag features to the df work.

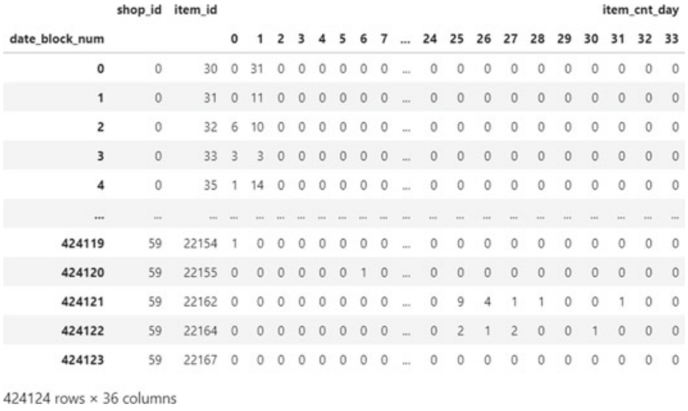
4.2.2 Stacked LSTM

In the case of stacked LSTM model pre-processing:

Figure 7 represents the created pivot table, indexed by shop_id and item_id created a column item_cnt_day which indicates the count of items sold on a day. We replaced the null values in the table with zero. Then we built the indexed dataframe for sales_train.

4.2.3 MLP with LSTM encoded features

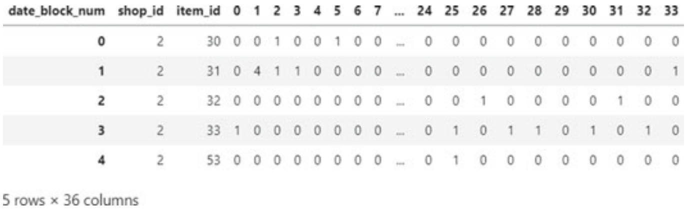
We are dropping all features but “item_cnt_day,” hence turning the problem into a univariate time-series problem.



	shop_id	item_id																			item_cnt_day
date_block_num			0	1	2	3	4	5	6	7	...	24	25	26	27	28	29	30	31	32	33
0	0	30	0	31	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	31	0	11	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	32	6	10	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	33	3	3	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	35	1	14	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
424119	59	22154	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
424120	59	22155	0	0	0	0	0	0	1	0	...	0	0	0	0	0	0	0	0	0	0
424121	59	22162	0	0	0	0	0	0	0	0	...	0	9	4	1	1	0	0	1	0	0
424122	59	22164	0	0	0	0	0	0	0	0	...	0	2	1	2	0	0	1	0	0	0
424123	59	22167	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

424124 rows × 36 columns

Fig. 7 Pivot table for stacked LSTM model



date_block_num	shop_id	item_id	0	1	2	3	4	5	6	7	...	24	25	26	27	28	29	30	31	32	33
0	2	30	0	0	1	0	0	1	0	0	...	0	0	0	0	0	0	0	0	0	0
1	2	31	0	4	1	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1
2	2	32	0	0	0	0	0	0	0	0	...	0	0	1	0	0	0	0	1	0	0
3	2	33	1	0	0	0	0	0	0	0	...	0	1	0	1	1	0	1	0	1	0
4	2	53	0	0	0	0	0	0	0	0	...	0	1	0	0	0	0	0	0	0	0

5 rows × 36 columns

Fig. 8 Pivot table for stacked MLP with LSTM encoder model

Figure 8 depicts the aggregation of all the data month-wise. Also "item_cnt" is greater than or equal to 0 and less than or equal to 20. Then the data is converted to a pivot table.

4.2.4 Exploratory Data Analysis

Given below are the techniques used for exploratory data analysis

This graph from Fig. 9 shows the sales of the company by month from January 2013 to November 2015. It can be observed that there is a decreasing trend and seasonality in sales.

The diagram shown in Fig. 10 represents the correlation matrix between the data fields of the sales train.

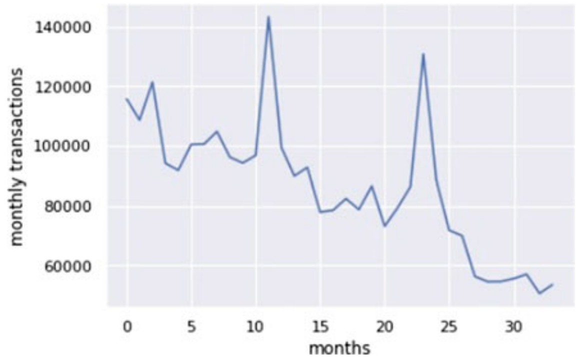


Fig. 9 Monthly sales of the company

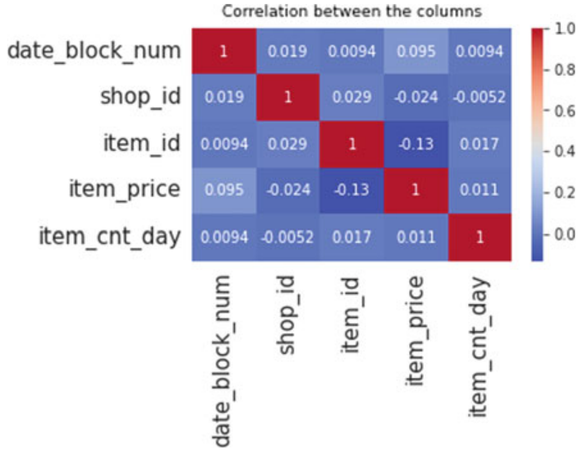


Fig. 10 Correlation matrix between columns of sales train

4.3 System Specification

The code of the proposed models is trained on the Kaggle notebook which is equipped with 4-core CPUs and 30 Gigabytes of RAM.

4.4 Results

The following graph is generated after training and testing the model.

Figure 11 depicts that the LGBM model performed better than the other models. This can be attributed to the generation of lag features which help in autocorrelation. The stacked LSTM model performed better than the model with MLP with LSTM encoded features.

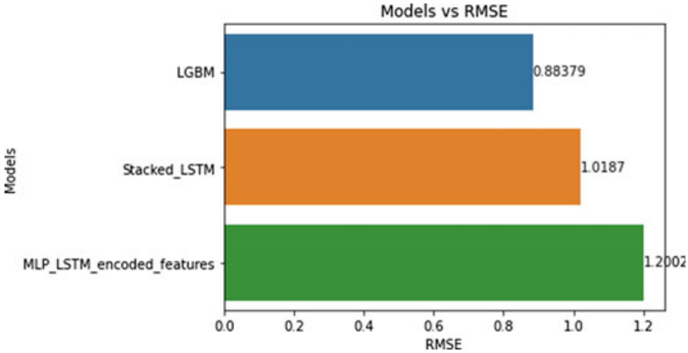


Fig. 11 RMSE metric comparison between the models

5 Conclusion and Future Scope

In this work, we presented some machine learning models for future sales prediction. Based on root mean square deviation (RMSD), the Light Gradient Boosting Machine (LGBM) model outperformed both stacked Long Short-Term Memory (LSTM) and Multi-Layer Perceptrons (MLP) with LSTM encoded features. This result can be attributed to the use of lag features in the LGBM model. Item_cnt_month-1 was the most important feature. There is a 13.24% decrease in RMSE of LGBM compared to stacked LSTM and 26.3% less than the model with LSTM encoded features.

There exists a firm yearning in every other business to fathom the demands of the customer and, consequently, refrain from any potential failures, losses, or shortages of the products. Thus, persistent research is shooting up in this sphere to make errorless and well-aimed predictions of sales. In our future works and research, we hope to use transfer learning, prophet, and N-beats to forecast future sales.

References

Bouvrie J (2006) Notes on convolutional neural networks

Chatfield C (1978) The holt-winters forecasting procedure. J Roy Stat Soc Ser C 27(3):264–279

Dey R, Salem FM (2017) Gate-variants of gated recurrent unit (gru) neural networks. In: 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS). IEEE, pp 1597–1600

Dietterich TG et al (2002) Ensemble learning. The handbook of brain theory and neural networks 2(1):110–125

Elalem YK, Maier S, Seifert RW (2022) A machine learning-based framework for forecasting sales of new products with short life cycles using deep neural networks. Int J Forecast 19:1874–1894

Ensafi Y, Amin SH, Zhang G, Shah B (2022) Time-series forecasting of seasonal items sales using machine learning—a comparative analysis. Int J Inf Manage Data Insights 2(1):100058

Gustriansyah R, Ermatita E, Rini DP (2022) An approach for sales forecasting. Expert Syst Appl 207:118043

Hannan EJ, Kavalieris L (1984) A method for autoregressive-moving average estimation. Biometrika 71(2):273–280

Hans C (2009) Bayesian lasso regression. Biometrika 96(4):835–845

Hawkins DM (2004) The problem of overfitting. J Chem Inf Comput Sci 44(1):1–12

Hidalgo B, Goodman M (2013) Multivariate or multivariable regression? Am J Public Health 103(1):39–40

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780

Jain AK, Mao J, Moidin Mohiuddin K (1996) Artificial neural networks: a tutorial. Computer 29(3):31–44

Jha BK, Pande S (2021) Time series forecasting model for supermarket sales using fb-prophet. In: 2021 5th international conference on computing methodologies and communication (IC-CMC). IEEE, pp 547–554

Kaunchi P, Jadhav T, Dandawate Y, Marathe P (2021) Future sales prediction for Indian products using convolutional neural network-long short term memory. In: 2021 2nd global conference for advancement in technology (GCAT). IEEE, pp 1–5

Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu T-Y (2017) LightGBM: a highly efficient gradient boosting decision tree. Adv Neural Inf Process Syst 30

Khandelwal EB (2017) Which algorithm takes the crown: light GBM vs XGBoost?

Kulkarni U, Meena SM, Gurlahosur SV, Mudengudi U (2019) Classification of cultural heritage sites using transfer learning. In: 2019 IEEE fifth international conference on multimedia big data (BigMM). IEEE, pp 391–397

Kulkarni U, Meena SM, Joshua P, Rodrigues K, Gurlahosur SV (2020) Integrated crowdsourcing framework using deep learning for digitalization of Indian heritage infrastructure. In: 2020 IEEE sixth international conference on multimedia big data (BigMM). IEEE, pp 200–208

Kulkarni U, Meena SM, Gurlahosur SV, Bhogar G (2021) Quantization friendly mobilenet (QF-MobileNet) architecture for vision based applications on embedded platforms. Neural Netw 136:28–39

Levitt T et al (1965)Exploit the product life cycle, vol 43. Graduate School of Business Administration, Harvard University

Li D, Lin K, Li X, Liao J, Du R, Chen D, Madden A (2022)Improved sales time series predictions using deep neural networks with spatiotemporal dynamic pattern acquisition mechanism. Inf Process Manage 59(4):102987

Maiorov VN, Crippen GM (1994) Significance of rootmean-square deviation in comparing three-dimensional structures of globular proteins. J Mol Biol 235(2):625–634

Nguyen HD, Tran KP, Thomassey S, Hamad M (2021) Forecasting and anomaly detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management. Int J Inf Manage 57:102282

Pan S-Y, Liao Q, Liang Y-T (2022) Multivariable sales prediction for filling stations via GA improved BiLSTM. Petrol Sci 19(5):2483–2496

Peter D, Silvia P (2012) Arima vs. arimax–which approach is better to analyze and forecast macroeconomic time series. In: Proc 30th international conference mathematical methods in economics, vol 2, pp 136–140

Punam K, Pamula R, Jain PK (2018) A two-level statistical model for big mart sales prediction. In: 2018 international conference on computing, power and communication technologies (GUCON). IEEE, pp 617–620

Ranstam J, Cook JA (2018) Lasso regression. J Br Surg 105(10):1348–1348

Sharma AK, Goel N, Rajput J, Bilal M (2020) An intelligent model for predicting the sales of a product. In: 2020 10th international conference on cloud computing, data science & engineering (confluence). IEEE, pp. 341–345

Soni M (2018) Understanding architecture of LSTM cell from scratch with code

Sreekanth P, Kulkarni U, Shetty S, Meena SM (2018) Head pose estimation using transfer learning. In: 2018 international conference on recent trends in advance computing (ICRTAC). IEEE, pp 73–79

Stone CJ (1985) Additive regression and other nonparametric models. Ann Stat 13(2):689–705

Sugiarto VC, Sarno R, Sunaryono D (2016) Sales forecasting using holt-winters in enterprise resource planning at sales and distribution module. In: 2016 international conference on information & communication technology and systems (ICTS). IEEE, pp 8–13

Svetnik V, Liaw A, Tong C, Christopher Culberson J, Sheridan RP, Feuston BP (2003) Random forest: a classification and regression tool for compound classification and QSAR modeling. J Chem Inf Comput Sci 43(6):1947–1958

Verstraete G, Aghezzaf E-H, Desmet B (2020) A leading macroeconomic indicators' based framework to automatically generate tactical sales forecasts. Comput Ind Eng 139:106169

Yin L, Ma P, Deng Z (2021) Jlgbmloc: a novel high-precision indoor localization method based on lightGBM

Rider Face Mask Detection and Alerting Using Machine Learning Techniques

V. Laxmi Deepak, C. Vineeth, N. Naga Jayanth, K. Madhan, Vani Vasudevan, and B. V. Shruti

1 Introduction

In recent years, face detection (Sharma et al. 2021) has become an increasingly popular issue in computer vision (CV). It has been included in cameras that ensure faces are in focus prior to snapping a photo, and when uploaded to social networking sites, persons will be automatically tagged. Face recognition has been widely implemented for security purposes. There are now ATMs with facial detection and recognition software. AI and computers can replicate thinking using logic and mathematics to do tasks such as problem solving and learning. Machine learning enables computers to learn without being explicitly programmed. This causes the computer to learn and improve based on experience, and it will continue to improve until the model's accuracy is sufficient to do the task. Computer vision (CV) is utilised for tasks including object detection and pattern recognition. Object recognition is capable of detecting objects and classifying images.

V. Laxmi Deepak (✉) · C. Vineeth · N. Naga Jayanth · K. Madhan · V. Vasudevan · B. V. Shruti
Computer Science Engineering, Nitte Meenakshi Institute of Technology (Visvesvaraya Technological University), Bengaluru, India
e-mail: deepakong28@gmail.com

V. Vasudevan
e-mail: vani.v@nmit.ac.in

B. V. Shruti
e-mail: shruthi.bv@nmit.ac.in

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_34

2 Related Work

Object detection is one of the world's leading technologies. It is an OpenCV application. We will be able to detect many items in a single image using object detection. It is simple for people to distinguish items in photographs, but computers struggle with this task. Object detection uses include unlocking a cell phone using facial recognition, autonomous vehicles, etc. Similarly, facial masks (Khan et al. 2019) and vehicle number plate detection come into play.

1960 saw the introduction of the face recognition technique (Wang et al. 2021) by Woody Bledsoe, Helen Chan, and Charles Bisson. They programmed computers to recognise human faces. Similarly, Face mask detection was developed by Natraj Mishra, a professor in the department of mechanical engineering and a member of the United States Public Service Commission. Using machine learning techniques, specifically the supervised learning methodology and a PC web camera, he devised a method for identifying and detecting whether or not a person is wearing a mask. This has been utilised in numerous settings, including universities, schools, airports, hospitals, and offices. In situations where the COVID-19 virus may increase, the likelihood of its transmission is great.

Numerous old experiments are performed using older CNN (Siradjuddin et al. 2021) with poorer precision and performance. This system processes images and separates frames using sophisticated machine learning models. For YOLO, like other region proposal classification networks (fast RCNN) that perform detection on various region proposals networks (Vasudevan and Gounder 2021) and end up performing prediction multiple times for various regions in an image, YOLO architecture is more like fully convolutional neural network (FCNN) and passes the image through the FCNN only once; the output of the FCNN is prediction. Real-time Graphics User Interface-based Automated Facial Recognition was discussed in Achyutha et al. (2022). In addition, the mask detection system that was built using Principal Component Analysis (PCA) and the HAAR cascade algorithm is also included.

The identification of employees' helmets is vital for their safety. Using this as their motto, (Benyang et al. 2020) was created. They built a system to detect helmets using YOLO because it more successfully meets the real-time needs. YOLOv4 is used for image processing, as it has a high processing speed and accuracy.

Through the years, numerous techniques (Benyang et al. 2020; Loey et al. 2021; Qin and Li 2020; Zhang et al. 2020; Razavi et al. 2021; Dey et al. 2021; Kabir and Tanimoto 2020; Chen et al. 2018; ErdincKocer and KursatCevik 2011) have been utilised to detect licence plates. Automatic number plate recognition (ANPR) was accomplished in Mohammed Shariff et al. (2021) by combining Optical Character Recognition (OCR) and image processing. OpenCV library and Python language were utilised for image processing techniques utilising pytesseract for the ANPR system. TESSERACT is an Optical Character Recognition system (OCR).

3 Proposed Methodology

The purpose of this section is to provide comprehensive information regarding the proposed system. Our system was built utilising the Common Objects in Context (COCO) dataset, which utilises a JSON format to convey information about each dataset and the photos contained inside it. Face mask and helmet classifiers were determined using YOLO v4 weights and the Caffe model (pre-trained model). Using an application programming interface (API) licence plate recognizer to extract the characters from the licence plate and send an alert to the rider's registered mail. This is what distinguishes us from other systems.

4 Design

Flow Diagram

The high-level design of the flow chart represents the working of the system from start to end with all conditions and possibilities.

The system's overall workflow is depicted in Fig. 1. After the system has run successfully, an interface (as seen in Fig. 2) with two options appears. The select option allows the user to upload an image from the device as input to predict whether the rider is wearing a helmet or not, then follows wearing a mask or not and extracts the number plate details to send an alert mail or SMS to the registered details. The capture option activates the system's main camera and takes a live photo as input to predict. The quit button ends the programme.

5 Implementation

1. Data Collection (Images)

The collecting of example images was the most challenging aspect of this project. The sample photographs of two-wheeler riders wearing helmets, without wearing helmets, and wearing masks with the number plate clearly visible. After collecting images, they must be classified into two categories. The positive consists of the real-time photos to be detected by the system, while the negative consists of the real-time images to be ignored by the system. The machine may not be able to detect characters due to the weather.

We must consider the circumstances in which it will be utilised. Several factors impact the accuracy of the object detection model, particularly in relation to the image. This can include the light intensity, the angle at which the photograph was recorded, and the background. In some older vehicles, the number on the licence plate may be obscured or absent. Black helmets with black visors and backgrounds

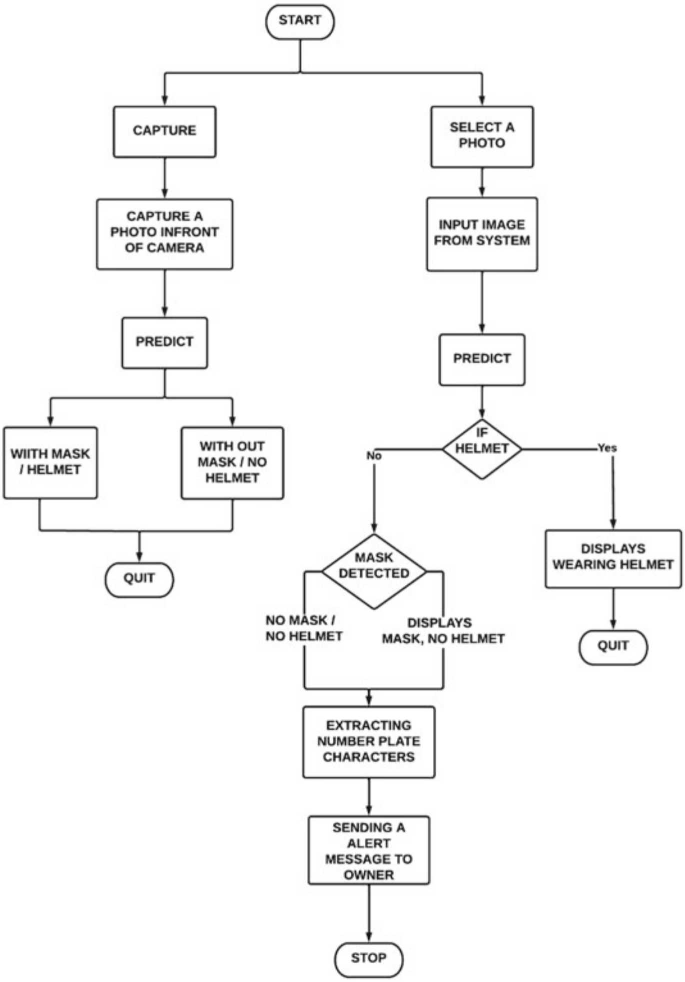


Fig. 1 Flow chart of the system working

can occasionally mislead the system. Therefore, the first step is to identify the issue and consider its use cases.

Taking into account all scenarios, we gathered over 50 real-time test cases including 25 riders in the idle position, at rest while in motion, and with blurred pictures. Once this is complete, we can advance the training model.

2. Machine Training

Transfer learning is a strategy for machine learning in which a model trained on one job is repurposed for a second task that is linked to the first. The Common Objects in Context (COCO) dataset contains Microsoft's released large-scale object identification, segmentation, and captioning dataset. The COCO model comprises a

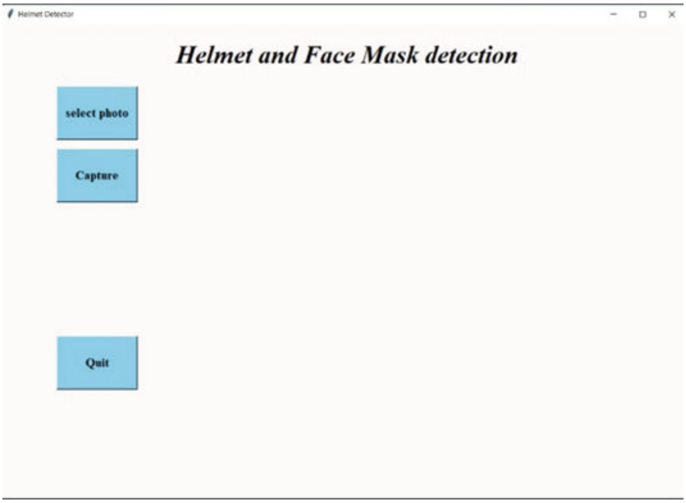


Fig. 2 User interface

massive amount of data that has been pre-trained. You Only Look Once (YOLO) is a transfer learning technique that is also used to train the data in YOLOV4. YOLO is one of the fastest learning and object detection systems. We employ YOLO weights and cfg files, with the cfg file including convolution layer configuration information such as batches, subdivisions, and learning rate. HAAR (Haar cascade frontal face detection) file may be used by any scripting language, including Java, Python, and CPP, to determine the real-time environment.

3. **Extracting the Characters from API (Plate Recognizer)**

Application programming interface (API) is a software code that helps two different software's communicate and exchange data. In our case, we will get the characters from the number plate using this API named plate recogniser.

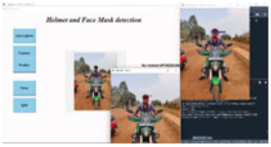
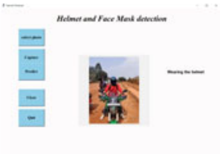
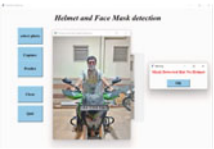
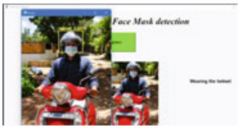
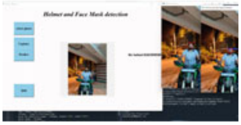
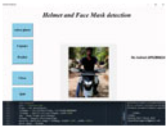
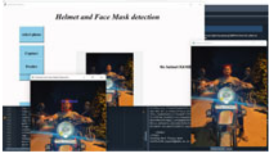
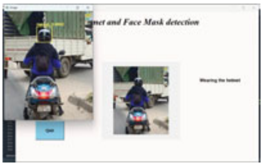
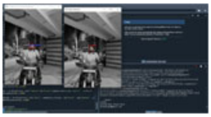
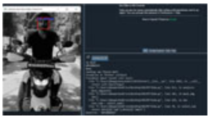
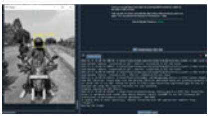
4. **Sending an Email and SMS**

When the system detects that the rider is not wearing a helmet or not wearing a mask, characters are extracted from the number plate characters using API stored it. We included the user defined set of details for vehicle number. As of now, for only those registered details, we can send mail. An email was sent using the Python libraries like smtplib, SSL, and requests to the concerned person with the image.

6 Results and Discussion

Table 1 depicts the images considered for the test cases, and Table 2 depicts the description of the test cases with various possibilities.

Table 1 Images considered for the test cases

 **Test case 1**	 **Test case 2**	 **Test case 3**
 **Test case 4**	 **Test case 5**	 **Test case 6**
 **Test case 7**	 **Test Case 8**	 **Test case 9**
 **Test case 10**	 **Test case 11**	 **Test case 12**

Tables 1 and 2 represent the test cases with each case being unique. In most instances, when the select button was pressed, the rider's motion ceased. All photos of test cases are captured in real time. We collected data from several vehicles with various riders. As described in Table 2, test cases and their speeds are gathered and various testing are conducted.

In Test cases 2, 4, and 8, photographs are captured from the front. In Test instance 4, the image was captured when the vehicle was moving at 7 km per hour. The system detects and provides an accuracy of 89% (here the helmet colour and background colour are same), 94% (the accuracy varies owing to motion), and 96%. (as the system was clear with the image).

In Test case 1, Test case 5, Test case 6, and Test case 7, the system recognises that the riders are not wearing a helmet and face shield and sends a warning message to the registered number of the vehicle. In Test case 3, a picture is acquired from a lower perspective of a rider wearing only a mask. The system recognises and sends an alarm message and displays that the rider is not wearing a helmet.

The system fails to recognise the cyclist in Test case 9 because the photo is blurry and the rider is travelling at 40 km/h. Ten, eleven, and twelve are black and white.

Table 2 Test cases for different possibilities

Test cases	Speed of the vehicle in (km/h)	Wearing helmet or not	Wearing mask or not	Number plate extracted or not	Output	Image dimensions
Test case 1	NA	No helmet	No mask	NA	No mask no helmet—email sent	960 × 1046
Test case 2	NA	Helmet	No mask	NA	Helmet—89%	960 × 1032
Test case 3	NA	No helmet	Mask	NA	Mask detected no helmet	960 × 1280
Test case 4	7	Helmet	No mask	NA	Wearing helmet—94%	1280 × 775
Test case 5	NA	No helmet	Mask	Yes	No helmet—email sent	1200 × 1600
Test case 6	NA	No helmet	No mask	Yes	No mask no helmet—email sent	655 × 829
Test case 7	5	No helmet	No mask	Yes	No mask no helmet—email sent	1200 × 1600
Test case 8 (back view)	NA	Helmet	NA	No	Wearing the helmet—96%	1032 × 720
Test case 9	40	No helmet	No mask	No	Not detected	790 × 1080
Test case 10 (black and white)	NA	No helmet	No mask	Yes	No mask no helmet—email sent	1200 × 1480
Test case 11 (black and white)	NA	No helmet	No mask	No	Email not sent	1200 × 1480
Test case 12 (black and white)	NA	Helmet	NA	No	Wearing helmet—89%	900 × 1280

In Test case 10, the detection is accurate and an alert email is sent. In Test case 11, it detects a biker without a helmet but is unable to extract the correct number plate characters. In Test case 12, the rider is detected correctly.

In some test situations of black-and-white photos, the algorithm detects them based on image lighting, number plate orientation, and other criteria. Black helmets with black visors and backgrounds can occasionally mislead the system. Figure 3 represents the result of the person as he is not wearing both a helmet and mask. Figure 4 represents the result of the rider at rest as he is not wearing a helmet.

Figures 5, 6, and 7 represent the output as system detects the rider violating rules and sends an alert mail with attached photo document.

Challenges Faced

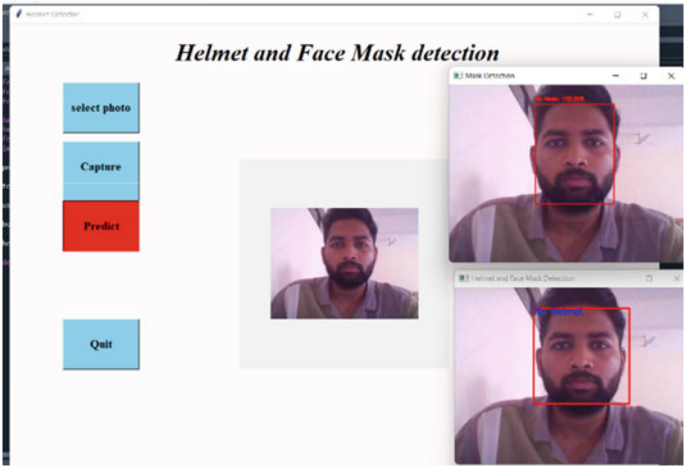

Helmet and Face Mask detection
select photo
Capture
Predict
Quit
Mask Detection
Helmet and Face Mask Detection

Fig. 3 Detecting person using capture button

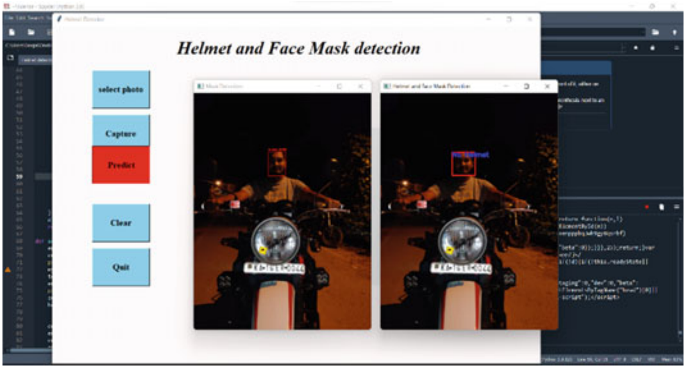
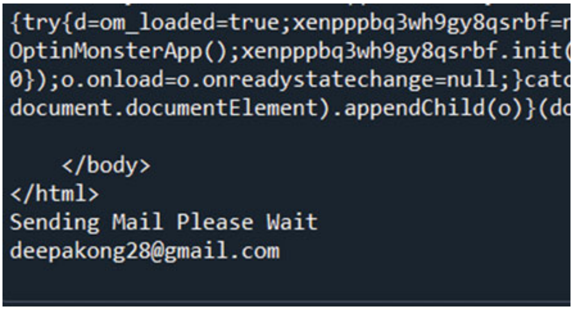
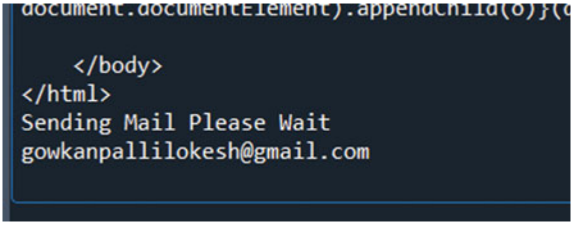


Fig. 4 Detecting rider status using the select button



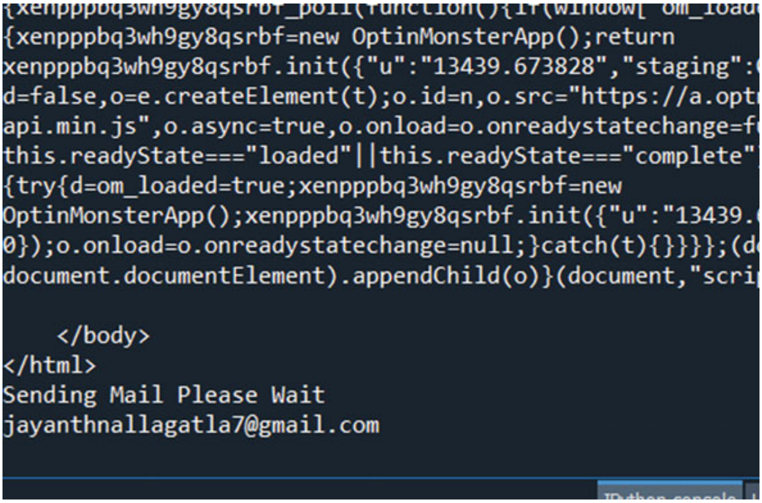
```
{try{d=om_loaded=true;xenpppbq3wh9gy8qsrbf=n
OptinMonsterApp();xenpppbq3wh9gy8qsrbf.init(
0});o.onload=o.onreadystatechange=null;}catc
document.documentElement).appendChild(o)}(dc

    </body>
</html>
Sending Mail Please Wait
deepakong28@gmail.com
```

Fig. 5 Sending mail to deepakong28@gmail.com



```
document.documentElement).appendChild(o)}(d

    </body>
</html>
Sending Mail Please Wait
gowkanpallilokesh@gmail.com
```

Fig. 6 Sending mail to gowkanapallilokesh@gmail.com



```
{xenpppbq3wh9gy8qsrbf_poll(function(){if(window["om_load
{xenpppbq3wh9gy8qsrbf=new OptinMonsterApp();return
xenpppbq3wh9gy8qsrbf.init({"u":"13439.673828","staging":
d=false,o=e.createElement(t);o.id=n,o.src="https://a.opt
api.min.js",o.async=true,o.onload=o.onreadystatechange=f
this.readyState==="loaded"||this.readyState==="complete"
{try{d=om_loaded=true;xenpppbq3wh9gy8qsrbf=new
OptinMonsterApp();xenpppbq3wh9gy8qsrbf.init({"u":"13439.
0});o.onload=o.onreadystatechange=null;}catch(t){}}}};(d
document.documentElement).appendChild(o)}(document,"scri

    </body>
</html>
Sending Mail Please Wait
jayanthnallagatla7@gmail.com
```

Fig. 7 Sending mail to jayanthnallagatla7@gmail.com

- In some cases, the system detects multiple helmets due to the windshield and headlight.
- It fails to give the desired results when we upload an image of some exceptional cases.
- The image should be sharp and proper to get more accuracy, and blurry images make the system undetectable.
- Although taking pictures from a side view is a bit challenging because only half face is visible and the number plate is not visible.
- The accuracy of the system gets deteriorated when the real-time images in motion was considered.
- It is a bit tough to detect the number plate for some of the motorcycles in dull light.
- Performance of the system is superior when the RGB colour images were considered over the mere black and white or grey scale images.

Future Work

- The challenges faced can be rectified for better results.
- Should consider more test cases and execute.
- System accuracy can be improved.
- By using any government API's, system can be able to collect the rider information and record as well.

7 Conclusion

The objective of identifying the motorcycle riders has been met. This system can detect whether or not a two-wheeler rider is wearing a helmet, and if the rider is not wearing a helmet, the system proceeds to detect the number plate, and the image of this number plate will be passed to the API (plate recognizer) for text extraction, which would be the vehicle number, and we will send a notification to the concerned person via email and SMS regarding the rule violation. The algorithms we employ detect colour images more precisely than black and white. In the proposed system, motorcyclists who do not wear helmets or face masks are in violation of the law.

References

Achyutha PN, Hebbale S, Vani V (2022) Real time COVID-19 facemask detection using deep learning. Int J Health Sci 6(S4):1446–1462

Benyang D, Xiaochun L, Miao Y (2020) Safety helmet detection method based on YOLO v4. In: 2020 16th international conference on computational intelligence and security (CIS), pp 155–158. https://doi.org/10.1109/CIS52066.2020.00041

Dey SK, Howlader A, Deb C (2021) MobileNet mask: a multi-phase face mask detection model to prevent person-to-person transmission of SARS-CoV-2. In: Proceedings of international conference on trends in computational and cognitive engineering. Springer, Dhaka, Bangladesh, pp 603–613, Dec 2021

ErdincKocer H, KursatCevik K (2011) Artificial neural networks-based vehicle license plate recognition. Procedia Comput Sci 3:1033–1037

Chen L, Chang W, Su J, Chen Y (2018) i-Helmet: an intelligent motorcycle helmet for rear big truck/bus intimation and collision avoidance. In: 2018 IEEE international conference on consumer electronics (ICCE), Las Vegas, NV, pp 1–2

Kashyap A, Suresh B, Patil A, Sharma S, Jaiswal A (2018) Automatic number plate recognition. In: 2018 international conference on advances in computing, communication control and networking (ICACCCN), pp 838–843. https://doi.org/10.1109/ICACCCN.2018.8748287

Kabir KMA, Tanimoto J (2020) Evolutionary game theory modelling to represent the behavioural dynamics of economic shutdowns and shield immunity in the COVID-19 pandemic. R Soc Open Sci 7(9):201095

Khan M, Chakraborty S, Astya R, Khepra S (2019) Face detection and recognition using OpenCV. In: 2019 international conference on computing, communication, and intelligent systems (ICCCIS), pp 116–119. https://doi.org/10.1109/ICCCIS48478.2019.8974493

Loey M, Manogaran G, Taha MHN, Khalifa NEM (2021) A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. Measurement 167:108288

Mohammed Shariff AS, Bhatia R, Kuma R, Jha S (2021) Vehicle number plate detection using python and open CV. In: 2021 international conference on advance computing and innovative technologies in engineering (ICACITE), pp 525–529. https://doi.org/10.1109/ICACITE51222.2021.9404556

Qin B, Li D (2020) Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19. Sensors 20(18):5236

Razavi M, Alikhani H, Janfaza V, Sadeghi B, Alikhani E (2021) An automatic system to monitor the physical distance and face mask wearing of construction workers in COVID-19 pandemic

Sharma A, Pathak J, Prakash M, Singh JN (2021) Object detection using OpenCV and python. In: 2021 3rd international conference on advances in computing, communication control and networking (ICAC3N), pp 501–505. https://doi.org/10.1109/ICAC3N53548.2021.9725638

Siradjuddin IA, Reynaldi, Muntasa A (2021) Faster region-based convolutional neural network for mask face detection. In: 2021 5th international conference on informatics and computational sciences (ICICoS), pp 282–286. https://doi.org/10.1109/ICICoS53627.2021.9651744

Vasudevan V, Gounder MS (2021) Advances in sports video summarization—a review based on cricket videos. In: Advances and trends in artificial intelligence. From theory to practice: 34th international conference on industrial, engineering and other applications of applied intelligent systems, IEA/AIE 2021, Kuala Lumpur, Malaysia, July 26–29, 2021, proceedings, part II. Springer-Verlag, Berlin, Heidelberg, 347–359. https://doi.org/10.1007/978-3-030-79463-7_29

Wang Z, Wang P, Louis PC, Wheless LE, Huo Y (2021) Wearmask: fast In-browser face mask detection with serverless edge computing for COVID-19

Zhang X, Saleh H, Younis EM, Sahal R, Ali AA (2020) Predicting coronavirus pandemic in real-time using machine learning and big data streaming system. Complexity 2020:1–10

Methods for Improving User Position Accuracy in IRNSS (NavIC) Receiver

Wasim Ahmed Selvakumar G. Abhilasha Barla and Valluri Sarimela

1 Introduction

Indian Regional Navigation Satellite System (IRNSS) Receiver named as Navigation with Indian Constellation (NavIC) is ISRO's initiative to build an independent and regional satellite navigation system based on a constellation of GEO and GSO satellites. An IRNSS receiver is used for precise determination of position and time of the receiver from the solutions as per the specified mode of configurations for Civilian SPS and Strategic Defense (RS) users. The IRNSS architecture mainly consists of space segment, ground segment and user segment. Figure 1 depicts the architecture of IRNSS.

1.1 IRNSS Space Segment

Based on various considerations, the IRNSS constellation is worked out to be a combination of GSO and IGSO satellites.

1.2 IRNSS Ground Segment

Ground segment is responsible for the maintenance and operation of the IRNSS constellation. The ground segment comprises ISRO Navigation Center, IRNSS Spacecraft Control Facility, IRNSS Range and Integrity Monitoring Stations, IRNSS

W. A. S. G. Abhilasha Barla (✉) · V. Sarimela
Central Research Laboratory, Bharat Electronics Limited, Bangalore, India
e-mail: wasimahmed@bel.co.in

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_35

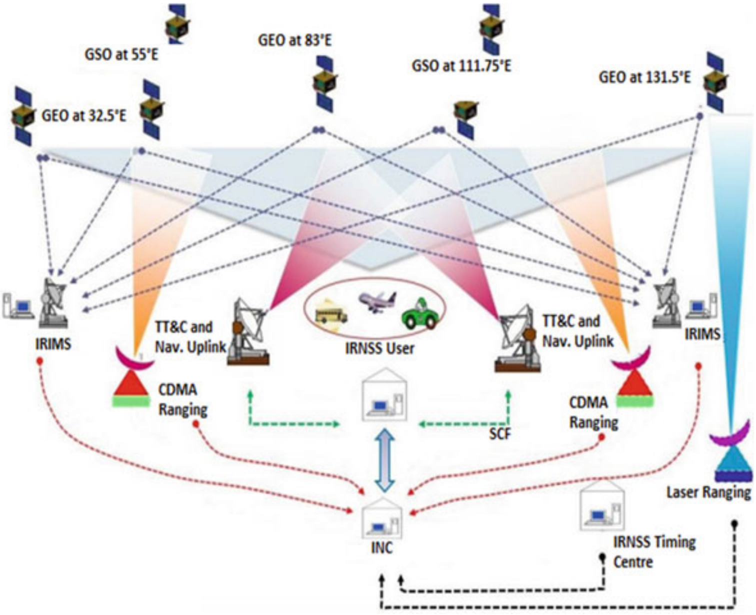


Fig. 1 Architecture of IRNSS

Network Timing Center, IRNSS CDMA Ranging Stations, Laser Ranging Stations and Data Communication Network.

1.3 User Segment

The user segment mainly consists of single-frequency IRNSS receiver capable of receiving SPS signal at L5 or S band frequency, a dual-frequency IRNSS receiver capable of receiving both L5 and S band frequencies and receiver compatible to IRNSS and other GNSS signals.

The IRNSS (NavIC) receiver operates at L5 band (1176.45 MHz) and S band (2492.028 MHz) frequencies. The IRNSS receiver system provides Position, Navigation and Timing (PNT) Services to both Civilian (SPS) and Strategic Defense Services (RS) users.

The rest of the paper is organized with following sections like navigational data processing, error correction algorithms, realization and implementation of NavIC receiver, test results and conclusion.

Fig. 2 Sub-frame structure

600 symbols	
Sync bit	Sub-frame
16 bit	584 Symbols

2 Navigation Data Processing

2.1 Sub-frame Architecture

IRNSS main frame consists of 2400 symbols divided into four sub-frames (Indian Space Research Organization 2014). Each sub-frame is rate 1/2 convolution encoded at 50 sps and ends up in 584 symbols followed by 16 bit sync word. The sub-frame architecture is shown in Fig. 2.

After receiving encoded navigation data, it is being de-interleaved and decoded using Viterbi decoder at the receiver side. Out of four sub-frames, sub-frame 1 and 2 transmit navigation parameters mainly consist of clock correction, ephemeris data, etc., of satellite vehicles, while sub-frames 3 and 4 consist of ionospheric grid delays and correction parameters, etc. Since IRNSS does not send satellite vehicle position in convenient format, instead it sends ephemeris data in sub-frame structure. Parsing of the parameters has been carried out as mentioned in the interface control document (ICD) (Indian Space Research Organization 2014) and used it for PVT algorithm.

2.2 Satellite-Related Calculations

The calculation of satellite position starts once the receiver tracks and acquires IRNSS satellites. Each satellite transmits its position information using low rate data modulation on spread spectrum signal. The position information is in the form of Keplerian (Kaplan and Hegarty 2005, 2017) orbital parameters. Receiver is able to calculate precise satellite position using these parameters once it tracks the signals.

To calculate the satellite position, primary navigation parameters and transmit time are required. Transmit time is nothing but the IRNSS system time at the time of transmission of satellite signal. Primary navigation parameters include correction parameters of system clocks, satellite orbits, satellite health and many other parameters. The value of transmit time will be given by baseband processing algorithms. The parsed ephemeris data have been processed utilizing the algorithm mentioned in ICD to solve for satellite vehicle position (Indian Space Research Organization 2014). The algorithm has been solved iteratively to reduce the error.

This algorithm solves for three unknown parameters listed as: x_i, y_i, z_i which describes the position of ith satellite in ECEF coordinate system. Some of the

Table 1 Ephemeris parameters

Parameters	No. of bits	Scale factor	Units
t_{oe}	16	16	s
Δ_n	22	2^{-41}	Semi-circles/s
M_0	32	2^{-31}	Semi-circles
e	32	2^{-33}	Dimensionless
$\sqrt{A}$	32	2^{-19}	Sqrt (m)
Ω_0	32	2^{-31}	Semi-circles
i_0	32	2^{-31}	Semi-circles

Table 2 IRNSS satellite

Coordinates	SV position
x_s	2.087670312380296 * 1.0e + 07
y_s	3.453030581883871 * 1.0e + 07
z_s	-1.242533848037527 * 1.0e + 07

important ephemeris parameters used to calculate satellite position are listed in Table 1.

By using the above ephemeris parameters and algorithm mentioned in ICD, satellite position has been calculated. One of the satellite positions for the tracked IRNSS satellite is mentioned in Table 2.

2.3 User Position Calculation

Receiver calculates distance from four or more visible satellites using time of arrival (TOA) ranging technique. Accurate range measurements are facilitated by the spread spectrum modulated signal transmitted by the satellites. Satellite to user receiver ranges and satellite positions are related according to following equations (when four satellites are being tracked):

$$\rho_1 = \sqrt{(x_1 - x_u)^2 + (y_1 - y_u)^2 + (z_1 - z_u)^2} + ct_u, \tag{1}$$

$$\rho_2 = \sqrt{(x_2 - x_u)^2 + (y_2 - y_u)^2 + (z_2 - z_u)^2} + ct_u, \tag{2}$$

$$\rho_3 = \sqrt{(x_3 - x_u)^2 + (y_3 - y_u)^2 + (z_3 - z_u)^2} + ct_u, \tag{3}$$

$$\rho_4 = \sqrt{(x_4 - x_u)^2 + (y_4 - y_u)^2 + (z_4 - z_u)^2} + ct_u, \tag{4}$$

where ρ_i is user to ith satellite range, x_i, y_i, z_i are positions of ith satellite, x_u, y_u, z_u are positions of receiver, t_u is receiver clock offset and c is velocity of light.

These are four nonlinear equations in four unknowns x_u, y_u, z_u and t_u. These nonlinear equations can be solved for the unknowns by employing closed-form solutions, iterative techniques based on linearization, or Kalman filtering (Kaplan and Hegarty 2005).

Distance measurment is done by Time Of Arrival (TOA) concept. The relative pseudo-range can be obtained by multiplying the propagation delay with speed of light.

$$\rho = c * (t_{\text{recv}} - t_{\text{sent}}), \tag{5}$$

where $(t_{\text{recv}} - t_{\text{sent}})$ is the propagation delay, t_{recv} time at which the satellite will receive signal, t_{sent} time at which the satellite will send signal.

By using Eq. (5), pseudo-range measurement has been carried out. One of the observed pseudo-ranges of the tracked satellite is

$$\rho = 4.202977290026457 * 1.0\text{e} + 07\,\text{m}.$$

Since Eqs. 1, 2, 3 and 4 are nonlinear equation, it is difficult to solve them. In order to obtain the solution, non linear equations are to be linearized by differentiating them. This can be done by Taylor series expansion. The equation looks like this after putting it into matrix form:

$$\begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & 1 \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & 1 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & 1 \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & 1 \end{bmatrix} \begin{bmatrix} \delta x_u \\ \delta y_u \\ \delta z_u \\ \delta b_u \end{bmatrix}, \tag{6}$$

where

$$\alpha_{i1} = \frac{x_i - x_u}{\rho_i - b_u}, \alpha_{i2} = \frac{x_i - x_u}{\rho_i - b_u}, \alpha_{i3} = \frac{x_i - x_u}{\rho_i - b_u}.$$

Equation (6) can be written as

$$\Delta\rho = H\Delta x. \tag{7}$$

If only four satellites of the IRNSS constellation are visible, then user can easily calculate user position since H is square 4 × 4 matrix by solving Eq. (8) by least square approximation method:

$$\Delta x = H^{-1}\Delta\rho. \tag{8}$$

Table 3 IRNSS constellation

Coordinates	User position
x_u (Mt)	2.087670312380296 * 1.0e + 07
y_u (Mt)	3.453030581883871 * 1.0e + 07
z_u (Mt)	− 1.242533848037527 * 1.0e + 07
Latitude (°)	13.056532927185359
Longitude (°)	77.553060802014460
Altitude (Mt)	858.98

But if more than four satellites are visible, then H can no longer be a square matrix, so it cannot be inverted directly. Since the equations are more than unknowns, least square approximation method is used to find the solution and Gauss–Jordan elimination method is used to find the inverse of the matrix given in Eq. (9).

$$\Delta x = \left[H^T H\right]^{-1} H^T \Delta\rho. \tag{9}$$

After solving $\left[H^T H\right]^{-1} H^T$, square matrix will be obtained.

Equation (8) can be used directly.

By using the above equations, user position has been calculated. The user position for the tracked IRNSS constellation is mentioned in Table 3.

3 Error Correction Algorithms

3.1 Earth Rotation Correction

In the Earth-Centered Earth-Fixed (ECEF) coordinate frame, the satellite position has now been solved using the algorithm mentioned in ICD (Indian Space Research Organization 2014). For the coordinate frames, there is one more major change to be done: Earth rotation correction. The satellite signal will fly about 72 ms until it hits the Earth's receiver antenna. The planet rotated slightly around its axis at that time. This effect is called as Sagnac effect. Sagnac effect can lead to position errors of about 30–50 m. The compensation is completed with a straightforward matrix rotation from ECEF to Earth-Centered Inertial (ECI) coordinate frame.

$$x_i' = r * (\cos(\Omega_{\text{er}}) * \cos(\varphi) - \sin(\Omega_{\text{er}}) * \sin(\varphi) * \cos(i)), \tag{10}$$

$$y_i' = r * (\sin(\Omega_{\text{er}}) * \cos(\varphi) + \sin(\Omega_{\text{er}}) * \sin(\varphi) * \cos(i)), \tag{11}$$

$$z_i' = r * \sin(\varphi) * \sin(i), \tag{12}$$

where φ is corrected argument of latitude, i is corrected inclination, r is corrected radius, Ω_{er} is corrected longitude of ascending node and x_i', y_i', z_i' are the error corrected position of the ith satellite (Tsui 2000).

3.2 Atmospheric Correction

The orbit of the IRNSS satellites is approximately 26,000 km from the center of the Earth. For the IRNSS receiver, the assumption is that it is located on the Earth's surface. As a result, satellites usually transmit their signal at least 20,000 km from the receiver. Most of these signals travel in space, which can be viewed as a vacuum. The signal reaches the Earth's atmosphere can no longer be considered to be in vacuum and undergoes atmospheric refraction, which affects signal propagation.

Grid-based ionospheric algorithm has been integrated to deal with the delays introduced due to ionosphere (Doberstein 2012). To reduce the error caused by ionosphere, satellites will broadcast ephemeris data in Message type 5 of sub-frames 3 and 4 over the Indian region.

After calculating user position, the corrections are provided at 5 * 5 (°) grids at 350 km altitude and then first compute the point where LOS cuts the 350 km altitude. This point is termed as the Ionosphere Pierce Point (IPP) of the user. Sub-frame 3 and Sub-frame 4 give the information of the grid points using two parameters, GIVD and GIVEI. Take the information from GIVD and GIVEI from the surrounding grid points to its IPP to compute vertical delay (Indian Space Research Organization 2014; Ganeshan et al. 2005). To determine the latitude and longitude of the IPP, the following equations have been used:

$$\psi_{\text{pp}} = \pi/2 - \text{El} - \sin^{-1}((R_e/R_e + h) * \cos(\text{El})), \tag{13}$$

where ψ_{pp} is the Earth's central angle between user and projection on IPP. (φ_{pp}, λ_{pp}) are the latitude and longitude of the pierce point.

$$\begin{aligned} \varphi_{\text{pp}} &= \sin^{-1}((\sin\varphi_u).\cos(\psi_{\text{pp}}) + \cos(\varphi_u).\sin(\psi_{\text{pp}}).\cos(Az)), \\ \lambda_{\text{pp}} &= \lambda_u + \sin^{-1}(\sin(\psi_{\text{pp}})\sin(Az)/\cos(\varphi_{\text{pp}})). \end{aligned} \tag{14}$$

Az and El are the Azimuth and Elevation angles of the satellite from user position (φ_u, λ_u) and h is height of maximum electron density (350 km). The observed Elevation and Azimuth angles are mentioned in Table 4.

Table 4 Elevation–Azimuth angles

SV PRN ID's	Elevation angle (°)	Azimuth angle (°)
2	75.3006	152.7297
3	66.5240	298.8784
5	45.0061	123.6277
9	48.8998	211.9072

4 Implementation

The IRNSS-based receiver system has been realized on a hardware platform. The hardware for the receiver system consist of RF card and Baseband card. The RF card has been realized using COTS RF front end. The RF signal received from the antenna into the RF card is conditioned and fed to the multi-channel navigation receiver. RF card is capable of processing L1 band: 1575.42 MHz and BW = 20 MHz, L5 band: 1176.45 MHz and BW = 20 MHz, and S band: 2492.028 MHz and BW = 16.5 MHz. The received signal is processed and converted to the desired IF. The tracking sensitivity of the receiver is -139 dBm. The minimum input signal level required at receiver input port of GNSS receiver is -95dBm for proper receiver operation. The receiver is capable of receiving signals from L5 band frequencies. Here, we have implemented a seven-channel IRNSS SPS receiver on FPGA with the use of COTS RF front end. The RF signal at 1776.45 MHz (L5 band signal) is converted to IF using COTS RF front end and digitized using 12-bit ADC. Three bits of this digitized signal is used as input to the IRNSS receiver as shown in Figs. 3 and 4.

Baseband processing is achieved using a combination of FPGA processing blocks and baseband software residing in ARM processor. Generally, receiver functions are divided into two categories, i.e., baseband receiver processing and navigation

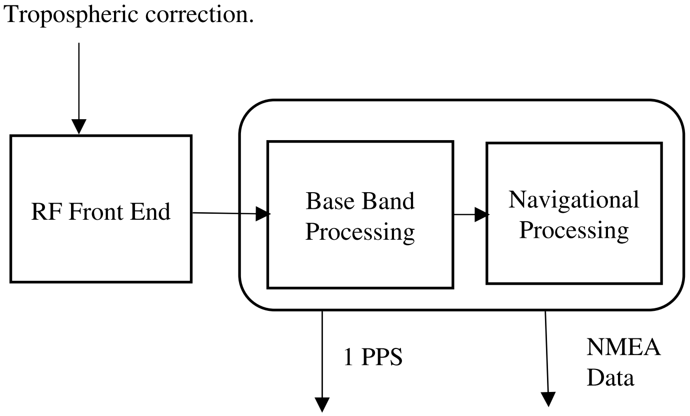


Fig. 3 Overall system block diagram

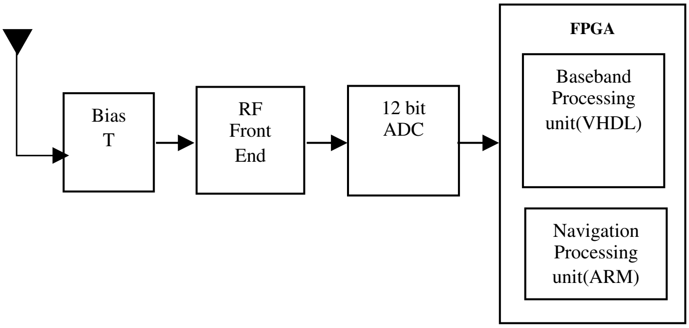

FPGA
Bias T
RF Front End
12 bit ADC
Baseband Processing unit(VHDL)
Navigation Processing unit(ARM)

Fig. 4 Functional block diagram

processing. The function of baseband receiver processing is to acquire and track the visible satellites. Once the receiver is able to track four or more satellites, it can then proceed to navigation processing for computing the position and time of the receiver. The baseband card is realized using FPGA-based hardware platform, which receives the IF input from RF card, digitizes it and processes it to extract the navigational details such as position, time and 1PPS. The FPGA SoC contains FPGA and two ARM Cores. The baseband processing modules are implemented on FPGA using VHDL language. The navigational algorithms are implemented on ARM SoC using C language. Navigation processing is done using software running in ARM processor. In this paper, our main focus is on navigation data processing and its performance after including error-correcting algorithms like Earth rotation algorithm, ionospheric error correction, tropospheric correction.

5 Test Results

The following test results are observed as mentioned in the below figures. The precise location of antenna is calculated by keeping 1 m accuracy and 1 h run time in reference GNSS receiver under survey in-mode. The measured antenna position is set as a reference point in the plots shown in Figs. 5, 6, 7, 8 and 9.

Test Case1: The result shown in Fig. 6 is without error correction algorithms. In this case, position of antenna as center reference point is manually fed and observed the positions with an offset of 30–40 m and positions are scattered in the east direction. In this case, the reference point is 13.05651095, 77.55305911.

Test Case2: The result shown in Fig. 7 is after implementing Earth rotation algorithm. Test case 1 was repeated after implementing Earth Rotation Correction and observed the output as shown below in Fig 7. After implementing Earth rotation algorithm, the plot is observed and shifted toward the reference point reducing the offset of about 40 m.

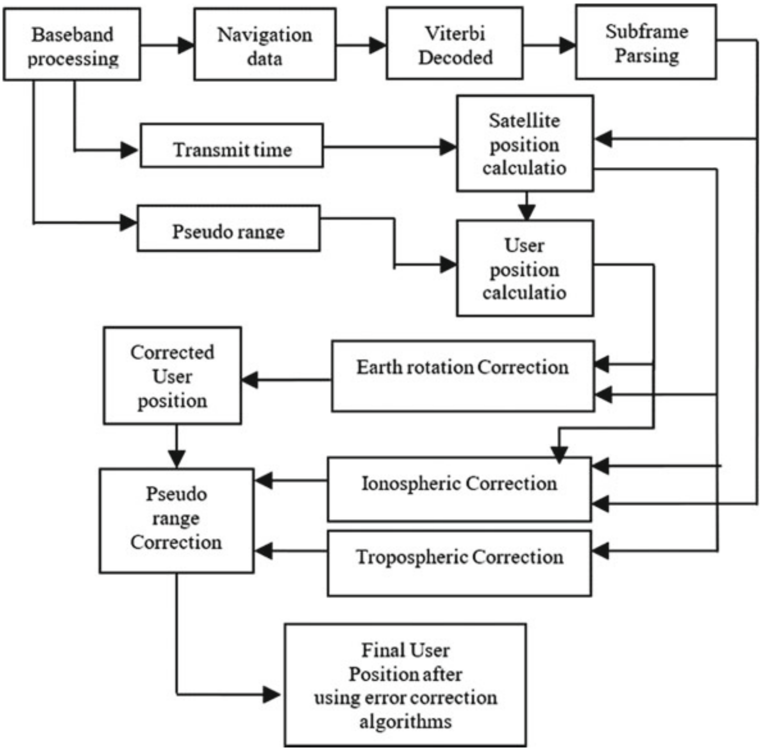


Fig. 5 Navigation unit processing

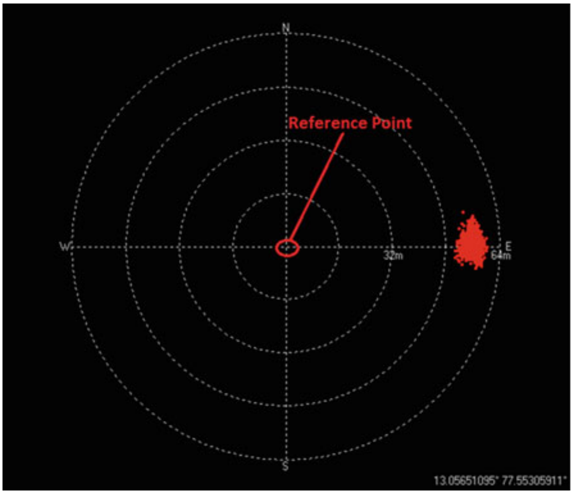


Fig. 6 Receiver position without Earth rotation correction

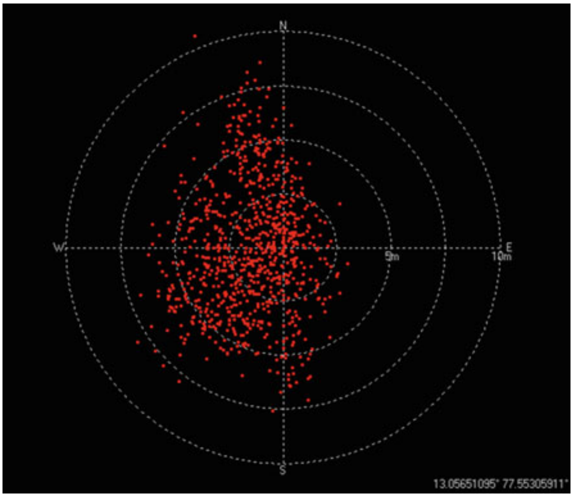


Fig. 7 Receiver position after implementing Earth rotation correction

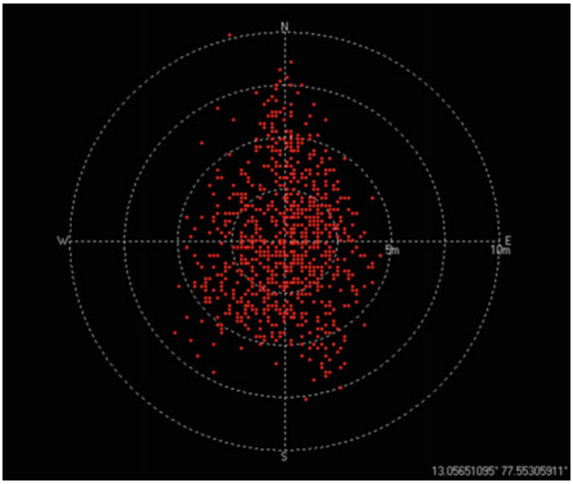
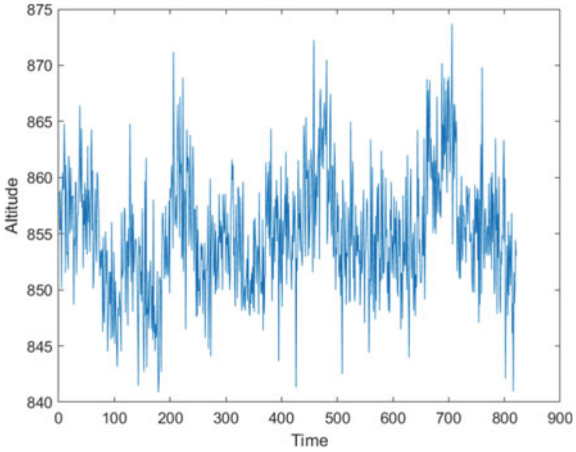


Fig. 8 Receiver position after implementing ionospheric correction

Test Case3: The result shown in Fig. 8 is after implementing ionospheric grid-based algorithm to Case 2. Here, in this case, it is observed that the position is shifted slightly toward the right side. The accuracy in this case is around 5 m after introducing ionospheric algorithm. In Matlab simulation, it is observed that the ionospheric delay for each satellite is around 7–12 m.

Fig. 9 Altitude with time



6 Conclusion

The IRNSS-based GNSS receiver as per the ICD has been realized. Various software modules for 7-channel correlators, signal processing algorithms, navigational algorithms, data acquisition algorithms, PVT calculation algorithms and error correction algorithms (ionosphere, troposphere errors) were developed. Matlab-based simulation has been carried. The modules were ported and tested in the FPGA-based IRNSS receiver system. Accuracy ~ 10 Mts has been achieved. The results with and without error corrections' algorithms are tested. Observed user position accuracy is significantly improved after incorporating error corrections algorithms in the navigational algorithm. To improve further user position accuracy, the entire computation of navigation process and algorithms can be further refined.

References

Doberstein D (2012) Fundamentals of GPS receivers. ISBN 978-1-4614-0408-8

Ganeshan AS, Rathnakara SC, Gupta R, Jain AK (2005) Indian regional navigation satellite system (IRNSS) concept. ISRO Satellite Center J Spacecraft Technol 15(2):19–23

Indian Space Research Organization (2014) Indian regional navigation satellite system signal in space ICD for standard positioning service (version 1.0, ISROIRNSS-ICD-SPS-1.0). http://irnss.isro.gov.in/, June 2014

Kaplan E, Hegarty C (2005) Understanding GPS: principles and applications, 2nd edn. Artech House

Kaplan ED, Hegarty CJ (2017) Fundamentals understanding GPS principles and applications, 3rd edn. ISBN-13: 978-1-63081-058-0

Tsui JB-Y (2000) Fundamentals of global positioning system receivers: a software approach. Wiley. ISBN 0-471-38154-3

Specular Reflection Removal Techniques in Cervix Image: A Comprehensive Review

Lalasa Mukku and Jyothi Thomas

1 Introduction

Biomedical image processing using computer vision techniques refers to the use of computer algorithms and software to analyze and manipulate medical images. This includes tasks such as segmentation (Chen et al. 2018; Muhammad et al. 2020) (identifying and isolating specific objects or regions of interest within an image), classification (assigning images to specific categories) (Cai et al. 2015), and registration (aligning images from different sources or perspectives) (El Meslouhi et al. 2010). These techniques can be applied to a broad range of medical image datatypes, including MRI (Chaddad 2015), X-ray (Morid et al. 2021), CT (Yang et al. 2022), and ultrasound imaging. Biomedical image processing aims to improve the accuracy and reliability of medical diagnoses and to facilitate the development of new treatments and therapies.

Artificial intelligence (AI) plays a remarkable role in the field of disease diagnosis through biomedical image processing. The use of AI algorithms and techniques can significantly improve the efficiency and accuracy of medical image analysis, allowing doctors to quickly and accurately identify the presence of diseases or abnormalities in medical images (Virasova et al. 2021). For example, AI can be used to automatically segment images and identify specific structures or regions of interest, such as tumors or abnormalities in the heart or lungs. AI can also be used to classify medical images by assigning them to specific categories based on their characteristics and features. This can help doctors quickly and accurately diagnose diseases and monitor their progression and response to treatment. Additionally, AI can be used to develop predictive models that can help doctors identify patients who are at risk of developing certain diseases and tailor treatment plans based on individual patient needs and characteristics.

L. Mukku (✉) · J. Thomas
CHRIST (Deemed to be University), Kengeri, Bangalore 560074, India
e-mail: mlalasa2020@gmail.com

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_36

Machine learning and deep learning are two important approaches widely used in medical image processing. Machine learning algorithms can be trained to recognize patterns and features in medical images (Ahishakiye et al. 2020), allowing them to automatically identify diseases or abnormalities. Deep learning algorithms, a type of machine learning algorithm that are particularly well suited to medical image processing because they can learn and identify complex patterns and features in images (Yuan et al. 2020). This allows them to provide highly accurate diagnoses and predictions and to improve the reliability and accuracy of medical image analysis. Additionally, machine learning and deep learning algorithms can be used to develop predictive models that can help doctors identify patients who are at risk of developing certain diseases and tailor treatment plans based on individual patient's needs and characteristics (Swarna et al. 2021).

Specular reflections are light reflection spots on the organ surface. Scoping devices are tools used to examine the interior of a body cavity or other hollow structures. These devices typically consist of a long, flexible tube with a light source and a camera at one end, which is inserted into the body through a small incision or natural opening. Scoping devices can be used to diagnose and treat a wide range of conditions, including cancer, digestive disorders, and reproductive problems. Scoping devices can provide detailed real-time images of the interior of the body, allowing doctors to diagnose conditions and perform procedures with greater precision and accuracy.

The moisture present in the organs reflects the light attached to the scoping devices to form specular reflections (SR). Specular reflections appear as small white areas, sometimes mimicking key diagnostic features, therefore, skewing the diagnosis results. This study aims to review specular reflection removal methods using artificial intelligence (AI) methods in colposcope images. Figure 1 displays a colposcope image with SR, cervical orifice, and acetowhite lesions.

Acetowhite lesions (AW) are abnormalities that appear white or pale in color when viewed under post-acetic acid application to the uterine cervix. In the context of cervix images, acetowhite lesions may be indicative of precancerous changes in the tissue (Li et al. 2008). They are identified through colposcope images. Since specular reflections have the same morphological appearance as acetowhite lesions, the diagnosis will be hindered by SR (Chen et al. 2018). In order to overcome the said drawback, it is essential to remove SR before classifying the cervigram. Over the last couple of decades, several researchers have proposed SR removal techniques using machine and deep learning methods.

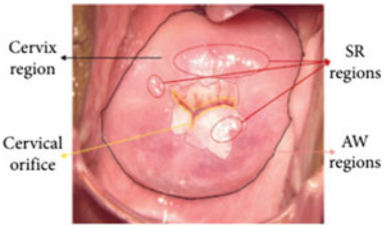


Fig. 1 Cervix image with lesions and SR

This paper systematically reviews the methods used for specular reflection identification, removal, and inpainting. The paper is structured as follows. Section 2 reviews the removal techniques Sect. 3 discusses the inpainting methods. Section 4 puts forth a straightforward solution to specular reflection handling. Section 5 details the evaluation metrics used, and Sect. 6 provides results and discussion. Section 7 concludes the study.

2 Review of Literature

The fundamental principle of specular reflection areas involves the reflection of light from a smooth, shiny surface. This type of reflection produces a clear and sharp image of the light source as opposed to diffuse reflection, which produces a more scattered and diffuse image. The angle at which the light strikes the surface, as well as the angle at which it is reflected, plays a critical role in determining the characteristics of the reflected light. Additionally, the smoothness and shininess of the surface can affect the clarity and sharpness of the reflected image.

Specular reflections obstruct the efficient analysis of cancerous organs. For instance, Morid et al. (2021) has explored the role of SR in confusing the endoscope procedure. SR removal has two phases. The first is to locate the specular region and remove the SR pixels. The second is to paint these areas back to their original morphology. In the detection phase, generally, the image is transformed into a different color space to facilitate further processing of the region of interest (ROI). For instance, the image formats used are RGB (Asiedu et al. 2019), grey-level (Shen et al. 2020), HSV (Li et al. 2019), HSI (Li et al. 2020; Yuan et al. 2017), and a threshold value to identify the SR. Subsequently, the removed pixels are replaced with inpainting to preserve the image morphology (Table 1).

Chaddad (2015) has employed saliency maps for recognizing the SR in combination with a dynamic search-based inpainting algorithm with an adaptive window pixel. In Morid et al. (2021), the authors proposed ELNet model, which averages the neighboring pixel values using the Navier-stokes method to rid the specular reflections followed by image segmentation. Yang et al. (2022) on the other hand, made use of a graph cut-based inpainting technique. Virasova et al. (2021) has adopted the RGB image thresholding method to identify the SR and subsequently filled the SR regions with inward interpolation. Despite using standard morphological methods for SR detection, Ahishakiye et al. (2020) has made use of *Laplace's* equation for inward interpolation to fill the SR pixels. In Yuan et al. (2020), authors used HSI image's 'I' and 'S' channel thresholding with Sobel operator for identification and exemplar-based model for image inpainting. Parallelly, Swarna et al. (2021) experimented with hybrid color attributes and wavelet-based edge projection-based identification, and Dynamic search-based inpainting with adaptive windowing. In Li et al. (2008), DRM (dichromatic reflection model) based on physical methods was employed to automatically remove SR regions and to fill in objects taken in a linear fashion, emphasizing both colour and texture. Shen et al. (2020) modelled RGB images as a gaussian

Table 1 An overview of specular reflection methods

Study	Method of specular reflection and replacement
Wu et al. (2021)	The authors proposed ELNet model, which averages the neighboring pixel values using the Navier-Stokes method
Liu and Caselles (2012)	Graph cut based inpainting technique
Asiedu et al. (2019, 2021)	Identification using RGB image thresholding and painting through inward interpolation
Das et al. (2011a)	Standard morphological methods for detection and Inward interpolation using Laplace's equation for replacement
Chaddad (2015)	Saliency maps and through adaptive window dynamic search-based inpainting algorithm
Yue et al. (2021)	HSI image's 'I' and 'S' channel thresholding with Sobel operator for identification and Exemplar base image inpainting
Alsaleh et al. (2015)	Hybrid color attributes and wavelet-based edge projection-based identification, combined with dynamic search-based inpainting utilizing adaptive windowing, is an effective method for image restoration
Shen et al. (2008)	A physical-based Dichromatic Reflection Model (DRM) is developed to automatically remove spectral regions, in-paint linear combinations of objects, and emphasize color transitions
Zimmerman-Moreno and Greenspan (2006)	SR is detected through Hue Saturation Intensity (HSI) format image polarized with filters, and the pixels for inpainting are filled with color information from the surrounding pixels
Van Raad (2003)	Gaussian mixture modelling on RGB image
Praba and Ranganathan (2013)	Gaussian mixture modelling on HSI image
Lange (2005)	In the RGB color space of cervix image, using the R channel, the authors completed threshold detection in an adaptive manner
Das et al. (2011b), Das and Choudhury (2017)	Similar to Lange (2005), in RGB color space using the intersection of all channels, authors employed threshold detection in adaptive manner
Gordon et al. (2006)	Utilizing predetermined cut off points, areas with high luminosity and low saturation are identified and chosen regions of SR were used to adjust the pixel range in the S-V space, which was mapped from the HSV color palette
El Meslouhi et al. (2010)	SR features in HSV colour space automatic luminance-chromaticity comparison

(continued)

Table 1 (continued)

Study	Method of specular reflection and replacement
El Meslouhi et al. (2011)	Automatic luminance-chromaticity comparison in planar *XYZ* color space
Kudva et al. (2017)	Combined the RGB, HSV, and lab color channel's SR to identify specular regions
Wang et al. (2019)	Arnold's approach, coupled with the exemplar-based approach, yielded improved outcomes
Wang et al. (2021)	Integration of local and global features in colposcope images
Suo et al. (2016)	Dichromatic reflection in orthogonal subspace
Akbar and Herman (2016)	The chaotic clonal selection algorithm
Xue et al. (2007)	Morphological top hat transformation

distribution to recognize the SR regions. Li et al. (2019) also used gaussian mixture modelling but on a HSI format of the image. Li et al. (2020) Utilized a technique to distinguish between different colors expressions in red, green, and blue channels that adjust the level of sensitivity depending on the color being observed. Whereas Yuan et al. (2017), Wu et al. (2021) have used adaptive threshold detection but in RGB colour space using the intersection of all channels. Liu and Caselles (2012) Utilizing predetermined upper and lower brightness limits, certain regions of low saturation were identified, leading to the choice of SR candidate zones to modify the range of pixels in the S-V space, derived from the HSV color scheme. Asiedu et al. (2021) the authors filtered the image and converted it to the *XYZ* colour plane. Then, they used the luminance component and normalized luminance component of the HSV colour space to automatically detect features of the super-resolution regions. Das et al. (2011a) adapted an Automatic luminance-chromaticity comparison in planar *XYZ* colour space to detect the SR areas. Yue et al. (2021) combined the SR features derived from RGB, HSV, and Lab colour spaces to identify SR regions. Alsaleh et al. (2015) used the Arnold method combined with the exemplar-based method, and Shen et al. (2008) incorporated the combination of global and local characteristics in colposcope images. Zimmerman-Moreno and Greenspan (2006) has employed dichromatic reflection in orthogonal subspace to detect the SR. Praba and Ranganathan (2013) used a Morphological top hat transformation for SR identification (Fig. 2).

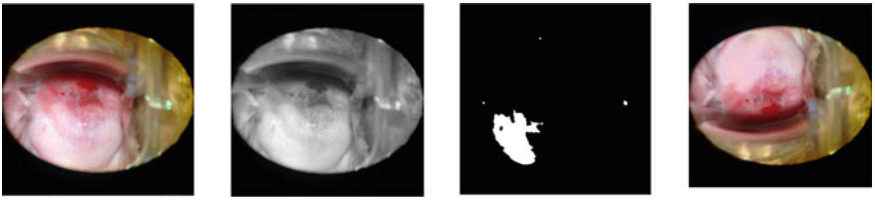

(a) RGB picture (b) HSI image (c) Specular pixels (d) Final image

Fig. 2 The pipeline of specular reflection methodology in cervigram image

3 Specular Reflection Identification and Inpainting of Cervigram Images

A cervigram is the image of the cervix captured by a colposcope (Fig. 1). Authors in Yuan et al. (2020) approached the two-stage solution for specular reflection detection and removal. In the detection phase, exploited the low saturation and high-intensity visual properties of SR pixels converting the image into an HSI (hue, saturation, intensity) format and providing a threshold value to S (saturation) and I (intensity) channels. They chose the Sobel operator to detect the SR boundaries. Coming to the inpainting process, they employed an exemplar-based method wherein the orthogonal vectors, or SR pixels, are considered for generating the filling patches. This method claims to preserve the original cervix tissue color. Considering Ω Omega to be an SR pixel, the detection is given by (1), and inpainting is given by (2).

$$\Omega = \{(x_o, y_o) | S(x_o, y_o) \le \beta_S, I(x_o, y_o) \ge \beta_I, \mathrm{GR}(x_o, y_o) \ge \beta_{\mathrm{GR}}\} \tag{1}$$

$$\mathrm{Sim}(\psi_p, \psi_q) = \left(\frac{\sqrt{(x_{\hat{p}} - x_q)^2 + (y_{\hat{p}} - y_q)^2}}{\sqrt{\mathrm{height}^2 - \mathrm{width}^2}} \right) * \left(\sum_{j=-\frac{\hat{m}}{2}}^{\frac{\hat{m}}{2}} \sum_{i=-\frac{\hat{m}}{2}}^{\frac{\hat{m}}{2}} \left(L(x_{\hat{p}} - i, y_{\hat{p}} - j) - L(x_q - i, y_q - j)\right)^2 \right) \tag{2}$$

Authors in Yue et al. (2021) proposed an algorithm for SR detection on basis of a standard deviation filter, following which they segmented the lesions successfully using a curvilinear structure enhancement. They converted in RGB image to HSV colour space. They combined the 'S' component of colour space and 'G' component of RGB, and the lightness component 'L' of CIE-lab colour space and found it to be able to extract SR pixels accurately. The combined feature image of this subspace is acquired by

$$F = ((1 - S) * G * L)^3 \tag{3}$$

And this feature-mapped image was, in due course, filtered using a standard deviation with filter size 3.

$$\text{Standard Deviation} = \sqrt{\frac{\sum\left(f_i - \overline{f}\right)^2}{m * n - 1}} \tag{4}$$

In Shen et al. (2008), after considering the fact that the SR pixels are spread across in a random fashion on a cervigram, the authors made use of global and local features to identify the SR with higher confidence. Following that, the inpainting is performed using exemplar-based methods. By using the nonlinear filter given in (3), the pixels meeting the criterion in (4) are detected to be SR.

$$\begin{pmatrix} R' \\ G' \\ C' \end{pmatrix} = (1 - S)\begin{pmatrix} R \\ G \\ C \end{pmatrix} = \frac{\min(R, G, B)}{\max(R, G, B)}\begin{pmatrix} R \\ G \\ C \end{pmatrix} \tag{5}$$

$$y \geq \omega Y_{\text{global}} = \omega \frac{Y}{X + Y + Z} \tag{6}$$

In a comparative evaluation of SR removal methods, Praba and Ranganathan (2013) used a morphological top-hat transform on the intensity (I) channel of the HSI image. They used the Otsu method to identify the SR borders in contrast to the Sobel used by Yuan et al. (2020). They also proposed modeling the high brightness (I) and low color saturation (S) into 2-dimensional SI space as models of gaussian functions. To achieve the inpainting, the authors established two possibilities of either using the mean of the pixel color surrounding the SR or the weighted mean of the surrounding pixels.

Asiedu et al. (2019) adopted the same mapping space but loosened the threshold to determine SR regions and non-SR regions more reasonably. The same mapping space was used, but the threshold for distinguishing SR regions from non-SR regions was made more flexible. Furthermore, a Gaussian distribution was applied to them, and then the iterative filling method was utilized to repair and fill in the missing regions. Nevertheless this method necessitates a great deal of pre-existing knowledge and is challenging to use in practice.

4 Proposed Method

The process of specular reflection removal detects and corrects specular reflections to eliminate any uneven visibility. The uneven illumination created by the differences in the intensity of the light sources leads to the appearance of white spots in colposcopy images due to specular reflection. The proposed method works in two stages.

4.1 SR Identification

A specular reflection is a type of reflection in which the reflected light rays are parallel to each other. In other words, the reflection is in the same direction as the incident light. In a bi-dimensional histogram, specular reflection would refer to the symmetrical nature of the histogram when it is reflected along the x-axis or y-axis. This means that the shape of the histogram remains the same after it is reflected, and the relative frequencies of the data points are preserved. Therefore a bi-directional histogram decomposition is used to detect specular reflections whose formula is given in Eq. (7).

$$m = \frac{1}{3}(b + g + r) \tag{7}$$

where 'm' stands for pixel intensity.

$$s = \left\{ \begin{array}{l} \frac{1}{2}\big(2r - g - b = \frac{3}{2}(r - m)\big)\text{in the event}(b + r) > 2g \\ \frac{1}{2}\big(r + g - b = \frac{3}{2}(m - b)\big)\text{in the event}(b + r) \geq 2g \end{array} \right\} \tag{8}$$

$$\left\{ \begin{array}{l} m_p \geq \frac{1}{2}m_{\max} \\ S_p \leq \frac{1}{3}s_{\max} \end{array} \right\} \tag{9}$$

here 's' denotes saturation and $(r, g, b) =$ (red, green, blue) Two important threshold values ($m_{\max}$, $S_{\max}$) determine the specular reflection pixels through bidirectional histogram. Two independent criteria that must be met for a pixel to be considered as SR are given in Eq. (9).

4.2 SR Removal and Inpainting

Image linear correction is a simple and effective way usually employed to improve the quality of an image and is often used as a pre-processing step for more advanced image analysis techniques. It involves applying a linear transformation to the pixel values of the image in order to stretch or compress the range of intensity values. There are several different techniques that can be used for linear image correction. After the transformation, the SR pixels are removed. The pixels must be replaced in such a way that the information of the cervix image is preserved. Routinely, the SR pixels are replaced with the mean of pixels surrounding the pixel that needs to be replaced. In this study, a Laplace equation with a Laplacian kernel is used to fill specular reflection pixels in order to remove glare regions from the underlying image colours.

The Laplacian equation is a partial differential equation that describes the behavior of a two-dimensional surface. The Laplacian equation can be used in image repainting, which is a technique used to restore damaged images. In this context, the

Laplacian equation can be used to identify SR in the image, which can then be used repainting the SR areas. To apply the Laplacian equation in image repainting, the image is first convolved with a Laplacian kernel to enhance the edges and boundaries. The repainting is then performed in the areas of the image that have SR, using the enhanced edges and boundaries as a guide. The final step is to smooth the repainted areas and blend them with the rest of the image, to produce a seamless and natural-looking result. The equation for Laplace transformation is given in Eq. (10).

$$F(s) = \int_{0}^{\infty} f(t)\mathrm{e}^{-st} t' \tag{10}$$

5 Evaluation Metrics

Removing specular reflections from medical images before analysis can help improve the accuracy of the analysis. Specular reflections can interfere with medical imaging algorithms, leading to incorrect or missed diagnoses. By removing specular reflections, the algorithms can then more accurately identify features in the images that may help in diagnosis. This can lead to more accurate diagnoses and improved patient care. Evaluating SR removal process can be done in two ways—qualitative and quantitative.

Quantitative evaluation:

Quantitative evaluation for specular reflection removal methods typically involves measuring the amount of specular reflection remaining in the image after the removal process is completed. This is usually done by calculating the mean squared error (MSE) between the original image and the modified image. It can also involve comparing the area of the specular reflection before and after the modification or measuring the amount of contrast between the original and modified image. Since the MSE is a basic metric, several works had sensitivity index and Jaccard index supplant the desirable evaluation.

$$\text{Sensitivity index(SM)} = \frac{|S \Uparrow G|}{|G|} \tag{11}$$

$$\text{Jaccard index(JI)} = \frac{|S \Uparrow G|}{|S \cup G|} \tag{12}$$

$$\text{Specificity index(SP)} = \frac{|S - G|}{|G|}$$

Qualitative evaluation:

Qualitative evaluation of specular reflection removal methods typically involves visually examining the quality of the resulting image, such as how much remaining specular reflections remain, how smooth the image is, and how accurately the details of the original image have been preserved. It can also consider the speed of the algorithm or the amount of memory required to implement the method. Several works have had the domain expert analyse the performance of their methods. The ablation studies performed by the works have established the superiority of the inpainting and removal methods.

6 Results and Discussion

The proposed method is implemented on a privately acquired dataset of colposcope cervix images after the application of 5% acetic acid. Data will be shared on academic requests. The current method achieved a 0.952824 sensitivity, 0.968281 specificity, 0.948564 dice score, and 0.697252 Jaccard index. Specular reflection removal from medical images before segmentation and classification can be highly beneficial in helping to improve the accuracy of the segmentation and classification system. Removing the specular reflection from the medical images can reduce noise and provide more accurate information about the underlying structure of the tissue. This can help to reduce false positives and make the segmentation and classification process more reliable. Additionally, removing the specular reflection can also make it easier to identify important features and objects in the images, which can further improve the accuracy of the segmentation and classification system. Finally, removing specular reflection can help to reduce the computational complexity of the segmentation and classification system, which can lead to better performance and faster processing times.

7 Conclusion

Overall, the literature review on specular reflection removal methods has shown that there are many different approaches to accomplish this task, each with its own set of advantages and disadvantages. The machine learning and deep learning models have significantly contributed to the identification and removal phases of SR. While some methods are extremely effective in removing specular reflections irrespective of illumination variance, others may be more suitable for specific types of images. In this study, the proposed method achieved a 0.952824 sensitivity, 0.968281 specificity, 0.948564 dice score, and 0.697252 Jaccard index. As the field of image processing continues to evolve, it is likely that new and improved methods for specular reflection removal will be developed in the near future.

References

Ahishakiye E, Wario R, Mwangi W, Taremwa D (2020) Prediction of cervical cancer basing on risk factors using ensemble learning. In: 2020 IST-Africa conference IST-Africa

Akbar H, Herman NS (2016) Removal of highlights in dichromatic reflection objects using segmentation and inpainting. In: 2016 International conference on robotics, automation and sciences (ICORAS). IEEE, pp 1–4

Alsaleh SM, Aviles AI, Sobrevilla P, Casals A, Hahn JK (2015) Automatic and robust single-camera specular highlight removal in cardiac images. In: 2015 37th Annual international conference on IEEE engineering in medicine and biology society, pp 675–678. https://doi.org/10.1109/EMBC.2015.7318452

Asiedu MN, Simhal A, Chaudhary U, Mueller JL, Lam CT, Schmitt JW et al (2019) Development of algorithms for automated detection of cervical pre-cancers with a low-cost, point-of-care, pocket colposcope. IEEE Trans Biomed Eng 66:2306–2318. https://doi.org/10.1109/TBME.2018.2887208

Asiedu M, Ramanujam N, Sapiro G (2021) Methods for automated detection of cervical pre-cancers with a low-cost, point-of-care, pocket colposcope

Cai Z, Xu D, Zhang Q, Zhang J, Ngai S-M, Shao J (2015) Classification of lung cancer using ensemble-based feature selection and machine learning methods. Mol Biosyst 11:791–800. https://doi.org/10.1039/c4mb00659c

Chaddad A (2015) Automated feature extraction in brain tumor by magnetic resonance imaging using gaussian mixture models. Int J Biomed Imaging. https://doi.org/10.1155/2015/868031

Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2018) DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans Pattern Anal Mach Intell 40:834–848. https://doi.org/10.1109/TPAMI.2017.2699184

Das A, Choudhury A (2017) A novel humanitarian technology for early detection of cervical neoplasia: ROI extraction and SR detection. In: 2017 IEEE Region 10 humanitarian technology conference. IEEE, pp 457–460

Das A, Kar A, Bhattacharyya D (2011a) Preprocessing for automating early detection of cervical cancer. In: Proceedings of international conference on information visualisation, pp 597–600. https://doi.org/10.1109/IV.2011.89

Das A, Kar A, Bhattacharyya D (2011b) Elimination of specular reflection and identification of ROI: the first step in automated detection of cervical cancer using digital colposcopy. In: 2011 IEEE International conference on imaging systems and techniques. IEEE, pp 237–241

El Meslouhi O, Allali H, Gadi T, Benksddour YA, Kardouchi M (2010) Image registration using opponent SIFT descriptor: Application to colposcopic images with specular reflections. In: 2010 Sixth international conference on signal-image technology and internet based systems. IEEE, pp 12–17

El Meslouhi O, Kardouchi M, Allali H, Gadi T, Benkaddour YA (2011) Automatic detection and inpainting of specular reflections for colposcopic images. Open Comput Sci 1:341–354. https://doi.org/10.2478/s13537-011-0020-2

Gordon S, Zimmerman G, Long R, Antani S, Jeronimo J, Greenspan H (2006) Content analysis of uterine cervix images: initial steps towards content based indexing and retrieval of cervigrams. Med Imaging 2006 Image Process 6144:61444U. https://doi.org/10.1117/12.653025

Kudva V, Prasad K, Guruvare S (2017) Detection of specular reflection and segmentation of cervix region in uterine cervix images for cervical cancer screening. IRBM 38:281–291

Lange H (2005) Automatic glare removal in reflectance imagery of the uterine cervix. Med Imaging 2005 Image Process 5747(SPIE):2183–2192

Li W, Ferris DG, Lieberman RW (2008) Computerized image analysis for acetic acid induced intraepithelial lesions. Med Imaging 2008 Image Process 6914:69143A. https://doi.org/10.1117/12.769298

Li R, Pan J, Si Y, Yan B, Hu Y, Qin H (2019) Specular reflections removal for endoscopic image sequences with adaptive-RPCA decomposition. IEEE Trans Med Imaging 39:328–340

Li L, Li X, Yang S, Ding S, Jolfaei A, Zheng X (2020) Unsupervised-learning-based continuous depth and motion estimation with monocular endoscopy for virtual reality minimally invasive surgery. IEEE Trans Ind Inform 17:3920–3928

Liu Y, Caselles V (2012) Exemplar-based image inpainting using multiscale graph cuts. IEEE Trans Image Process 22:1699–1711

Morid MA, Borjali A, Del Fiol G (2021) A scoping review of transfer learning research on medical image analysis using ImageNet. Comput Biol Med 128. https://doi.org/10.1016/j.compbiomed.2020.104115

Muhammad M, Zeebaree D, Brifcani AMA, Saeed J, Zebari DA (2020) Region of interest segmentation based on clustering techniques for breast cancer ultrasound images: a review. J Appl Sci Technol Trends 1:78–91. https://doi.org/10.38094/jastt20201328

Praba R, Ranganathan H (2013) Wavelet transform based automatic lesion detection in cervix images using active contour

Van Raad V (2003) Frequency space analysis of cervical images using short time Fourier transform. In: Proceedings of the IASTED international conference on biomedical engineering, vol 1, pp 77–81

Shen D-F, Guo J-J, Lin G-S, Lin J-Y (2020) Content-aware specular reflection suppression based on adaptive image inpainting and neural network for endoscopic images. Comput Methods Programs Biomed 192:105414

Suo J, An D, Ji X, Wang H, Dai Q (2016) Fast and high quality highlight removal from a single image. IEEE Trans Image Process 25:5441–5454. https://doi.org/10.1109/TIP.2016.2605002

Swarna SR, Kumar A, Dixit P, Sairam TVM (2021) Parkinson's disease prediction using adaptive quantum computing. In: 2021 Third international conference on intelligent communication technologies and virtual mobile networks. IEEE, pp 1396–401

Shen H-L, Zhang H-G, Shao S-J, Xin JH (2008) Chromaticity-based separation of reflection components in a single image. Pattern Recognit 41:2461–2469

Virasova AY, Klimov DI, Khromov OE, Gubaidullin IR, Oreshko VV (2021) Rich feature hierarchies for accurate object detection and semantic segmentation. Radioengineering 115–126. https://doi.org/10.18127/j00338486-202109-11

Wang X, Li P, Du Y, Lv Y, Chen Y (2019) Detection and inpainting of specular reflection in colposcopic images with exemplar-based method. In: 2019 IEEE 13th international conference on anti-counterfeiting, security, identification ASID. IEEE, pp 90–94

Wang X, Li P, Lv Y, Xue H, Xu T, Du Y et al (2021) Integration of global and local features for specular reflection inpainting in colposcopic images. J Healthc Eng

Wu Z, Ge R, Wen M, Liu G, Chen Y, Zhang P et al (2021) ELNet: automatic classification and segmentation for esophageal lesions using convolutional neural network. Med Image Anal 67:101838

Xue Z, Antani S, Long LR, Jeronimo J, Thoma GR (2007) Comparative performance analysis of cervix ROI extraction and specular reflection removal algorithms for uterine cervix image analysis. Med Imaging 2007 Image Process 6512:1507–1515

Yang C, Qin L, Xie Y, Liao J (2022) Deep learning in CT image segmentation of cervical cancer: a systematic review and meta-analysis. Radiat Oncol 17:1–14

Yuan Y, Li D, Meng MQ-H (2017) Automatic polyp detection via a novel unified bottom-up and top-down saliency approach. IEEE J Biomed Heal Inform 22:1250–1260

Yuan C, Yao Y, Cheng B, Cheng Y, Li Y, Li Y et al (2020) The application of deep learning based diagnostic system to cervical squamous intraepithelial lesions recognition in colposcopy images. Sci Rep 10. https://doi.org/10.1038/s41598-020-68252-3

Yue Z, Ding S, Li X, Yang S, Zhang Y (2021) Automatic acetowhite lesion segmentation via specular reflection removal and deep attention network. IEEE J Biomed Heal Inform 25:3529–3540. https://doi.org/10.1109/JBHI.2021.3064366

Zimmerman-Moreno G, Greenspan H (2006) Automatic detection of specular reflections in uterine cervix images. Med Imaging 2006 Image Process 6144:61446E. https://doi.org/10.1117/12.653089

An Exploration of State of Art Approaches on Substrate-Integrated Waveguide Technology-Based Antennas for Wireless Applications

Ramprasad Ravula, G. Nagarajan, and P. Elavarasan

1 Introduction

Antenna is one of the significant elements in the Radio Frequency (RF) systems for transmitting or receiving the radio waves or electromagnetic signals from and into the air as the medium or channel. For upcoming progress in communication technology, an antenna is being the important part in the complete communication system. Modern wireless systems are placing increased and focus on antenna designs. Antenna can work on different frequency bands. As per Institute of Electrical and Electronics Engineering (IEEE) standard, the electromagnetic frequency spectrum bands are designated as High-Frequency (HF) band (3–30 MHz), Very High-Frequency (VHF) band (30–300 MHz), Ultra High-Frequency (UHF) band (0.3–1 GHz), L band (1–2 GHz), S band (2–4 GHz), C band (4–8 GHz), X band (8–12 GHz), Ku band (12–18 GHz), K-band (18–26 GHz), Ka band (27–40 GHz), V band (40–75 GHz), and W band (75–110 GHz).

The Microstrip Patch (MSP) antenna is a significant option for wireless communication despite of having a number of important properties like lightweight, less expensive, low volume, low profile, easy manufacturing, and simple fabrication. These microstrip antennas are ineffective in higher frequency range applications due to the narrow wavelength at High Frequencies. Essentially resonating at a single frequency, microstrip patch elements have some drawbacks in terms of considerable losses, radiation leakage, and a constrained power handling capacity.

R. Ravula (✉) · G. Nagarajan
Puducherry Technological University, Pillaichavadi, Puducherry, India
e-mail: falserams.ravula@gmail.com

G. Nagarajan
e-mail: nagarajanpec@ptuniv.edu.in

P. Elavarasan
Rajiv Gandhi College of Engineering, Kirumampakkam, Puducherry, India

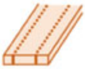
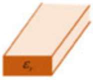
N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_37

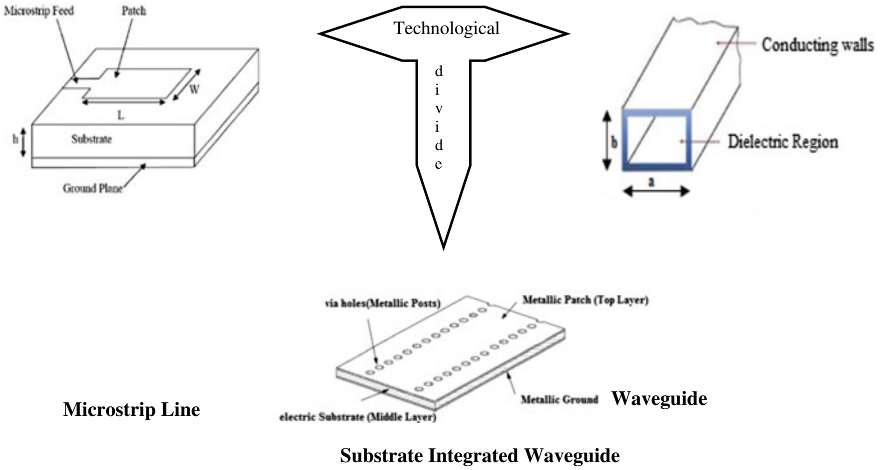
At High Frequencies, waveguide devices are chosen; however, their manufacturing process is difficult and the size is large. Therefore, a new perception is emerging, namely, Substrate-Integrated Waveguide (SIW). It fills the technological gap between microstrip line and Dielectric-Filled Waveguide (DFW). Table 1 represents the comparative analysis between the microstrip line, Substrate-Integrated Waveguide, and waveguide technologies (Wu et al. 2021). Accordingly, Substrate-Integrated Waveguide is harvesting improved parametric results in terms of return loss, gain, directivity, bandwidth, and antenna efficiency as compared with microstrip line and waveguides (Fig. 1).

Millimeter wave (mm wave) applications are being launched and developed in wireless devices and systems have grown in recent years. Several applications, such as Wireless Networks (Daniels 2007), Radars (Fleming 2008) and Medicinal equipment (Mizuno et al. 2007). It is suggested in Lacik and Mikulasek (2011) to use a linearly polarized SIW cavity-backed slot antenna. When an antenna is fed by a coaxial-to-SIW conversion, its radiation comes from a rectangular one-point shorted slot.

The comprehensive survey on SIW-based antenna is divided into seven sections. Section 1 consists of introduction, and in Sect. 2, SIW design and structures are discussed. Section 3 describes methodologies in SIW antenna. In Sect. 4, description and analysis of SIW-based antennas on radiating frequency are presented and depict the comparison of existing SIW-based antenna with proposed structure dimensions, type of dielectric substrate, and also parameters. Section 5 concludes the paper.

Table 1 Comparative analysis of microstrip, SIW, and waveguide

Type	Microstrip	SIW	Waveguide
Figure			
Transmission loss	High	Low	Low
Power handling Capacity	Low	High	High
Physical size	Small	Small	Bulky
Ease of manufacturing	Simple	Simple	Difficult
Integration	Easy	Easy	Difficult
Cost	Low	Low	Expensive
Fundamental mode	EH/HE	TE_{10}	Quasi TEM
Modal dispersion	Critical	Good	Good
Bandwidth	Low	High	Medium



Microstrip Line

Waveguide

Substrate Integrated Waveguide

Fig. 1 Diagram shows Substrate-Integrated Waveguide (SIW)-based technology which divides the technological gap between the microstrip line and waveguide. *Source* Sayyed Arif Ali et al.

2 SIW Design Structures

In this section, the various designs of SIW structures are presented. SIW antenna comprises the top layer as metallic patch, substrate as middle layer, conducting metal vias and ground metal plane as bottom layer which is the same as that of the microstrip patch.

In example, transitions from microstrip to SIW based on a taper (Fig. 2), microstrip feed SIW (Fig. 3), and 3D Sketch of a SIW (Fig. 4) with dimensions are mentioned, and the SIW structure and the microstrip are both integrated onto the same substrate (Deslandes 2010).

Fig. 2 Microstrip to SIW conversion-based taper

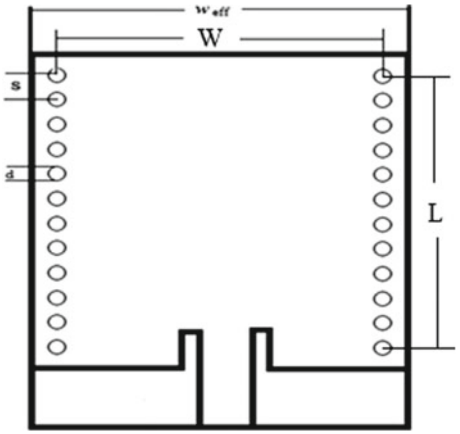


Fig. 3 Microstrip feed SIW

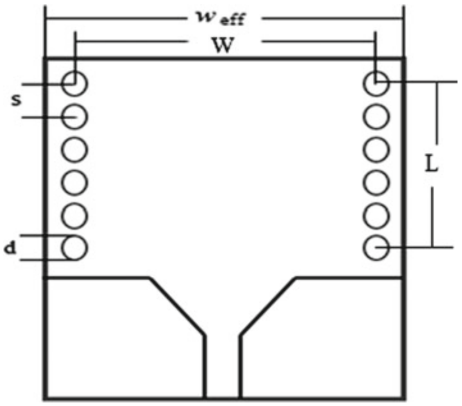
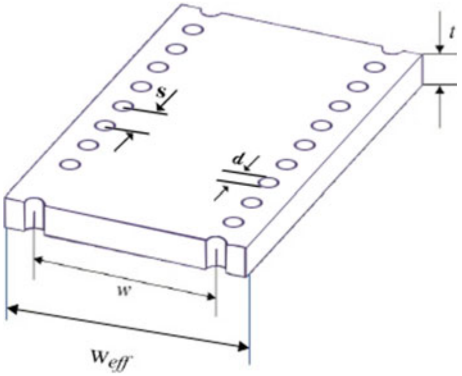


Fig. 4 Representation of SIW in 3D



The various notations have been listed as follows:

w_{eff}, effective width.

W, width of the cavity.

L, length of the cavity.

d, diameter of the metal vias.

s or p, spacing between metal vias.

t or h, thickness.

In order to connect a microstrip implemented in a thin substrate with a thicker SIW design structure, microstrip-to-SIW conversion in a multi-layer substrate has been reported (Ding 2007). Due to the use of thick substrates, reduced conductor losses are possible in SIW constructions. For the implementation and approach of microstrip-to-SIW, design equations have been given in Gupta et al. (2000).

3 Design of SIW Antenna

With some significant variations, the propagation inside a SIW is similar to the propagation inside a conventional hollow waveguide. The propagating modes are at the TE_{10} fundamental mode similar to the waveguide's TE_{n0} mode. There are important formulae that connect the size of a SIW with an effective waveguide with continuous lateral walls which is given in the literature (Cassivi et al. 2002).

The dimensions of the SIW and the effective width 'w_{eff}' of the rectangular shape waveguide with the same propagation parameters have been found to have empirical relationships as a result of the similarities between the two types of waveguides. These relationships make it possible to dimension and design SIW components at preliminary stage without the use of full-wave analysis techniques. One of the design equation relationships is documented in Xu et al. (2005).

$$w_{\mathrm{eff}} = w - \frac{d^2}{0.95s}, \tag{1}$$

where 'd' is the diameter of the metal vias, 'w' signifies width of the cavity, 'w_{eff}' denotes effective width of the SIW, and 's' represents spacing between the vias (Fig. 2).

$$w_{\mathrm{eff}} = w - 1.08\frac{d^2}{s} + 0.1\frac{d^2}{w}. \tag{2}$$

When the 'vias' diameters are wide enough and their distances from one another are narrow enough, there are minimal radiation losses. The vias spacing (s) can be determined empirically as a function of diameter (d) (Bozzi et al. 2008).

$$\frac{s}{d} < 2.5. \tag{3}$$

The modes of propagation can be calculated using the empirical equation (4):

$$fmn0 = \frac{c}{\sqrt[2]{\varepsilon_r}}\sqrt{\left(\frac{m}{L}\right)^2 + \left(\frac{n}{w}\right)^2 + \left(\frac{s}{h}\right)^2}, \tag{4}$$

where $m = 1$, $n = 1$, and $s = 0$ for the fundamental mode (TE_{110}), L and W are the length and width of the cavity, c is the light velocity in free space, and ε_r is the relative permittivity of the dielectric substrate. Accordingly, when the metal vias are closely spaced apart and radiation losses are taken into account, SIW design structures exhibit propagation characteristics that are comparable to those of rectangular-shaped waveguides.

The TE_{mn0} modes, with $m = 1, 2, 3\ldots$ and $n = 1, 2, 3\ldots$, are a subset of the guided modes of the rectangular shape waveguide, in which SIW modes are essentially

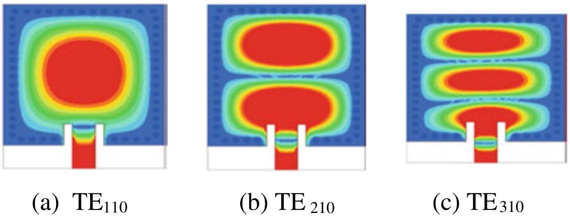

(a) TE$_{110}$ (b) TE$_{210}$ (c) TE$_{310}$

Fig. 5 Propagation modes of SIW in electric field distribution. *Source* Bollavathi Lokeshwar et al.

overlap. Due to the gaps between metal vias, SIW does not support TM modes (Daniels 2007) (Fig. 5).

4 Description and Analysis of SIW Antenna

The detailed exploration analysis of SIW antenna reported based on number of radiating bands (single, dual, triple, and quad) with different substrate materials, thicknesses, dimensions, and feeding techniques which include current advances in the modeling, designs, technological implementations, and applications has been reported.

A. ***Number of Frequency Bands: Single***

- A single-frequency, Substrate-Integrated Waveguide (SIW)-based resonant series slot linearly polarized antenna is projected (Chaturvedi 2020). The antenna is using microstrip line feed technique. The dimensions of the design are 38 × 30 × 1.52 mm^3 which is made up of RO 4003C as dielectric laminate as represented in Fig. 6 with dielectric constant of 3.55. The radiating frequency of the antenna is 5.8 GHz. Two 24° slanted slots and two metallic vias make up the suggested antenna which generates alternating inductive and capacitive loads. At the top metallic surface, the rectangular slots are introduced at a 24° angle from the Y-axis. The obtained peak gain for single band is 7.15 dBi and fractional bandwidth is 5.2%.
- Two bilateral slots' SIW antenna is designed for single frequency (Niu and Tan 2019). The antenna uses coaxial or probe feed technique. The dimensions of the design are 41.8 × 41.8 × 1.57 mm^3 which is made up of F4B-2

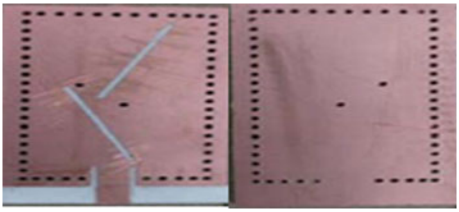


Fig. 6 Top and bottom views

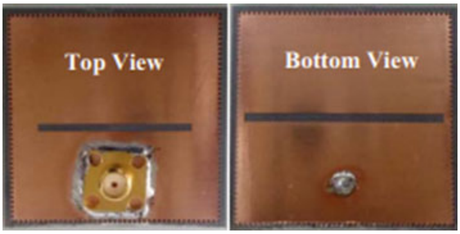


Fig. 7 Top and bottom views

substrate as shown in Fig. 7 with dielectric constant of 2.5. The radiating frequency of the antenna is 3.45 GHz. Slots on either side of an antenna are employed as radiating components. By etching an additional slot, a new resonant mode is harvested at the bottom plane. The range of frequency of the antenna can be considerably expanded by integrating these three modes into a single radiating band. The obtained peak gain for single band is 4.9 dBi and fractional bandwidth is 11.9%.

- The standard cavity-backed with slot SIW antenna designed for single frequency (Ali et al. 2018). The antenna is using microstrip line feed technique. The dimensions of the design are 37.8 × 39.1 × 0.76 mm^3. It is made up of Taconic TLX-9 as substrate as represented in Fig. 8 with dielectric constant of 2.5. The radiating frequency of the antenna is 4 GHz. The reported structure consists of two cavities stacked one on top of the other and coupled through a slot, with a support of slot antenna supporting them. The obtained peak gain for single band is 4.42 dBi and fractional bandwidth is 3.75%.
- A miniaturized Substrate-Integrated Waveguide (SIW)-based antenna is designed for Wireless Body Area Network (WBAN) applications which is used at single frequency (Chaturvedi and Raghavan 2018). The antenna is using microstrip line feed technique. The dimensions of the design are 47 × 41 × 1.57 mm^3. It is made up of RT/Duroid 5880 as substrate as represented in Fig. 9 with dielectric constant of 2.2. The radiating frequency of the antenna is 5.8 GHz. The recommended structure is formed by cutting a circular patch into four quarters along the magnetic walls and the quarter-mode concept is introduced. The obtained peak gain for single band is 6.25 dBi and fractional bandwidth is 6.2%.

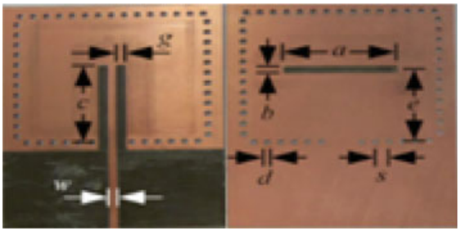


Fig. 8 Top and bottom views

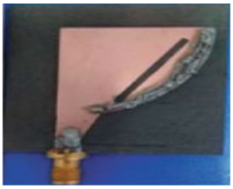


Fig. 9 Top view

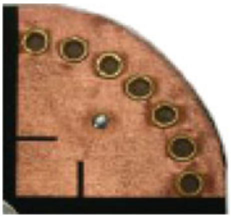


Fig. 10 Top view

- The projected miniaturized wearable antenna is designed specifically for off-body communication with single band (Agneessens et al. 2015). The antenna using coaxial or probe feed technique is made as substrate as represented in Fig. 10 with dielectric constant of 1.495 and thickness of 3.7 mm. The radiating frequency of the antenna is 2.4 GHz. The design is perfectly suitable for off-body communication in wearable electronics and smart textile applications because it combines small dimensions with good Body-Antenna Isolation. The obtained peak gain for single band is 4.2 dBi and fractional bandwidth is 5.1%.
- Two low-profile hybrid antennas are designed for single frequency (Dashti and Neshati 2014). The antenna is using microstrip line feed technique. The dimensions of the design are 23.2 × 21.6 × 0.787 mm^3. It is made up of Rogers RT/Duroid 5880 as substrate as represented in Fig. 11 with a dielectric constant of 2.2. The radiating frequency of the antenna is 7.8 GHz. One of the structures produces a circularly polarized (CP) radiated field and the other generates a linearly polarized (LP) wave. According to the simulation results, installing the patch increases antenna gain. The obtained peak gain for single band is 7.5 dBi and fractional bandwidth is 10%.

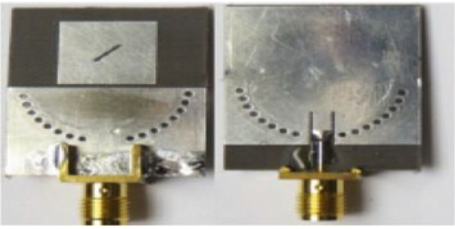


Fig. 11 Top and bottom views

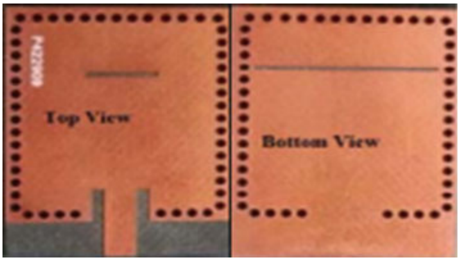


Fig. 12 Top and bottom views

B. ***Number of Frequency Bands: Dual***

- A design of a low-profile cavity-backed antenna comprises bilateral slots for harvesting dual band (Lokeshwar et al. 2020). The dimensions of the design are 29.7 × 22 × 0.8 mm^3 which is made up of RT/Duroid 5880 as substrate as represented in Fig. 12 with dielectric constant of 2.2. The radiating frequencies of the antenna are 9.85 and 14 GHz. The cavity's TE_{210} mode excites the long transversal slot at the ground plane which results in one hybrid mode resonance at 9.85 GHz. The TE_{310} mode is perturbed when the short transverse slot cut is included into the top plane of the cavity which causes the generation of an extra hybrid mode resonance at 14 GHz. The obtained peak gains for dual band are 6.62 and 6.44 dBi and fractional bandwidths are 5.48% and 3.15% for band 1 and band 2, respectively.
- Two parallel rectangular slots are etched on a SIW cavity and are designed for dual-band frequency (Niu et al. 2018). The antenna is using coaxial or probe feed technique. The dimensions of the design are 80 × 39.6 × 3 mm^3 which is made up of F4B-2 as substrate as represented in Fig. 13 with dielectric constant of 2.5. The radiating frequencies of the antenna are 3.28 and 3.77 GHz. The projected design is for the improvement of the stop band properties. The obtained gains for dual band are 5.1 and 5.2 dBi for band 1 and band 2, respectively.
- A wideband SIW-based antenna with multi-resonance slots is presented (Kumar and Raghavan 2017). The antenna is using microstrip line feed technique. The dimensions of the design are 22.85 × 24.3 × 1.57 mm^3 which is made up of RT/Duroid 5880 as substrate as represented in Fig. 14 with dielectric constant of 2.2. The radiating frequencies of the antenna are 13 and

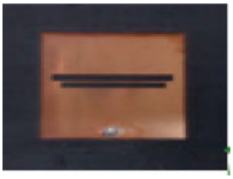


Fig. 13 Top view

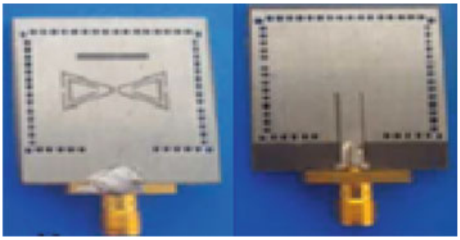


Fig. 14 Top and bottom views

14 GHz. On the cavity's backside, multi-resonant slots are strategically positioned to generate hybrid modes. The obtained peak gains for dual band are given by 6.0 and 8.03 dBi and fractional bandwidths are 11.8% and 11.0% for band 1 and band 2, respectively.

- Dual-band low-profile aperture antenna with high gain using SIW-corrugated structure is reported in Honari et al. (2016). The antenna is using aperture coupling feed technique. The recommended antenna is made up of Rogers 5880 as layer 1 substrate with dielectric constant of 2.2 and thickness of 0.508. Rogers TMM4 as layer 2 substrate is represented in Fig. 15 with dielectric constant of 4.5 and thickness of 3.175. The radiating frequencies of the antenna are 10 and 12.8 GHz. Instead of using a waveguide feeder, the suggested construction uses three simple printed circuit boards to implement a resonant transmission line and particular SIW structures. The obtained gains for dual band are 12.5 and 11.3 dBi and fractional bandwidths are 2.7% and 1.9% for band 1 and band 2, respectively.
- A wideband half-mode SIW cavity-backed slot antenna is designed for dual-band frequency (Caytan et al. 2015). The antenna is using microstrip line feed technique. The dimensions of the design are 42.5 × 31.9 × 3 mm^3. The reported antenna is made of cork substrate which is represented in Fig. 15 with dielectric constant of 1.17. The radiating frequencies of the antenna are 5.26 GHz and 5.85 GHz. A 5 GHz Wi-Fi covering wideband half-mode SIW cavity-backed slot antenna is mounted on a cork substrate. Additionally, the antenna offers consistent performance in various environments. The peak gain is 4.3 dBi and the maximum fractional bandwidth is 23.7%.

Fig. 15 Top view

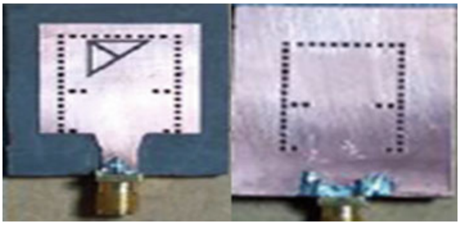


Fig. 16 Top and bottom views

- Modified triangular-ring-slot SIW-based cavity-backed antenna is used to design dual-frequency response projected in Kumar and Raghavan (2018a). The antenna is using tapered microstrip line feed technique. The dimensions of the design are 15.4 × 32.7 × 1.57 mm^3. The recommended antenna is made up of RT/Duroid 5880 substrate as represented in Fig. 16 with dielectric constant of 2.2. The radiating frequencies of the antenna are 14.6 and 15.7 GHz. Due to the significant loading impact in the SIW cavity resonator, the typical triangular ring slot successfully harvested two hybrid modes close to the TE$_{110}$ and TE$_{120}$ cavity modes. The obtained peak gains for dual band are 4.7 and 4.2 dBi, and the maximum fractional bandwidth is 13.53%.

C. ***Number of Frequency Bands: Triple***

- A millimeter-wave triple-band SIW antenna with dual-sense Circular Polarization (CP) etched on a single substrate is recommended in Chen et al. (2020). The antenna is using tapered microstrip line feed technique. The dimensions of the design are 30 × 15 × 0.254 mm^3. The suggested antenna is made up of RT/Duroid 5880 substrate as represented in Fig. 17 with dielectric constant of 2.2. The radiating frequencies of the antenna are 28, 33, and 38 GHz. The circular slot for lower band radiation is filled with a patch in the shape of a circle. The obtained peak gains for triple band are 6.89, 8.01, and 9.41 dBi and fractional bandwidths are 3.7%, 2.3%, and 3.1% for band 1, band 2, and band 3, respectively.
- A wideband Half-Mode Substrate-Integrated Waveguide (HMSIW)-based cavity-backed slot antenna is suggested in Chaturvedi et al. (2019). The antenna is using tapered microstrip line feed technique. The dimensions of the design are 56 × 32 × 1.57 mm^3. The recommended antenna is made up of RT/Duroid 5880 substrate as represented in Fig. 18 with dielectric constant of 2.2.

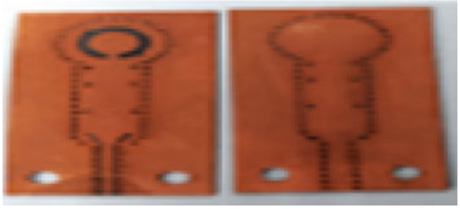


Fig. 17 Top and bottom views

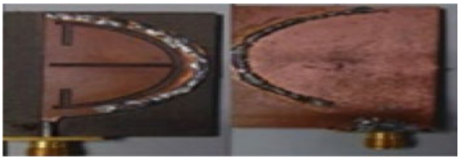


Fig. 18 Top and bottom views

The radiating frequencies of the antenna are 5.2, 5.5, and 5.8 GHz for band 1, band 2, and band 3, respectively. On the cavity's top plane, an epsilon-shaped radiating slot is etched to produce a large −10 dB level impedance bandwidth of 13.2%. (5.092–5.817 GHz). The suggested antenna has a unidirectional radiation pattern, is low-profile, lightweight, and maintains planar integrity, all of which make it an excellent option for wireless fidelity applications. The obtained peak gain is 6.7 dBi and the maximum fractional bandwidth is 13.2%.

- A triple band cavity-backed slot antenna with SIW resonator as cavity is represented in Mohan et al. (2017). The antenna is using microstrip line feed technique. The dimensions of the design are 40 × 57.6 × 1.524 mm^3. The projected antenna is made up of Rogers R04003C substrate as represented in Fig. 19 with dielectric constant of 3.38. The radiating frequencies of the antenna are 3.3, 4, and 4.9 GHz. The TE_{210} and TE_{310} modes are combined to create the first band (3.3 GHz). The higher order modes of the TE130 and TE_{230} are the foundations for the second (4 GHz) and third (4.9 GHz) bands, respectively. The obtained peak gains for triple band 5.81, 4, and 2.43 dBi and fractional bandwidths are 1.49%, 0.72%, and 0.56% for band 1, band 2, and band 3, respectively.
- A compact SIW-based wearable triband leather antenna is designed for optimal on-body performance in Mandal and Parui (2015). The antenna is using microstrip line feed technique. The dimensions of the design are 80 × 80 × 2 mm^3. The projected antenna is made up of leather substrate as represented in Fig. 19 with dielectric constant of 2.72. The radiating frequencies of the antenna are 2.4, 3.51, and 4.69. Brass eyelets are used to form a substrate integrated waveguide cavity with a ground plane and conducting and nonconductive leather materials are combined. The obtained peak gains for triple band are 1.1, 0.9, and 2 dBi and fractional bandwidths are 11%, 7%, and 5% for band 1, band 2, and band 3, respectively.

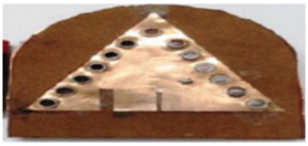


Fig. 19 Top view

D. ***Number of Frequency Bands: Quad***

- Low-profile cavity-backed cross-shaped slot antenna using SIW technology is used for harvesting quad-band response in Lokeshwar et al. (2022). The antenna uses microstrip line feed technique. The dimensions of the design are 29.7 × 22 × 0.8 mm^3. The projected antenna is made up of RT/Duroid 5880 substrate as represented in Fig. 19 with dielectric constant of 2.2. The radiating frequencies of the antenna are 8.6, 10.2, 13.2, and 14.8 GHz. On the back of the dielectric laminate, a crossing slot is etched and three balanced shorting vias (metal vias) are added. The obtained peak gains for quad band are 6.30, 5.85, 7.10, and 6.95 dBi and fractional bandwidths are 3.65%, 2.4%, 1.13%, and 1.68% for band 1, band 2, band 3, and band 4, respectively.
- Substrate-Integrated Waveguide (SIW), Cavity-Backed Slot (CBS) antenna are recommended for bandwidth enhancement in Wu et al. (2019). The antenna is using microstrip line feed technique. The dimensions of the patch are 23.2 × 20.15 × 1 mm^3. The projected antenna is made up of substrate represented with dielectric constant of 2.2. The radiating frequencies of the antenna are 8.9, 9.3, 10.2, and 10.7 GHz. Quad-resonance SIW CBS antenna is projected using a plus-shaped slot and loading unbalanced shorting metal vias. The peak gain is 4.93 dBi and the maximum fractional bandwidth is 20%.
- A quad and triple resonance SIW cavity-backed slot antenna using shorting vias are presented in Shi et al. (2017). The antenna is using microstrip line feed technique. The dimensions of the design are 23.2 × 18.8 × 1 mm^3. The reported antenna is made up of substrate as represented with dielectric constant of 2.2. The radiating frequencies of the antenna are 9.32, 9.97, 10.70, and 11.12 GHz. One more cavity mode is intended to be excited by the quad resonance antenna (TE$_{310}$ mode). The obtained peak gain is 7.3 dBi and the maximum fractional bandwidth is 17.5%.

Compared with the existing SIW antennas with their design and performance parameters based on the radiating frequency bands.

From Table 2, the following observations can be made:

- In Dashti and Neshati (2014) the dimensions of the antenna are less, maximum of radiating frequency at 7.8 GHz and shows the peak gain of 7.5 dBi.
- Niu and Tan (2019) gives high fractional bandwidth 11.9% and used for applications of 5G and others for 4G communications.

From Table 3, the following observations can be made:

- In Kumar and Raghavan (2018a), the dimensions of the antenna are less and maximum radiating frequencies are 14.6 and 15.7 GHz.
- Honari et al. (2016) show the peak gains of 12.5 and 11.3 dBi.
- Kumar and Raghavan (2017) give high fractional bandwidths of 11.8% and 11.0%.

Table 2 Comparative analysis of SIW antennas with single band

References	Dimensions ($L \times W \times h$)	Dielectric constant (ε_r)	Type of feed	Radiating freq. (GHz)	Peak gain (dBi)	Fractional bandwidth (%)	Applications
Chaturvedi (2020)	38 × 30 × 1.52	3.55	Microstrip line	5.8	7.15	5.2	ISM band
Niu and Tan (2019)	41.8 × 41.8 × 1.57	2.5	Coaxial	3.45	4.3	11.9	5G
Ali et al. (2018)	37.8 × 39.1 × 0.76	2.5	Microstrip line	4	4.42	3.75	Wireless
Chaturvedi and Raghavan (2018)	47 × 41 × 1.57	2.2	Microstrip line	5.8	6.25	6.2	Wireless Body Area Network
Agneessens et al. (2015)	*x* × 3.7	1.49	Coaxial	2.4	4.2	5.1	On and Off body
Dashti and Neshati (2014)	23.2 × 21.6 × 0.78	2.2	Microstrip line	7.8	7.5	10	C band

*Not mentioned

Table 3 Comparative analysis of SIW antennas with dual band

References	Dimensions ($L \times W \times h$)	Dielectric constant (ε_r)	Type of feed	Radiating freq. (GHz)	Peak gain (dBi)	Fractional bandwidth (%)	Applications
Lokeshwar et al. (2020)	29.7 × 22 × 0.8	2.2	Microstrip line	9.85 and 14	6.62 and 6.44	5.48 and 3.15	X/Ku band
Niu et al. (2018)	80 × 39.6 × 3	2.5	Coaxial	3.28 and 3.77	5.1 and 5.2	NM	S band
Kumar and Raghavan (2017)	22.85 × 24.3 × 1.7	2.2	Microstrip line	13 and 14	6.0 and 8.03	11.8 and 11.0	Ku band
Honari et al. (2016)	*x* × 0.50 *x* × 3.17	2.2 4.4	Aperture coupling	10 and 12.8	12.5 and 11.3	2.7 and 1.9	X/Ku band
Caytan et al. (2015)	42.5 × 31.9x3	1.17	Microstrip line	5.26 & 5.85	4.3	23.7	IoT
Kumar and Raghavan (2018a)	15.4 × 32.7 × 1.57	2.2	Tapered microstrip	14.6 and 15.7	4.7 and 4.2	13.57	Ku band

*Not mentioned

Table 4 Comparative analysis of SIW antennas with triple band

References	Dimensions ($L \times W \times h$)	Dielectric constant (ε_r)	Type of feed	Radiating freq. (GHz)	Peak gain (dBi)	Fractional bandwidth (%)	Applications
Chen et al. (2020)	30 × 15 × 0.25	2.2	Microstrip line	28, 33, and 38	6.89, 8.01, and 9.41	3.7, 2.3, and 3.1	mm Wave
Chaturvedi et al. (2019)	56 × 32 × 1.57	2.2	Tapered microstrip	5.2, 5.5, and 5.8	6.7	13.2	Wi-Fi
Mohan et al. (2017)	40 × 57.6 × 1.52	3.38	Microstrip line	3.3, 4, and 4.9	5.81, 4, and 2.43	1.49, 0.72, and 0.56	S/C band
Mandal and Parui (2015)	80 × 80 × 2	2.72	Coaxial	2.4, 3.51, and 4.69	1.1, 0.9, and 2	11, 7, and 5	ISM/ Wi-Max

- Lokeshwar et al. (2020) and Honari et al. (2016) are used for X/Ku band applications.
- Kumar and Raghavan (2017) and Kumar and Raghavan (2018a) are used for Ku band application.
- Niu et al. (2018) and Caytan et al. (2015) are used for S band and Internet of Things (IoT) applications.

From Table 4, the following observations can be made:

- In Chen et al. (2020) the dimensions of the antenna are less, maximum of radiating frequencies at 28, 33 and 38 GHz shows the peak gains of 6.89, 8.01 and 9.41 dBi and used for applications of millimeter wave (mm Wave) also others for wireless applications, respectively.
- Chaturvedi et al. (2019) gives a high fractional bandwidth of 13.2%.

From Table 5, the following observations can be made:

- In Shi et al. (2017), the dimensions of the antenna are less maximum of radiating frequencies at 9.32, 9.97, 10.7, and 11.12 GHz and show the peak gain of 7.3 dBi.
- Wu et al. (2019) gives high fractional bandwidth which is 20%.
- Lokeshwar et al. (2022) used for applications of X/Ku band and others for X band applications.

Table 5 Comparative analysis of SIW antennas with quad band

References	Dimensions ($L \times W \times h$)	Dielectric constant (ε_r)	Type of feed	Radiating freq. (GHz)	Peak gain (dBi)	Fractional bandwidth (%)	Applications
Lokeshwar et al. (2022)	29.7 × 22 ×0.8	2.2	Microstrip line	8.6, 10.2, 13.2, and 14.8	6.30, 5.85, 7.10, and 6.95	3.65, 2.4, 1.13, and 1.68	X/Ku band
Wu et al. (2019)	23.2 × 20.15 ×1	2.2	Microstrip line	8.9, 9.3, 10.2, and 10.7	4.93	20	X band
Shi et al. (2017)	23.2 × 18.8 × 1	2.2	Microstrip line	9.32, 9.97, 10.70, and 11.12	7.3	17.5	X band

5 Conclusion

In this article, SIW antennas are summarized based on a variety of radiating frequency bands, including single, dual, triple, and quad. Additionally, recent developments in modeling, design, technological implementation, and performance enhancement, such as increased bandwidth and gain, are discussed. This paper also describes the type of substrate with dielectric constant values, various feeding strategies, the size of the antenna and implementation techniques such as slots and multiple slots for each recommended antenna with respect to the frequency band (rectangular, ring or circular, triangular, bow tie, and dumbbell shaped) metal vias to the ground plane and top layer patch. The various methods are defined to enhance the bandwidth and the gain of the antenna. The paper broadly analyzed the radiating frequency, gain, and fractional bandwidth parameters of projected antennas in detail. Based on the survey, SIW antenna is having amicable characteristics, e.g., low profile, lightweight, less expensive, simple fabrication, low volume, directional radiation pattern, good power handling capacity and significantly used for millimeter wave, 5G systems, Internet of Things (IoT), radar, satellite, and other wide band applications. This survey can also be extended to Multiple Input Multiple Output (MIMO)-based SIW antennas.

References

Agneessens S et al (2015) Wearable, small, and robust: the circular quarter-mode textile antenna. Antennas Wirel Propag Lett 14:1482–1485

Ali HA et al (2018) Increasing the bandwidth of cavity-backed SIW antennas by using stacked cavities. Int J Microw Wireless Technol 10(8):942–947

Bozzi M et al (2008) Modeling of conductor, dielectric, and radiation losses in substrate integrated waveguide by the boundary integral–resonant mode expansion method. IEEE Trans Microw Theory Technol 56:3153–3161

Cassivi Y et al (2002) Dispersion characteristics of substrate integrated rectangular waveguide. IEEE Microw Wireless Compon Lett 12:333–335

Caytan O et al (2015) Half-mode substrate-integrated-waveguide cavity-backed slot antenna on cork substrate. IEEE Antennas Wirel Propag Lett 15:162–165

Chaturvedi D (2020) SIW cavity-backed 24° inclined-slots antenna for ISM band application. Int J RF Microwave Comput Aided Eng 30(5):e22160

Chaturvedi D, Raghavan S (2018) Circular quarter-mode SIW antenna for WBAN application. IETE J Res 64(4):482–488

Chaturvedi D, Kumar A, Raghavan S (2019) Wideband HMSIW-based slotted antenna for wireless fidelity application. IET Microwaves Antennas Propag 13(2):258–262

Chen H et al (2020) A millimeter-wave triple-band SIW antenna with dual-sense circular polarization. IEEE Trans Antennas Propag 68(12):8162–8167

Daniels RC et al (2007) 60 GHz wireless communications: emerging requirements and design recommendations. IEEE Vehicle Technol Mag 2:41–50

Dashti H, Neshati MH (2014) Development of low-profile patch and semi-circular SIW cavity hybrid antennas. IEEE Trans Antennas Propag 62(9):4481–4488

Deslandes D (2010) Design equations for tapered microstrip-to-substrate integrated waveguide transitions. In: IEEE MTT-S International Microwave Symposium. Digest, Anaheim, CA, USA, pp 704–707

Ding Y et al (2007) Substrate integrated waveguide-to-microstrip transition in multilayer substrate. IEEE Transactions Microwave Theory Technology, pp 2839–2844

Fleming WJ (2008) New automotive sensors a review. IEEE Sensor J 8(11):1900–1921

Gupta KC et al (2000) Analysis and design of integrated circuit antenna modules. Wiley, New York

Honari MM et al (2016) A dual-band low-profile aperture antenna with substrate-integrated waveguide grooves. IEEE Trans Antennas Propag 64(4):1561–1566

Kumar A, Raghavan S (2017) Wideband slotted substrate integrated waveguide cavity-backed antenna for Ku-band application. Microw Opt Technol Lett 59(7):1613–1619

Kumar A, Raghavan S (2018a) Broadband SIW cavity-backed triangular-ring-slotted antenna for Ku-band applications. AEU-Int J Electron Commun 87:60–64

Lacik J and Mikulasek T (2011) Substrate integrated waveguide rectangular ring slot antenna. In: Proceedings of the 13th International Conference on Electromagnetics in Advanced Applications (ICEAA '11), pp 1164–1167

Lokeshwar B et al (2020) Dual-band low profile SIW cavity-backed antenna by using bilateral slots. Prog Electromagn Res C 100:263–273

Lokeshwar B et al (2022) Quad band substrate integrated waveguide cavity backed slot antenna using balanced shorting vias. Wireless Pers Commun 125(3):2565–2579

Mandal B, Parui SK (2015) Wearable tri-band SIW based antenna on leather substrate. Electron Lett 51(20):1563–1564

Mizuno K et al (2007) Millimeter-wave imaging technologies and their applications. In IEEE International Vacuum Electronics Conference(IVEC '07), 15–17 May 2007

Mohan MP et al (2017) Triple band SIW cavity backed slot antenna. In: IEEE Asia Pacific Microwave Conference (APMC).

Niu BJ, Tan JH (2019) Bandwidth enhancement of low-profile SIW cavity antenna with bilateral slots. Prog Electromagn Res Lett 82:25–32

Niu BJ et al (2018) SIW cavity-backed dual-band antenna with good stopband characteristics. Electron Lett 54(22):1259–1260

Shi Y, Liu J, Long Y (2017) Wideband triple-and quad-resonance substrate integrated waveguide cavity-backed slot antennas with shorting vias. IEEE Trans Antennas Propag 65(11):5768–5775

Wu Q et al (2019) Broadband planar SIW cavity-backed slot antennas aided by unbalanced shorting vias. IEEE Antennas Wirel Propag Lett 18(2):363–367

Wu KE et al (2021) Substrate integrated transmission lines: Review and applications. IEEE J Microw 1(1):345–363

Xu F et al (2005) Guided-wave and leakage characteristics of substrate integrated waveguide. IEEE Trans Microw Theory Technol 53(1):66–73

Daily Platelet Count Prediction in Treating Dengue Patients Using Deep Learning Algorithm

S. Ruban, Mohammed Moosa Jabeer, and Sanjeev Rai

1 Introduction

Health care is undergoing a revolutionary change, thanks to artificial intelligence. AI applications are having a huge impact on Medical Field. It is all about creating tools or programs that can help people by mimicking different facets and levels of human intellect. According to a recent assessment (World Health Organization 2020), AI can revolutionize health care around the world. They do insist, though, on using strong ethical principles to safeguard and shape how AI-based solutions are designed, developed, and implemented. The challenges that arise during the creation of AI-enabled technology are listed in another recent paper by Modu et al. (2017). Can the machines think? was one of Alan Turing's questions that defined the goals for AI research many years ago. In Centers for Disease Control and Prevention (2020), the application of AI in health care has received a great deal of attention recently. Solutions that forecast procedure outcomes and help predict crises like respiratory arrests and lung malignancies have been made possible by recent advancements in AI in health care, allowing healthcare institutions to provide patients with better care. The field of artificial intelligence is wide. AI also has the potential of identifying trends that lead to epidemics. To investigate and identify the potential cause of the epidemic, deep learning models can be created over the clinical notes kept in medical facilities, which can serve as the foundation for the creation of noninvasive techniques or prediction models.

S. Ruban (✉)
Department of Software Technology, St Aloysius College, Mangalore, India
e-mail: ruban@staloysius.ac.in

M. M. Jabeer
Department of MCA, St Aloysius College, Mangalore, India

S. Rai
Father Muller Medical College, Mangalore, India

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_38

According to data kept by the World Health Organization, dengue is becoming more common and spreading to more areas. Research indicates that fifty million illnesses occur annually (Hommelsheim et al. 2014; Hawkes et al. 2009; Weng et al. 2020). An earlier detection of illness can significantly lessen these figures. Saving money and time on diagnostic tests are two additional advantages (Javier Rodriquez-vera et al. 2002; Lee et al. 2016). Another piece of research (Gers et al. 2000) in this field contends that using machine learning techniques for medical diagnosis improves outcomes. There is now a need for more preventative care and awareness about dengue due to the sharp rise of cases in midst of unseasonal rains. While preventing dengue fever is still the best course of action, it's still critical to focus on prompt recovery and minimize consequences, one of which is a sharp decline in platelet count.

One dreaded complication of dengue patients is platelet loss, which can hinder recovery. Many people who are fighting an infection also need blood transfusions, and how stable one's platelet and blood count levels are will determine how healthy they are (Sangwon et al. 2018). While more patients have mild symptoms, few develop a more severe and occasionally fatal condition, which leads to dengue shock syndrome (DSS) (Wang et al. 2019; Bhatt et al. 2017). Few related symptoms are thrombocytopenia, that can cause severe bleeding, and organ involvement, that can develop to catastrophic organ failure (Shivakumar et al. 2015). Several teams have made an effort to pinpoint risk indicators and create DSS prediction models (Symptoms of Malaria 2021; ICD code of Malaria 2021; Smith 2007; Sidey-Gibbons and Sidey-Gibbons 2019; XGBOOST machine learning algorithm 2021; Gradient Boost machine learning algorithm 2021; Random Forest machine learning algorithm 2021; Lohumi et al. 2020).

The following part discusses the many works that have already been done in this field: Sect. 3 provides the approach that we used, Sect. 4 lists out the findings, and Sect. 5 the conclusion.

2 Existing Work

A few of the early publications that used machine learning algorithms to detect dengue refer to the use of the deep learning algorithms (Ruban and Rai 2021). SVM and artificial neural network (ANN) utilization for dengue detection are also highlighted in a recent study by researchers in Paraguay (Ruban et al. 2021). ML has been used to choose the best course of action (Ensemble methods 2021), identify people who have a chance of specific disease, recommend medication options, and even foresee an illness. In India, vector-borne disease causes over 7 lakh fatalities yearly, making it a significant public health issue. The state of Karnataka's favorable coastline location and irrigation encourage mosquito breeding, which results in a high dengue transmission rate (Arali et al. 2019). Understanding disease epidemics can be aided by AI and big data technology (Wong et al. 2019). By developing predictive models and assisting in the correlation with other aspects like the climate, rainfall, and other variables, it can also be employed in the medical industry to enhance the clinical work's professionalism and effectiveness (Guo and Li 2018). The Thiruvananthapuram district of Kerala hosted one of the studies on dengue (Valson and Soman 2017).

In Malaysia, in the state of Penang, this dengue surveillance equipment was put into use (Sundram et al. 2019). This study claims an accuracy rate of 81% and claims to have predicted 37 outbreaks in the area one month in advance. The assertion that machine learning can be used as a tool for live monitoring has been supported by this study. Another experimental investigation (Guo et al. 2017) was carried out using features gathered at China. Authors gathered climatic data from 2011 to 2014 and created a model to forecast. Although researchers looked at using many machine learning algorithms for their research, they ultimately chose to employ the SVR method because of its superior performance. In Manila, a similar experimental investigation was conducted (Carvajal et al. 2018). Various meteorological conditions were all recorded by the researchers. The study employed a number of modeling methods, including General Additive Modeling, Gradient Boosting, and Random Forest. They found that Random Forest performed well for the provided data set.

In this experimental work, we examined information stored in the clinical history of patients, aged 8–90, who were admitted to the Father Muller Medical College Hospital in Mangalore City between 2014 and 2018, with clinically suspected dengue. This prediction model was created using artificial neural network (ANN). With only a few more variables plus the platelet counts from the previous two days, this model can accurately predict the platelet count for the following day to within about 90% of the actual amount.

3 Methodology

3.1 Data Collection

Within the state of Karnataka is the seaside city of Mangalore. One of the biggest vector-borne diseases that are spread over the world by mosquitoes is dengue. On the basis of the clinical notes in the hospital records, the records of the dengue patients treated in the hospital from 2015 to 2018 were taken and digitalized for this research study with approval from the Medical College's Ethical Committee. A person, having dengue, can also suffer from symptoms like the following: eye pain, fever, platelet loss, cerebral ache. Dengue symptoms often last 2–7 days. The majority of people will recover after approximately a week. ICD codes are used to arrange the clinical records (ICD code for Dengue 2020). We obtained the records of patients whose platelet counts were recorded for 7 consecutive days in order to conduct this study on platelet count. 721 patients' records from among those who had dengue treatment at the Hospital between the years 2014 and 2018 were considered

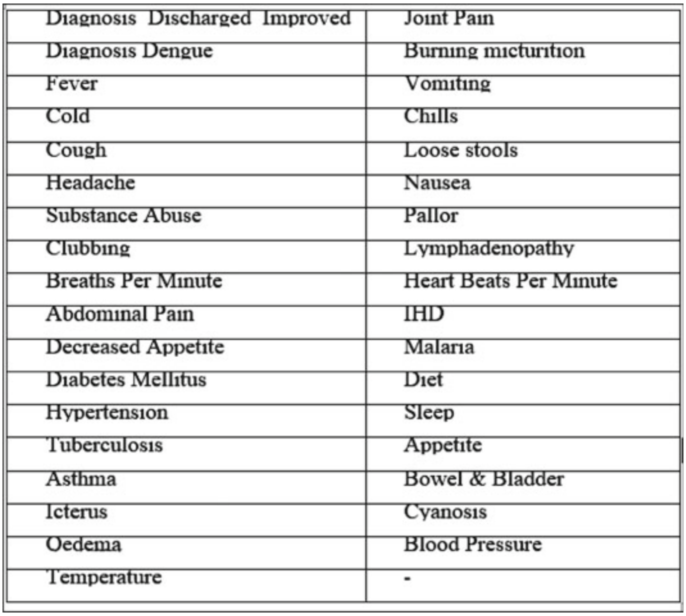
Table 1 Summary of patients whose platelet counts were used for the experimental study

Predictor	Number	Summary statistic	
		Category	Count
Age	721	≤ 2	259
		> 2 and ≤ 4	252
		> 4 and ≤ 7	189
		> 7	21
Gender	721	Male	441
		Female	280
Headache	721	Yes	427
		No	294
Chills and rigor	721	Yes	602
		No	119
Vomit	721	Yes	301
		No	420
Cough	721	Yes	63
		No	658
Urine complaint	721	Yes	28
		No	693
Joint pain	721	Yes	350
		No	371
Abdominal discomfort	721	Yes	42
		No	679
Temperature	721	≤ 100	462
		> 100	259

for this experimental study. They had platelet count values that had been kept for seven days following their hospital admission. Table 1 provides some insight.

3.2 Data Pre-processing

The clinical notes' data were unedited, with the majority of the entries being written by hand. The majority of the medical history were handwritten sections put down by many healthcare professionals in charge of treatment. The typed portion of the medical prescription is the discharge report. Considering the quality of data, for the machine learning endeavor, null entries, discrepancies, typing mistake, and semantic problems that were present in the raw data were clarified and corrected. Even yet, this phase doesn't provide any insightful information. It aids in determining the proper underlying features. The features were extracted from the medical records images.



Diagnosis Discharged Improved	Joint Pain
Diagnosis Dengue	Burning micturition
Fever	Vomiting
Cold	Chills
Cough	Loose stools
Headache	Nausea
Substance Abuse	Pallor
Clubbing	Lymphadenopathy
Breaths Per Minute	Heart Beats Per Minute
Abdominal Pain	IHD
Decreased Appetite	Malaria
Diabetes Mellitus	Diet
Hypertension	Sleep
Tuberculosis	Appetite
Asthma	Bowel & Bladder
Icterus	Cyanosis
Oedema	Blood Pressure
Temperature	-

Fig. 1 Data Dictionary built in this experimental study

The images that had less clarity had a moderate impact on the accuracy of the data that was recovered.

3.3 Data Processing

From the day the dengue patient was admitted to the hospital until the day he or she was discharged, every image including raw data of the patient was taken. Python-Tesseract was employed, to detect and read text that is contained in images (Fig. 1). All of the images were processed using the Python program, which turned them into textual contents. The clinical records were checked for the symptom list. Based on the symptoms, a Data Dictionary was created. The full data processing technique is described in detail in the study work that the authors have previously published (Ruban et al. 2021).

3.4 Building a Model

Deep learning algorithms called ANNs are modeled after how the human brain works. They are systems with the capacity to alter their internal organization in response to a

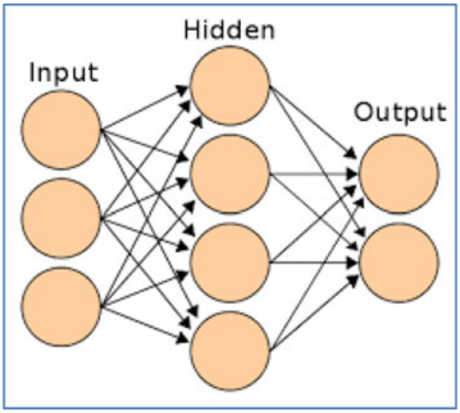


Fig. 2 Architecture of the artificial neural network

function objective. Due to their ability to recreate the fuzzy rules governing the ideal solution for problems of this type, they are particularly well suited for dealing with nonlinear difficulties. The nodes are the fundamental components of an ANN. Each node communicates with other nodes or the environment and contains a function f that it uses to convert its own global input into output (Fig. 2).

4 Results and Discussion

After the pre-processing was finished, the real-time data was analyzed. Based on the created Data Dictionary, the analysis was completed. A selection of the insights that were produced is shown in Fig. 3 through Figs. 4, 5, and 6.

The medical records that were considered for this experimental study had platelet counts that ranged from 1000 to 180,000 (mcL), however the majority of patients'

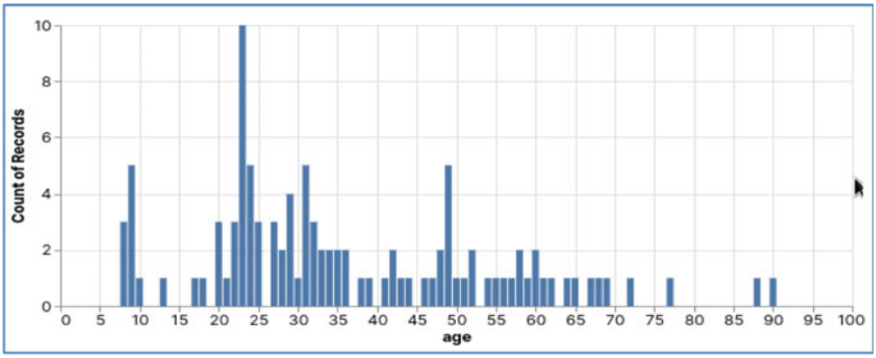


Fig. 3 Patients with their age

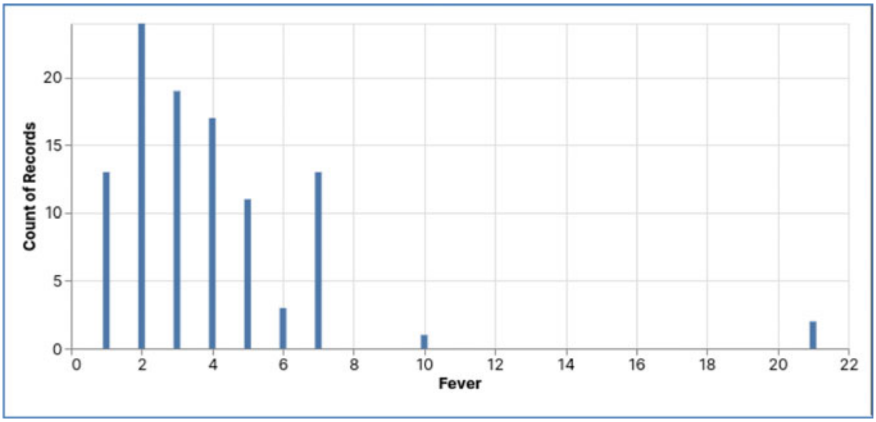


Fig. 4 Patients according to their days of fever

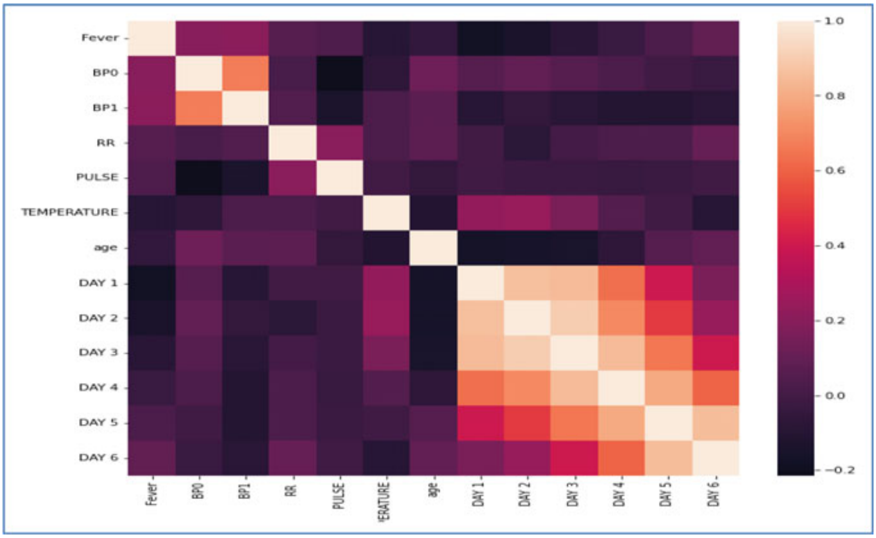


Fig. 5 Confusion matrix generated based on the correlation factors

counts on their first day were in the 100,000–130,000 range. The subsequent platelet count, which was measured over the course of the following six days, revealed a little increase. Adults, having less than 150,000 platelets per microliter, indicate a low platelet count (Fig. 7).

The ANN architecture that was built in this experimental study is shown in Fig. 8.

Few of the predicted results by our artificial neural network (ANN) are shown in Fig. 9.

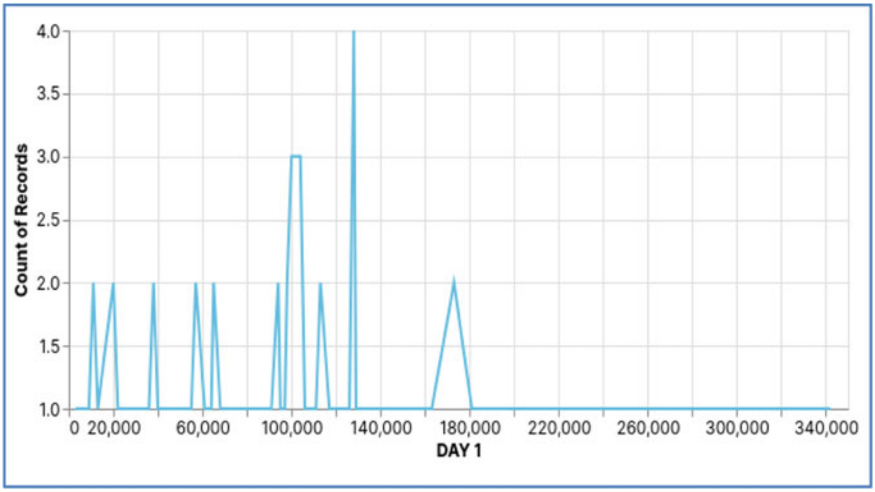


Fig. 6 Platelet count (mcL) range for the patients on Day 1 of admission

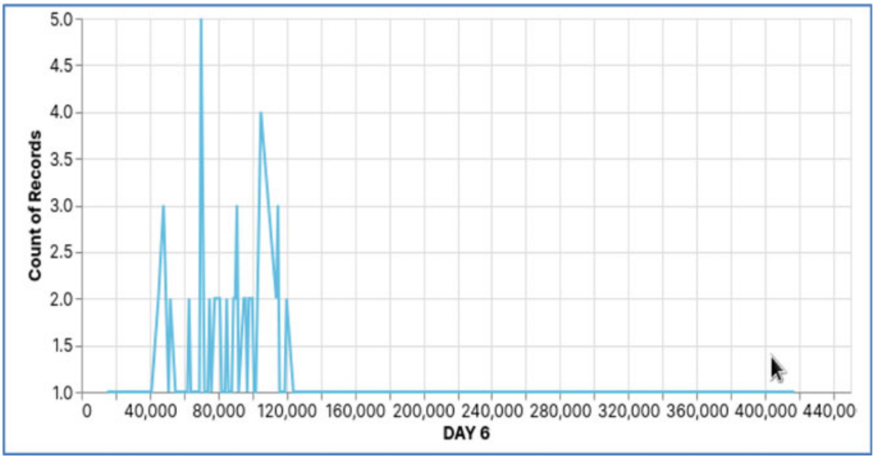
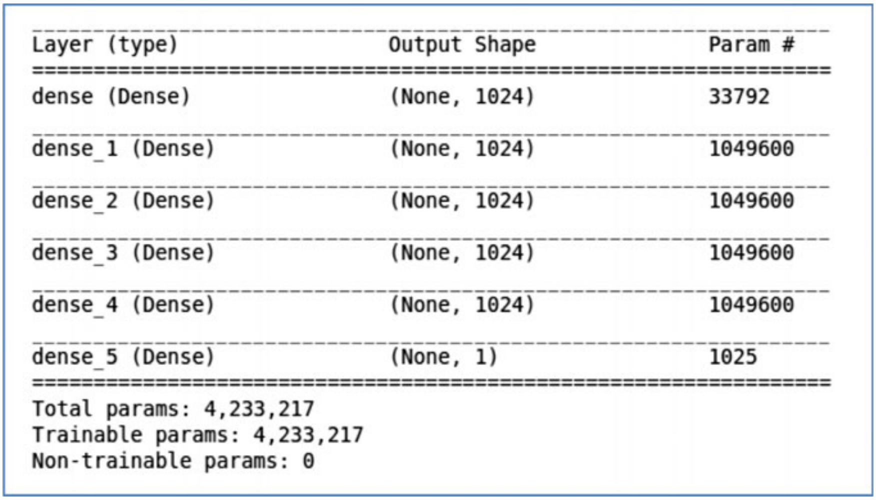


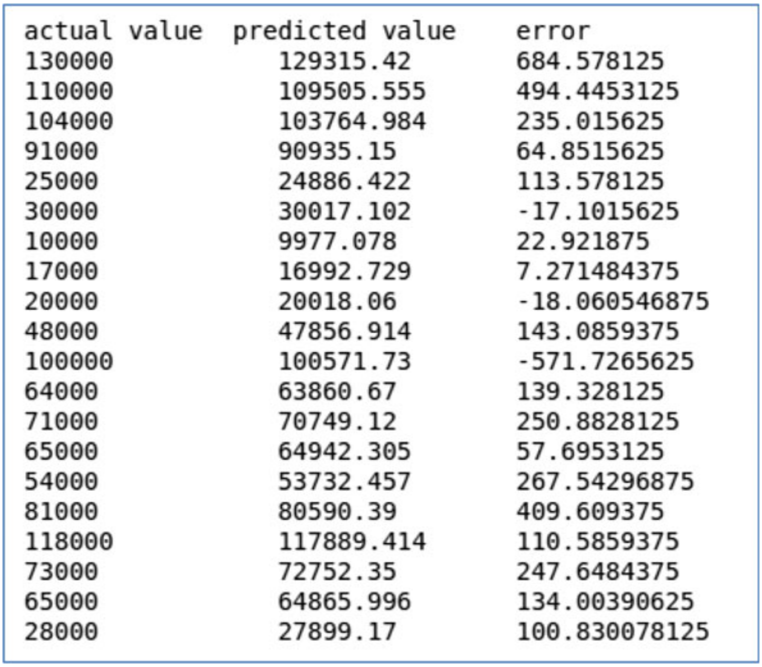
Fig. 7 Platelet count (mcL) range for the patients on Day 6 of admission

Given the two prior platelet count values that were obtained in the previous two days, the model that was created was able to forecast the patient's platelet count. There is, however, an error value that is recorded, which is calculated using Mean Squared Logarithmic Error (MSLE). It is the proportional difference between the expected and actual log-transformed data. The Mean Squared Error (MSE) is MSE = 147,127.45482531292, the Root Mean Squared Error (RMSE) is 1444.9609375, and the Mean Absolute Error is 200.7128855738147, and the error value according to MSLE is 1.08019955579586e−05. Figure 10 elaborates on the error loss.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 1024)              33792
_________________________________________________________________
dense_1 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dense_2 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dense_3 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dense_4 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 1025
=================================================================
Total params: 4,233,217
Trainable params: 4,233,217
Non-trainable params: 0
```

Fig. 8 ANN model architecture



```
actual value  predicted value    error
130000           129315.42       684.578125
110000           109505.555      494.4453125
104000           103764.984      235.015625
91000            90935.15        64.8515625
25000            24886.422       113.578125
30000            30017.102       -17.1015625
10000            9977.078        22.921875
17000            16992.729       7.271484375
20000            20018.06        -18.060546875
48000            47856.914       143.0859375
100000           100571.73       -571.7265625
64000            63860.67        139.328125
71000            70749.12        250.8828125
65000            64942.305       57.6953125
54000            53732.457       267.54296875
81000            80590.39        409.609375
118000           117889.414      110.5859375
73000            72752.35        247.6484375
65000            64865.996       134.00390625
28000            27899.17        100.830078125
```

Fig. 9 Actual values versus predicted value

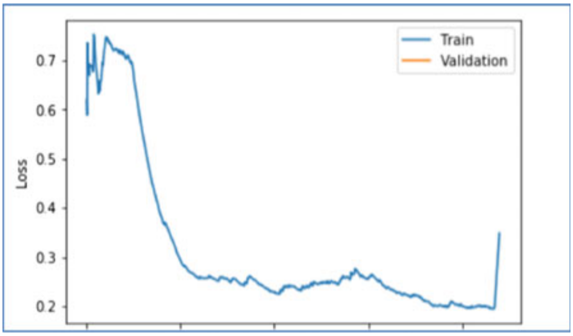


Fig. 10 Error loss using MLSE after the pre-processing was finished, the real-time data was analyzed, based on the created data

5 Conclusion

This study investigates the possibility of an AI-based prediction model using an artificial neural network (ANN), for predicting the platelet count in the patient, given the platelet count values from the previous two days. This model is built, using the real-time data set generated from the medical records of patients treated for dengue at Father Muller's, Mangalore, between 2014 and 2018. The severity of the illness worsens when the platelet count continues to decline on a regular basis. The earlier the platelet count can be predicted, more time can be saved, to prevent the negative effects of dengue, it is beneficial to begin the treatment well in advance. More information from many other hospital settings could boost the system's effectiveness, and real-time clinical validation will enhance its use in a real-time situation.

Acknowledgements Data were received from Father Muller Hospital using a protocol number (FMMCIEC/CCM/149/2019), and the study was carried out in a lab supported by the Karnataka Government's VGST, K-FIST(L2)-545.

References

Arali PK et al (2019) Assessment of national vector borne disease control programme in state of Karnataka. Int J Community Med Public Health 6(2):525–532

Bhatt S, Cameron E, Flaxman SR, Weiss DJ, Smith DL, Gething PW (2017) Improved prediction accuracy for disease risk mapping using Gaussian process stacked generalization. J R Soc Interface 14(134)

Ensemble methods. https://towardsdatascience.com/ensemble-methods-in-machine-learning-what-are-they-and-why-use-them-68ec3f9fef5f. Accessed on 10 July 2021

Carvajal TM, Viacrusis KM, Hernandez LFT, Ho HT, Amalin DM, Watanabe K (2018) Machine learning methods reveal the temporal pattern of dengue incidence using meteorological factors in Metropolitan Manila, Philippines. BMC Infect Dis 18(1):183

Centers for Disease Control and Prevention. CDC_Malaria. Available at http://www.cdc.gov/malaria/about/biology. Accessed on 16 Aug 2020

Cuddeback J. Using big data to find hypertension patients hiding in plain sight. AMGA Analytics

Gers FA, Schmidhuber J, Cummins F (2000) Learning to forget: continual prediction with LSTM. Neural Comput 12(10):2451–2471. https://doi.org/10.1162/089976600300015015

Gradient Boost machine learning algorithm. https://towardsdatascience.com/machine-learning-part-18-boosting-algorithms-gradient-boosting-in-python-ef5ae6965be4. Accessed on 2 July 2021

Guo J, Li B (2018) The application of medical artificial intelligence technology in rural areas of developing countries. Health Equity 2(1)

Guo P, Liu T, Zhang Q et al (2017) Developing a dengue forecast model using machine learning: a case study in China. PLoS Neglected Trop Dis 11(10), Article e0005973

Hawkes M, Katsuva J, Masumbuko C (2009) Use and limitations of malaria rapid diagnostic testing by community health workers in war-torn Democratic Republic of Congo. Malar J 8(1):308. https://doi.org/10.1186/1475-2875-8-308

Hommelsheim CM, Frantzeskakis L, Huang M, Ülker B (2014) PCR amplification of repetitive DNA: a limitation to genome editing technologies and many other applications. Sci Rep 4:5052. https://doi.org/10.1038/srep05052

ICD code for Dengue. https://icd.codes/icd10cm/A90. Accessed on 21 Sept 2020

ICD code of Malaria. https://www.icd10data.com/ICD10CM/Codes/A00-B99/B50-B64/B54-/B54. Accessed on 1 July 2021

Query ID="Q5" Text="Reference 'Cuddeback J. Using big data to find hypertension …' is given in the list but not cited in the text. Please cite this in text or delete from the list." Javier Rodriquez-vera F, Marin Y, Sanchez A, Borrachero C, Pujal E (2002) Illegible handwriting in medical records. J R Soci Med 95:545–546

Lee KY, Chung N, Hwang S (2016) Application of an artificial neural network (ANN) model for predicting mosquito abundances in urban areas. Ecol Inf 172–180

Lohumi P, Garg S, Singh TP, Gopal M (2020) Ensemble learning classification for medical diagnosis. In: 5th International conference on computing, communication and security (ICCCS), pp 1–5. https://doi.org/10.1109/ICCCS49678.2020.9277277

Modu B et al (2017) Towards a predictive analytics-based intelligent malaria outbreak warning system. Appl Sci 7:836. https://doi.org/10.3390/app7080836

Random Forest machine learning algorithm. https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/. Accessed on 2 July 2021

Ruban S, Rai S (2021) Enabling data to develop an AI-based application for detecting malaria and dengue. In: Tanwar P, Kumar P, Rawat S, Mohammadian M, Ahmad S (eds) Computational intelligence and predictive analysis for medical science: a pragmatic approach. De Gruyter, Berlin, pp 115–138. https://doi.org/10.1515/9783110715279-006

Ruban S, Naresh A, Rai S (2021) A noninvasive model to detect malaria based on symptoms using machine learning. In: Advances in parallel computing technologies and applications. IOS Press, pp 23–30

Sangwon C, Sungjun K, Donghyun L (2018) Predicting infectious disease using deep learning and big data. Int J Environ Res Public Health 5(8):1596

Shivakumar, Rajesh BV, Kumar A, Achari M, Deepa S, Vyas N (2015) Malarial trend in Dakshina Kannada, Karnataka: an epidemiological assessment from 2004 to 2013. Indian J Health Sci 8:91–94

Sidey-Gibbons and Sidey-Gibbons (2019) Machine learning in medicine: a practical introduction. BMC Med Res Methodol 19:64. https://doi.org/10.1186/s12874-019-0681-4

Smith R (2007) An overview of the Tesseract OCR engine. In: Proceedings of the ninth international conference on document analysis and recognition (ICDAR). IEEE Computer Society, pp 629–633

Sundram BM, Raja DB, Mydin F, Yee TC, Raj K (2019) Utilizing artificial intelligence as a Dengue surveillance and prediction tool. J Appl Bioinforma Comput Biol 8:1

Symptoms of Malaria. https://www.cdc.gov/malaria/about/faqs.html. Accessed on 26 June 2021

Valson JS, Soman B (2017) Spatiotemporal clustering of Dengue cases in Thiruvananthapuram district, Kerala. Indian J Public Health 61:74–80

Wang M, Wang H, Wang J, Liu H, Lu R, Duan T et al (2019) A novel model for malaria prediction based on ensemble algorithms. PLoS ONE 14(12). https://doi.org/10.1371/journal.pone.0226910

Weng J, McClelland J, Pentland A, Sporns O, Stockman I, Sur M et al (2020) Autonomous mental development by robots and animals. Science 291(5504):599–600

Wong ZSY et al (2019) Artificial Intelligence for infectious disease big data analytics. Infect Dis Health 24:44–48

World Health Organization (WHO). Malaria report available at https://www.who.int/malaria/publications/world-malaria-report-2019/en/. Accessed on 2 Sept 2020

XGBOOST machine learning algorithm. https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/. Accessed on 2 July 2021

Hybrid Data Science Approaches to Predict the Academic Performance of Students

Saleem Malik and Saleem Malik

1 Introduction

Timely categorization as well as prediction of student level of performance provides a heads-up and even a plan for enhancing both the students' and many other administrative settings substandard performance (Romero and Ventura 2010). Therefore, our goal is to address the unidentified student behavior pattern that has an impact on performance. The achievement of students is influenced by a variety of elements, including student background and behavior, assessments, teaching, and domain knowledge (Kotsiantis et al. 2007). These associated elements served as predictive criteria for forecasting student success. In order to track students' academic development, supervised learning was utilized to predict, categorize, and assess their performance. The difficult challenge is selecting the best algorithm that can deliver pleasing outcomes, though. Student success was often reviewed as well as predicted using machine learning methods such as support vector machine, random forest, K-nearest neighbor, logistic regression, artificial neural network, decision tree, XGboost, AdaBoost, and Naïve Bayes (Farid and Ahmad 2019; Sokkhey and Okazaki 2019). Depending on the dimensions and nature of the data, each model's behavior differs from dataset to dataset. In the classification problem, poor data quality can lead to misclassification, which lowers model accuracy and baffles the algorithms. Sokkhey and Okazaki (2020) released an article in which the author's main objective was to apply principal component analysis to break down the connected observable

Present Address:
S. Malik (✉) · S. Malik
CSE Department, KVGCE, Sullia 574327, India
e-mail: baronsaleem@gmail.com

S. Malik
e-mail: baronsaleem@gmail.com

S. Malik
CSE Department, Graphic Era University, Uttrakhand 248002, India

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_39

predictors into a more manageable collection of independent dimensions or components. We used factor analysis as a feature extraction approach to turn the original dataset into a new dataset of high-quality data with the intention of delving deeper to investigate and test the model for the existence of hidden variables that may be driving the variance in the data. The models' predictive performance is assessed, and their performance in a fresh dataset, testing samples, or test data is assessed using tenfold cross-validation. This research intends to propose a machine learning framework called arbitrator miniature for the classification issue. The suggested framework incorporates tenfold cross-validation, factor analysis, and nine fundamental machine learning algorithms (support vector machine, random forest, K-nearest neighbor, logistic regression, artificial neural network, decision tree, AdaBoost, extra trees, and Naïve Bayes).

1.1 Significant Objectives of this Examination

- To develop an arbitrator miniature framework for predicting the academic performance of students
- Apply arbitrator miniature to academic datasets.
- Evaluating arbitrator miniature framework with other ML models and compare their scores like accuracy and Root Mean Square Error (RMSE).

2 Related Works

The efficient prediction model is crucial for supervised machine learning in order to address prediction as well as classification issues. As mentioned in the introduction, the field of educational data mining (EDM) has explored various machine learning approaches to assess their effectiveness in predicting students' future academic performance. These approaches include machine learning techniques such as support vector machines, decision trees, naive Bayes, and random forests (Sokkhey and Okazaki 2020; Kumar et al. 2022; Polyzou and Karypis 2016; Phauk and Takeo 2020). In order to improve classification performance by addressing the misclassification problem, we expanded it in this work by introducing a hybrid approach of factor analysis in conjunction with nine machine learning algorithms: support vector machine, random forest, K-nearest neighbor, logistic regression, artificial neural network, decision tree, extra trees, AdaBoost, and Naïve Bayes. The common and cutting-edge classification techniques being used to evaluate student performance are listed in Table 1. Finding the most effective algorithms to forecast performance has involved examining a variety of studies (Ghorbani and Ghousi 2020).

Although the hold-out approach is more frequently employed in predicting student performance, the k-fold cross-validation is still a robust cross-validation technique (Pandey and Taruna 2014). In their study, Sassirekha and Vijayalakshmi (2022)

Table 1 Overview of existing works

Reference	Algorithms used	Subject of research	Disadvantages
Sokkhey and Okazaki (2020)	Support vector machines, decision trees, Naive Bayes, random forests	Student academic performance prediction	Only grade prediction no prediction of marks
Kumar et al. (2022), Phauk and Takeo (2020)	Support vector machine, random forest, K-nearest neighbor, logistic regression, artificial neural network, decision tree, extra trees, AdaBoost, Naive Bayes	Student academic performance prediction with factor analysis	Accuracy and RMSE is low
Ghorbani and Ghousi (2020)	Discriminative and generative classification models	Evaluation of various ML studies	Only grade prediction no prediction of marks
Sassirekha and Vijayalakshmi (2022)	K-nearest neighbor, Naive Bayes, decision tree	Student performance prediction using tenfold cross-validation	RMSE is low
Zohair (2019a), Kamala (2019)	Ensemble supervised learning	Various data mining techniques' comparison	Accuracy is low
Sharma and Mishra (2022)	Logistic regression, decision tree	Predicting academic achievement with imbalanced class issue	Model was tested only on one course
Rawat and Malhan (2019)	Feature importance techniques, algorithms	Addressing classification issue	Only five attributes are used for prediction, and other attributes are ignored
Pandey and Taruna (2014)	K-fold cross-validation	Predicting student grades for future courses	Only grade prediction no prediction of marks

implemented contrasted K-nearest neighbors, Naive Bayes, and decision tree algorithms while utilizing tenfold cross-validation to assess the performance of various machine learning methods in a classification task. Other studies, including (Zohair 2019a) and (Kamala 2019), used k-fold cross-validation to compare various data mining techniques using models to predict student performance.

This is common knowledge how raising a model's forecast accuracy may be difficult. Accurate prediction may be improved by a variety of things. The feature selecting and addressing the misclassification are some of the key elements. Sharma and Mishra (2022) created the logistic regression and decision tree models which can anticipate students' academic achievement despite managing an uneven class issue (Rawat and Malhan 2019). Emphasis was placed on creating diverse utilizing the feature importance technique and algorithms to address the classification issue.

2.1 Identifying Research Gap

Prognosticating student accomplishment is a challenging endeavor that calls for novel, inventive ways that take into account the changing conditions and factors that affect students' academic performance. The effect of these variables and situations may change from institution to institution and from one class of students to another (Livieris et al. 2018; Rao and Arunachalam 2021). Our thorough analysis of the linked studies in Table 1 found deficiencies in the following areas:

1. Absence of hybrid systems that simplify and enhance the accuracy of predictions of student academic achievement and combine the benefits of supervised and unsupervised learning.
2. The rigidity of current models when it comes to analyzing various academic and non-academic aspects that are thought to affect how well students learn. While some techniques just take into account a limited portion of relevant components, some approaches forecast student outcomes without linking them to the enabling factors or potential shortcomings.
3. The validity of the model is questioned since so many student prediction models are verified using a single dataset.

3 Materials and Methods

3.1 Datasets

Throughout this research, we made an effort to compile all latent factors influencing students' performance. We used four datasets to verify the resilience and efficiency of our suggested methods. Details of this dataset are shown in Table 5. Dataset included one target variable that described the performance levels of students based on their score and 48 characteristics that described the information about each student's learning activities. The features observing from the three primary affected factors make up the predicted features. The forty-nine variables made up of these main elements are described in Table 2. The target variable's preset classes are listed in Table 3. The real dataset, known as AD1, has 1500 samples, which was gathered from polytechnic colleges in Karnataka, India. Use of questionnaires was employed for the data collecting. Students were requested to provide demographic data pertaining to outside influences including family, individual, or student, and school issues. The administrative offices in each school provided the math, it skills, and statistics test results for the first semester's students.

Table 2 Impacting attributes on student performance

Category of factors	Variables
Domestic factors	College, gender, family size, mother education, father education, mother occupation, father occupation, reason for join, guardian, travel time, failure, discrimination, higher education, romantic, family relation, free time, going out, alcohol consumption, smoking, heath, category, previous medium of study, remedial classes, educational loan, poverty level, SSLC/PUC marks, math score, IT skill score, Statistics score, Final score
Individual factors	Self-study, doing homework, absence preparing for exams, interest in subjects, enjoyment in class, attention in class, motivation, anxiety, nervous, helpless, internet at home, possession of computer, study desk at home
Soft skill factors	Reading skill, writing skill, listening skill, speaking skill, basic computer skill

Table 3 Output variables

Level—1	Level—2	Level—3	Level—4	Level—5
Exemplary	Good	Average	Below average	Poor
A	B	C	D	E
Grade between 16 and 20	Grade between 14 and 15	Grade between 12 and 13	Grade between 10 and 11	Grade between 0 and 9

3.2 Preprocessing

In this research, preprocessing activities included operations like data cleansing, data transformation, and data discretization. The completion of the questionnaire during data collection included ignoring some items and entering erroneous values (outliers) (Marbouti et al. 2016). Since there were few missing values in our datasets, we cleaned the data using the imputing method. We substituted its modes or high frequency category values for the missing value in our categorical variables. We replaced a few missing values and outliers in the output variable with the mean value. We converted several numerical properties into ordinal types for convenience. As indicated in Table 3, we discretized the output variables in our study into five performance levels (Chaudhury and Hrudaya 2020; Febro 2019).

3.3 Machine Learning Models

Machine learning models may be categorized in many different ways. A variety of classifiers are used in this study, including random forest (RF) (Mohamed et al. 2022; Hussain et al. 2018), K-nearest neighbor (KNN) (Badal and Sungkur 2022; Al-Zawqari et al. 2022), artificial neural network (ANN) (Arun et al. 2021; Aladeemy et al. 2017; Wang et al. 2015), extra tress (Raj and Renumol 2022; Alshanqiti and Namoun 2020), support vector machine (SVM) (Huijuan et al. 2017; Zorarpacı and Ozel 2016), decision tree (DT) (Sharma and Mishra 2022; Alyahyan and Düştegör 2020), AdaBoost (Malik et al. 2023; Khan et al. 2021), logistic regression (LR) (Rai et al. 2021; Kou et al. 2020), and Naive Bayes (NB) (Hasan et al. 2020; Badal and Sungkur 2022). It is commonly known that the majority of machine learning classifiers, including artificial neural networks (ANNs), K-nearest neighbor (KNN), random forests (RFs), logistic regressions (LRs), decision trees (DT), and Naive Bayes naturally allow multiclass classification (NB). Since support vector machine (SVM) and extra trees do not naturally support multiclass classification, the support vector machine model is applied using the one-versus-one approach, and the extra trees model is applied using the one-versus-all method (Hussain et al. 2019).

Table 4 lists all of the machine learning models that were employed in this study along with the values for each of their unique parameters. Similar parameter setting is utilized in Francis and Babu (2019). The parameter setting described in Huynh-Cam et al. (2022) has been adjusted in our study to better fit our issue description. To create predictions, these factors are essential.

Table 4 Machine learning models with their specific parameters' settings

Methods	Parameter setting
Random forest (RF)	Number of trees: 100, max. depth: 10, min. samples leaf: 2, max. features: sqrt
K-nearest neighbor (KNN)	Number of neighbors: 5, distance metric: Euclidean, algorithm: Ball tree
Artificial neural network (ANN)	Number of hidden layers: 2, number of units per layer: 64, activation function: ReLU, learning rate: 0.001, batch size: 32
Extra trees	Number of trees: 200, max. depth: none (unlimited), min. samples leaf: 5, max. features: log2
Decision tree (DT)	Criterion: Gini, max. depth: 8, min. samples split: 10
Logistic regression (LR)	Regularization: L2, C (inverse of regularization strength): 1.0, solver: LBFGS
Naive Bayes (NB)	Distribution type: Gaussian
AdaBoost	Number of estimators: 50, learning rate: 0.1, loss function: exponential
Support vector machine (SVM)	Kernel: RBF, C (regularization parameter): 1.0, Gamma: 0.01

Table 5 Description of datasets used in the study

Datasets	Source	Type	Samples	Features	Labels
DS1	Generated	Academic	2000	30	5
DS2	Generated	Academic	1000	5	5
DS3	Generated	Academic	3000	30	5
AD1	Real	Academic	1500	49	5

4 Proposed Arbitrator Model

Besides incorporating factor analysis as well as machine learning technologies, arbitrator model is presented. At initial stage, baseline models are suggested. With k-fold cross-validation, performance of suggested baseline models is enhanced. Finally, by merging factor analysis with machine learning, as shown in Fig. 1, arbitrator model is suggested.

4.1 Baseline Models

A number of efficient machine learning techniques have been widely used in academic settings. We need to use a variety of machine learning approaches in learning environments, including association rule mining, regression analysis, classification, and clustering (Brahim and Limam 2016; Zohair 2019b). In classifying and predict the target variables' preset classes, classification is a typical machine learning approach. In this study, we looked at a number of machine learning classifiers and chose five cutting-edge techniques that are often used to forecast academic success (Alyahyan and Düştegör 2020; Gajwani and Chakraborty 2021). This study proposes a hybrid approach known as arbitrator miniature that combines factor analysis with the following nine machine learning techniques: support vector machine, random forest, K-nearest neighbor, logistic regression, artificial neural network, decision tree, AdaBoost, extra trees, and Naïve Bayes.

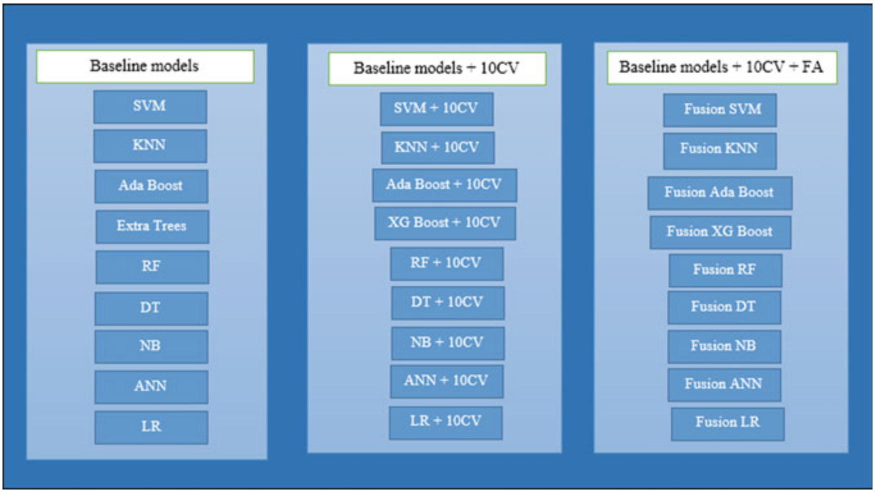


Fig. 1 Proposed arbitrator miniature model

4.2 Cross-validation

A quantitative approach used to gauge how well machine learning algorithms work is cross-validation. There are different cross-validation techniques (Sokkhey and Okazaki 2020), but the k-fold cross-validation is used because it is well-liked and simple to comprehend and because it often produces less distortion than the other techniques (Ghareb et al., 2016). Tenfold cross-validation approach is used in this study, also known as 10-CV, to reach proposed model.

4.3 Arbitrator Miniature

In supervised machine learning, classification comprises the preponderance of tasks. A challenging topic in data mining and machine learning is the classification problem (Romero and Ventura 2010). We suggested the nine renowned classifiers with lots of advantages. Nevertheless, the main the issue with all of above-mentioned classifiers is noisy data and over fitting, which makes it possible to determine the correctness of the classification. In an effort to solve this issue, we strive to unnecessary and unrelated traits that are disruptive in the process of categorization. When analyzing data, it requires additional computing resources, which results in longer processing times. Data often come in large volumes. Consequently, the feature extraction method is employed to eliminate noise from the data, which helps save time, reduce resource usage, and restore high-quality data. By integrating a dimensional reduction with classification approaches, the accuracy and efficiency is enhanced. Steps used in this study are shown in Fig. 2.

The process of converting the initial dataset X of size l, which contains potentially related features, into a new dataset Z of size $m(m < \mathrm{l})$, which contains linear unrelated characteristics is as follows:

Step 1: Find the mean of each feature based on the previously processed data using the following equation:

$$\mu = 1/n \sum_{i=0}^{n} * x_i. \tag{1}$$

Step 2: Calculate the variance using below formula in order to explore and deviation of each feature in the dataset

$$\mathrm{Var}(X) = s_x^2 = 1/n \sum_{i=0}^{n} * (x_i -)^2. \tag{2}$$

Step 3: Equation (3) is used to determine the covariance and correlation for two variables, labeled as X and Y.

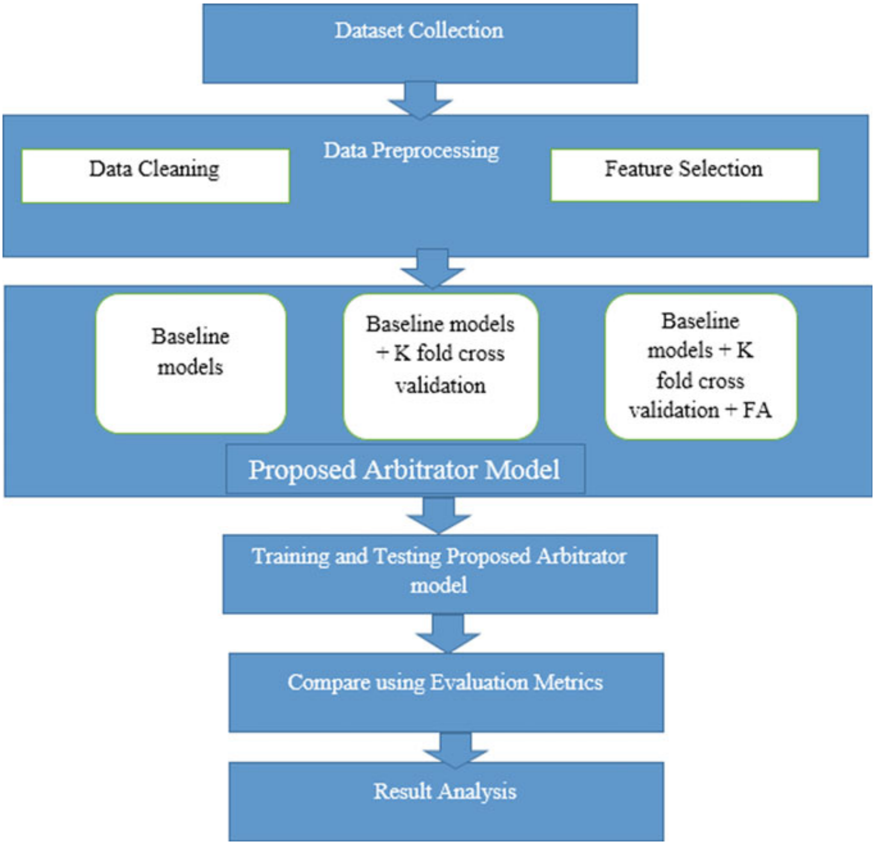

Dataset Collection
Data Preprocessing
Data Cleaning
Feature Selection
Baseline models
Baseline models + K fold cross validation
Baseline models + K fold cross validation + FA
Proposed Arbitrator Model
Training and Testing Proposed Arbitrator model
Compare using Evaluation Metrics
Result Analysis

Fig. 2 Using the proposed model in this study

$$\mathrm{Cov}(X, Y) = s_x^2 = 1/n \sum_{i=0}^{n} *(x_i - \mu_x)y_i - \mu_y). \tag{3}$$

Step 4: The mean of the eigenvectors and eigenvalues is employed to characterize features across multiple datasets. The acquired eigenvectors indicate the directions, while the eigenvalues represent their magnitudes in the new feature space. It is feasible to find the eigenvalues by solving the equation.

$$\mathrm{Det}(S - \lambda I) = 0. \tag{4}$$

We need to select a dataset with a range of feature sets that is related to the students' academic performance. Any and all inconsistencies that could have existed in the dataset at the time the data were gathered must be removed during the preparation stage. It is now time to test our dataset in two different ways: the first mode will test our dataset using all of the characteristics that are now there, and the second mode will pick the factors to test using proposed arbitrator miniature. In this case, we find the accuracy of the machine learning algorithm in three phases. Phase 1 is to find the accuracy of baseline models. Phase 2 is to improve the accuracy of baseline models by tenfold cross-validation (10-CV). Phase 3 is to execute proposed arbitrator miniature which is the combination of the baseline models with 10-CV and FA.

5 Evaluation Metrics

The assessment of the pictorial confusion matrix can be used to assess how well each proposed model has performed in analyzing and forecasting student performance. Our output variable can be divided into five ordinal categories, as shown in Table 1, without losing coherence. The proportion of successfully predicted outcomes is measured by accuracy (Khan and Ghosh 2021). By evaluating the percentage of accurately estimated student performance levels, we can assess the potential of our prediction model in this case (Pandey and Taruna 2014).

$$\text{Accuracy} = \frac{\sum \mathbf{X}}{\sum \mathbf{Y}} * \mathbf{100}, \tag{5}$$

$$\text{RMSE} = \sqrt{\sum_{i=1}^{m} \frac{(X - Y)2}{\text{M}}}. \tag{6}$$

Our objective is to determine how near our forecast is to the academic achievement of the students as well as their skill levels. These ordinal skill sets, average, good, and excellent—were encoded as 1, 2, 3, and 4, respectively. Calculating the RMSE involves: The predicted skill level is given by Y = {1, 2, 3, 4} and the actual performance level is given by $X = \{1, 2, 3, 4\}$. In contrast to accuracy, the model is better, the smaller the RMSE. The prediction model is perfect if the RMSE is equal to 0 (Miguéis et al. 2018).

6 Result Analysis

6.1 Experimental Setup

The goal of this research is to construct a unique ensemble learning-based intelligent prediction model to forecast student performance. The suggested strategy made use of cross-validation, machine learning, and baseline model combinations. The classification results of these classifiers were subjected to ensemble learning. In order to evaluate the effectiveness of ensemble learning-based classifiers, tests were carried out using a Python program with Tensor Flow, Keras, and other related libraries. On an 11th generation Intel Core i7 with two NVIDIA GeForce RTX 3060 Laptop GPUs with 6.0 GB and 7.9 GB of RAM, research was conducted.

6.2 Sources of Datasets

We used four datasets to verify the resilience and efficiency of our suggested methods. Details of this dataset are shown in Table 5, four datasets are used to access the whole approach, and our focus is on predicting student's grades and factors for only the next upcoming semester. As previously indicated, it might be challenging to locate the necessary scholarly datasets for this strategy in readily accessible sources. Additionally, none of the associated university-level research that we are aware of provided their datasets for experimental replication, most likely due to data privacy limitations (Sokkhey and Okazaki 2020).The three datasets, DS1, DS2, and DS3, are produced datasets that were built using suggested characteristics for the output variable as indicated in Chaudhury and Hrudaya (2020), Febro (2019), Marbouti et al. (2016). Fourth one is the real dataset, known as AD1, which was gathered from polytechnic colleges in Karnataka, India. Use of questionnaires was employed for the data collecting. Investigations are conducted on three different stages. Stage 1 involves putting the results of the baseline models into practice. Stage 2 involves a tenfold cross-validation improvement of the baseline models (10-V). Stage 3 involves putting the proposed model—a fusion of the baseline models with 10-CV and FA—into action.

6.3 Implications of Baseline Models

We put forward the nine most well-liked machine learning methods. The tables display the two performance measures, classification accuracy and RMSE. Tables 6 and 7 (Bold values represent the best performance achieved by each classifier for their respective datasets) reveal that the extra trees model was the least, but the random forest approach produced the best results in terms of classification accuracy and RMSE, indicating that it might be a good model.

6.4 Implications of Baseline Models with K-Fold Cross-Validation

In prediction and classification models, the k-fold cross-validation approach is frequently used to divide the dataset into k-1 sub-folds for training sets and onefold

Table 6 Accuracy of classification algorithms utilizing various datasets

Models		Accuracy (baseline) (%)	Accuracy (baseline + 10-CV) (%)	Accuracy (proposed model) (%)
	SVM	85.36	86.66	**87.32**
	Decision tree	73.50	75.34	**80.50**
	AdaBoost	74.45	75.48	**81.33**
	Random forest	85.71	87.28	**91.79**
	Extra trees	60.76	72.66	**80.57**
	NB	77.64	83.62	**87.75**
	Neural network	75.87	81.43	**88.65**
	KNN	80.65	85.34	**90.45**
	Logistic regression	65.76	77.45	**83.65**
Performance of classifiers using dataset DS2	SVM	85.96	87.76	**88.46**
	Decision tree	74.10	77.94	**81.98**
	AdaBoost	74.89	76.41	**83.67**
	Random forest	86.11	89.88	**92.70**
	Extra trees	61.76	73.09	**83.07**
	NB	76.68	82.12	**88.75**
	Neural network	77.34	83.33	**87.65**
	KNN	83.62	86.56	**87.45**
	Logistic regression	68.16	79.56	**83.65**
Performance of classifiers using dataset DS3	SVM	86.16	88.21	**91.12**
	Decision tree	74.94	79.42	**85.67**
	AdaBoost	76.29	80.81	**87.45**
	Random forest	87.15	90.63	**95.89**
	Extra trees	63.36	79.09	**82.33**
	NB	78.78	87.10	**90.89**
	Neural network	79.04	85.56	**89.09**
	KNN	82.60	89.87	**93.34**
	Logistic regression	70.16	77.58	**84.31**
Performance of classifiers using dataset AD1	SVM	87.46	91.66	**93.32**
	Decision tree	75.57	83.34	**89.50**
	AdaBoost	77.03	84.48	**91.33**
	Random forest	89.73	93.28	**97.79**
	extra trees	67.56	80.66	**86.57**
	NB	83.78	88.90	**93.89**

(continued)

Table 6 (continued)

Models		Accuracy (baseline) (%)	Accuracy (baseline + 10-CV) (%)	Accuracy (proposed model) (%)
	Neural Network	79.97	83.14	**92.78**
	KNN	84.09	91.97	**95.07**
	Logistic Regression	73.45	81.59	**90.31**

for testing sets, then rotate the folds. Since it functions best at this split, we performed tenfold cross-validation in this trial. Ten % of the data was utilized for testing, while 90% was used in the training stage. The average of all assessment criteria is then calculated when all interactions have been completed. The accuracy of SVM was increased by 4% as shown in Tables 6 and 7. The weak extra trees classifier's performance was subsequently markedly enhanced.

6.5 Implications of Proposed Model

By combining the baseline models with a feature reduction strategy called FA, we were able to create the fusion models we have suggested. One of the effective techniques in classification models for eliminating unrelated or unnecessary features is feature extraction. Dimensionality reduction by FA (Rawat and Malhan 2019; Mohamed et al. 2022) may most certainly be used as regularization to avoid over fitting and boost model accuracy. People frequently fall into the trap of believing that FA chooses certain characteristics from the dataset while discarding others. Actually, the algorithm creates a fresh dataset of attributes by combining the previous ones. Tables 6 and 7 demonstrate how the suggested model helps classifiers become more accurate (Khan and Ghosh 2021).

Figures 3 and 4 show how each model performed according to its accuracy during each step. We discovered that the 10-CV improvement in conjunction with FA produces the greatest results in forecasting student performance. Figures 3 and 4 display how well the models' RMSE performed at each phase. The proposed models that have been suggested could produce relatively little RMSE. In this prediction scenario, the hybrid RF algorithm provided the least value of RMSE, demonstrating its superiority as the best predictive model. According to the findings, we may enhance the performance of our basic models by employing 10-CV. We also noticed that the innovative fusion models that were suggested may improve classification performance and produce better outcomes. The suggested proposed models can be viewed as the best prediction models for resolving categorization and prediction issues.

Table 7 RMSE of classification algorithms utilizing various datasets

Models		RMSE (baseline)	RMSE (baseline + 10-CV)	RMSE (proposed model)
	SVM	0.701	0.691	**0.410**
	Decision tree	1.033	0.914	**0.681**
	AdaBoost	1.134	0.721	**0.521**
	Random forest	0.602	0.474	**0.321**
	Extra trees	1.164	0.931	**0.664**
	NB	0.876	0.603	**0.489**
	Neural network	0.908	0.827	**0.532**
	KNN	0.779	0.546	**0.369**
	Logistic regression	1.023	0.943	**0.787**
Performance of classifiers using dataset DS2	SVM	0.611	0.521	**0.401**
	Decision tree	0.903	0.804	**0.654**
	AdaBoost	1.022	0.771	**0.489**
	Random forest	0.634	0.454	**0.309**
	Extra trees	1.055	0.841	**0.598**
	NB	0.89	0.765	**0.633**
	Neural network	0.9	0.667	**0.609**
	KNN	0.8	0.698	**0.577**
	Logistic regression	0.955	0.9	**0.799**
Performance of classifiers using dataset DS3	SVM	0.7	0.554	**0.266**
	Decision tree	0.89	0.676	**0.356**
	AdaBoost	0.877	0.6	**0.334**
	Random forest	0.614	0.289	**0.119**
	Extra trees	0.767	0.556	**0.455**
	NB	0.745	0.599	**0.476**
	Neural network	0.776	0.567	**0.324**
	KNN	0.696	0.435	**0.235**
	Logistic regression	0.8	0.676	**0.514**

(continued)

Table 7 (continued)

Models		RMSE (baseline)	RMSE (baseline + 10-CV)	RMSE (proposed model)
Performance of classifiers using dataset AD1	SVM	0.813	0.391	**0.141**
	Decision tree	0.845	0.414	**0.288**
	AdaBoost	0.837	0.521	**0.321**
	Random forest	0.506	0.174	**0.021**
	Extra trees	1.16	0.431	**0.364**
	NB	0.778	0.567	**0.276**
	Neural network	0.656	0.453	**0.309**
	KNN	0.567	0.369	**0.176**
	Logistic regression	0.805	0.654	**0.377**

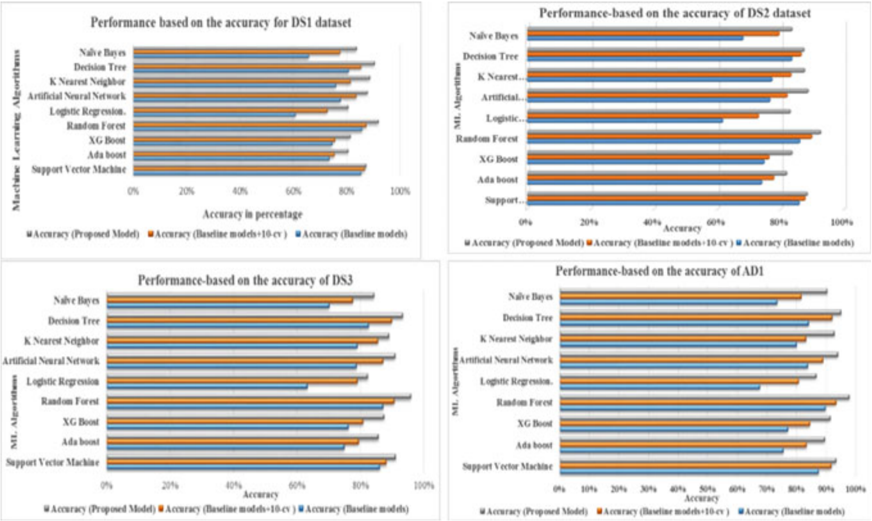


Fig. 3 Accuracy for DS1, DS2, DS3, and AD1 datasets

7 Final Thoughts and Way Forward

The nine well-known machine learning classifiers for predicting student performance were introduced in this study. The nine suggested algorithms are: support vector machine, random forest, K-nearest neighbor, logistic regression, artificial neural network, decision tree, extra trees, AdaBoost, and Naïve Bayes. Three stages make up

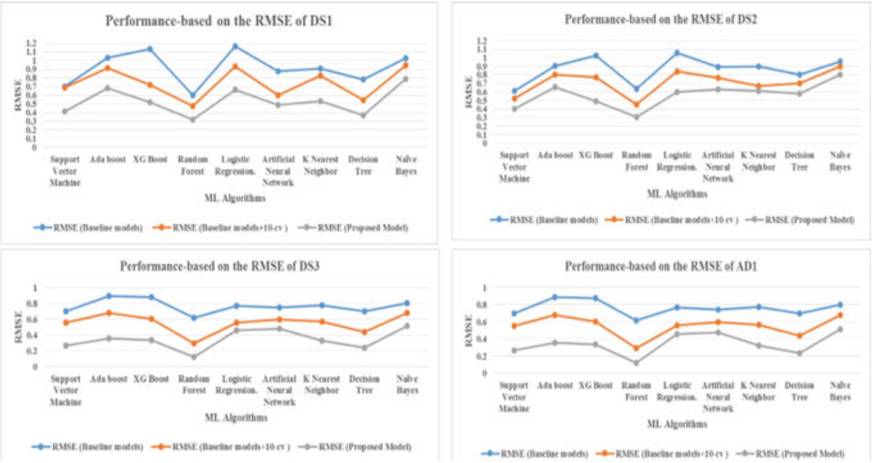


Fig. 4 RMSE for DS1, DS2, DS3, and AD1 datasets

the operation. First, we evaluated how well various baseline techniques performed. Second, we used 10-CV to boost performance. Finally, to enhance classification performance, we integrated the 10-CV approach with FA to baseline models. Based on classification accuracy and RMSE as measurement parameters, it can be seen that the suggested arbitrator miniature models gave highly pleasing results when combined with FA and 10-CV. In conclusion, the proposed models provided a high performance that demonstrates itself as a possible approach for solving the problem by merging the baseline models with factor analysis and assessed by k-fold cross-validation. We consider a thorough examination of algorithms using different performance indicators since it helps us to comprehend the algorithms better. Different performance criteria, including accuracy and RMSE, were taken into account. By contrasting several algorithms using various criteria, it is possible to assess an algorithm's predictability. The literature study indicates that because each problem has specific features, it is challenging to forecast which performance metrics are better for certain challenges. As a result, combining numerous measurements is advised for improved algorithm performance. Using several feature selection techniques, we examined the effectiveness of various selected data mining methods on the relevant dataset. Finally, we can state that any classification algorithm's performance when combined with feature prediction gives better result. This study's methodology will help educational leaders and legislators create new regulations and curricula pertaining to student retention in higher education. Additionally, by identifying students who are at danger of quitting school early, this research enables prompt assistance and intervention. We compare the performance accuracy and efficiency of individual classifiers and classifier ensembles. This model can assist academics in identifying children with learning difficulties, developing the learning experience, and lowering educational failure rates. It can also assist managers in managing more effectively depending on the outcomes of the learning system. The model could be expanded, refined in the future to encompass a broad range of student dataset attributes.

Preprint Version

Saleem Malik & Jothimani K. Arbitrator Miniature: A Paradigm using Data Science Methods to Predict Academic Performance. Preprint at https://doi.org/10.21203/rs.3.rs-2271636/v1 (2022).

References

Aladeemy M, Tutun S, Khasawneh M (2017) A new hybrid approach for feature selection and support vector machine model selection based on self-adaptive cohort intelligence. Expert Syst Appl 88:118–131. https://doi.org/10.1016/j.eswa.2017.06.030

Alshanqiti A, Namoun A (2020) Predicting student performance and its influential factors using hybrid regression and multi-label classification. IEEE Access. 8:203827–203844. https://doi.org/10.1109/ACCESS.2020.3036572

Alyahyan E, Düştegör D (2020) Predicting academic success in higher education: Literature review and best practices. Int J Educ Technol Higher Edu 17(1), Dec 2020

Alyahyan E, Düştegör D (2020) Predicting academic success in higher education: literature review and best practices. Int J Educ Technol High Educ 17(1):3

Al-Zawqari A, Peumans D, Vandersteen G (2022) A flexible feature selection approach for predicting students' academic performance in online courses. Comput Educ Artif Intell 3, [100103]. https://doi.org/10.1016/j.caeai.2022.100103

Arun DK, Namratha V, Ramyashree BV, Jain YP, Choudhury AR (2021) Student academic performance prediction using educational data mining, 2021. In: International conference on computer communication and informatics (ICCCI) (2021)

Badal Y, Sungkur R (2022) Predictive modelling and analytics of students' grades using machine learning algorithms. Educ Inf Technol. https://doi.org/10.1007/s10639-022-11299-8

Brahim AB, Limam M (2016) A hybrid feature selection method based on instance learning and cooperative subset search. Pattern Recognit Lett 69:28–34

Chaudhury P and Hrudaya T (2020) A novel academic performance estimation model using two stage feature selection. Indonesian J Electric Eng Comput Sci 19:1610. https://doi.org/10.11591/ijeecs.v19.i3.pp1610-1619

Farid J, Ahmad AS (2019) Building student's performance cesissiion tree classifier using boosting algorithm. Indonesian J Electri Eng Comput Sci 14(3):1298–1304

Febro J (2019) Utilizing feature selection in identifying predicting factors of student retention. Int J Adv Comput Sci Appl 10. https://doi.org/10.14569/IJACSA.2019.0100934

Francis BK, Babu SS (2019) Predicting academic performance of students using a hybrid data mining approach. J Med Syst 43:162. https://doi.org/10.1007/s10916-019-1295-4

Gajwani J, Chakraborty P (2021) Students' performance prediction using feature selection and supervised machine learning algorithms. In: International conference on innovative computing and communications. Springer, Singapore, pp 347–354

Ghareb AS, Bakar AA, Hamdan AR (2016) Hybrid feature selection based on enhanced genetic algorithm for text categorization. Expert Syst Appl 49:31–47

Ghorbani R, Ghousi R (2020) Comparing different resampling methods in predicting students' performance using machine learning techniques. IEEE Access 8(1):67899–67911, Apr 2020

Hasan R, Palaniappan S, Mahmood S et al (2020) Predicting student performance in higher educational institutions using video learning analytics and data mining techniques. Appl Sci 10(11):3894

Huijuan L, Chen J, Yan K, Jin Q, Gao Z (2017) A hybrid feature selection algorithm for gene expression data classification. Neurocomputing 256:56–62

Hussain M, Zhu W, Zhang W et al (2019) Using machine learning to predict student difficulties from learning session data. Artif Intell Rev 52:381–407. https://doi.org/10.1007/s10462-018-9620-8

Hussain M, Zhu W, Zhang W, Abidi R (2018) Student engagement predictions in an e-learning system and their impact on student course assessment scores. Comput Intell Neurosci 2018:1–21. https://doi.org/10.1155/2018/6347186

Huynh-Cam T-T, Chen L-S, Huynh K-V (2022) Learning performance of international students and students with disabilities: early prediction and feature selection through educational data mining. Big Data Cognitive Comput 6:94. https://doi.org/10.3390/bdcc6030094

Kamala R, Thangaiah RJ (2019) An improved hybrid feature selection method for huge dimensional datasets. IAES Int J Artif Intell (IJ-AI) 8(1):77–86, Mar 2019. ISSN: 2252-8938. https://doi.org/10.11591/ijai.v8.i1.pp77-86

Khan A, Ghosh SK, Ghosh D, Chattopadhyay S (2021) Random wheel: an algorithm for early classification of student performance with confidence. Eng Appl Artif Intell

Khan A, Ghosh SK (2021) Student performance analysis and prediction in classroom learning: a review of educational data mining studies. Educ Inf Technol 26:205–240. https://doi.org/10.1007/s10639-020-10230-3

Kotsiantis S, Piarrekeas C, Pintelas P (2007) Predicting students' performance in distance learning using machine learning techniques. Appl Artif Intell 18:411–426

Kou G, Yang P, Peng Y et al (2020) Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods. Appl Soft Comput 86:105836

Kumar M, Nidhi BS, Handa D (2022) Building predictive model by using data mining and feature selection techniques on academic dataset. Int J Mod Educ Comput Sci (IJMECS) 14(4):16–29. https://doi.org/10.5815/ijmecs.2022.04.02

Livieris IE, Drakopoulou K, Mikropoulos TA, Tampakas V, Pintelas P (2018) An ensemble-based semi-supervised approach for predicting students performance. In: Research on e-learning and ICT in education. Springer, Cham, Switzerland, pp 25–42

Malik S, Jothimani K, Ujwal UJ (2023) A comparative analysis to measure scholastic success of students using data science methods. In: Shetty NR, Patnaik LM, Prasad NH (eds) Emerging research in computing, information, communication and applications. lecture notes in electrical engineering, vol 928. Springer, Singapore. https://doi.org/10.1007/978-981-19-5482-5_3

Marbouti F, Diefes-Dux H, Madhavan K (2016) Models for early prediction of at-risk students in a course using standards-based grading. Comput Educ 103. https://doi.org/10.1016/j.compedu.2016.09.005

Miguéis VL, Freitas A, Garcia PJ et al (2018) Early segmentation of students according to their academic performance: a predictivemodelling approach. Decis Support Syst 115:36–51

Mohamed Y, Alkawsi G, Mustafa A, Alkahtani A, Alsariera Y, Ali A, Hashim W, Kiong T (2022) Toward predicting student's academic performance using artificial neural networks (ANNs). Appl Sci 12. https://doi.org/10.3390/app12031289

Pandey M, Taruna S (2014) A comparative study of ensemble methods for students' performance modeling. Int J Comput Appl 103(8):26–32, Oct 2014

Phauk S, Takeo O (2020) Study on dominant factor for academic performance prediction using feature selection methods. Int J Adv Comput Sci Appl 11:492–502. https://doi.org/10.14569/IJACSA.2020.0110862

Polyzou A, Karypis G (2016) Grade prediction with models specific to students and courses. Int J Data Sci Anal 2(3–4):159–171

Rai S, Shastry KA, Pratap S et al (2021) Machine learning approach for student academic performance prediction. In: Evolution in computational intelligence. Springer, Singapore, pp 611–618

Raj NS, Renumol VG (2022) Early prediction of student engagement in virtual learning environments using machine learning techniques. E-Learning and Digital Media

Rao CS, Arunachalam AS (2021) Ensemble based learning style identification using VARK. NVEO-Natural Volatiles Essent OILS J| NVEO, pp 4550–4559

Rawat KS, Malhan IV (2019) A hybrid classification method based on machine learning classifiers to predict performance in educational data mining. In: Krishna C, Dutta M, Kumar R (eds) Proceedings of 2nd international conference on communication, computing and networking. Lecture notes in networks and systems, vol 46. Springer, Singapore. https://doi.org/10.1007/978-981-13-1217-5_67

Romero C, Ventura (2010) Educational data mining: a review of the state of art. IEEE Trans Syst Man Cybern Part C Appl Rev 40(6):601–618

Sassirekha MS, Vijayalakshmi S (2022) Predicting the academic progression in student's standpoint using machine learning. Automatika 63(4):605–617. https://doi.org/10.1080/00051144.2022.2060652

Sharma A, Mishra PK (2022) Performance analysis of machine learning based optimized feature selection approaches for breast cancer diagnosis. Int J Inf Technol 14:1949–1960. https://doi.org/10.1007/s41870-021-00671-5

Sokkhey P, Okazaki T (2019) Comparative study of prediction models on high school student performance in mathematics. J IEIE Trans Smart Process Comput 8(5):394–404, Oct 2019

Sokkhey P, Okazaki T (2020) Hybrid machine learning algorithms for predicting academic performance. Int J Adv Comput Sci Appl (IJACSA), 11(1)

Wang A, An N, Chen G, Li L, Alterovitz G (2015) Accelerating wrapper-based feature selection with K-nearest-neighbour. Knowl Based Syst 83:81–91

Zohair LMA (2019a) Prediction of student's performance by modelling small dataset size. Int J Educ Technol Higher Edu 16(1):27

Zohair LMA (2019b) Prediction of student's performance by modelling small dataset size. Int J Educ Technol Higher Educ16(1):1–18

Zorarpacı E, Ozel SA (2016) A hybrid approach of differential evolution and artificial bee colony for feature selection. Expert Syst Appl 62:91–103

GAN-Based Image Restoration and Colorization

Aliyah Kabeer, Manali Tanna, K. N. Milinda, Mohammed Uzair Rizwan, and Pooja Agarwal

1 Introduction

Over the past two decades, mobile phone imagery has seen a significant evolution. Thicker lenses or huge image sensors cannot be put on cell phones due to the structure's restriction on the size of the camera module. Both the image's volume and picture quality are impacted by this. It is challenging to capture perceptually pleasing images in light and dark environments especially during rapid movement (Gampala et al. 2020). The best images frequently require extensive postprocessing on the computer. The aesthetic enhancement of a photograph can be achieved through image restoration and image colorization. In order to make a grayscale image more aesthetically pleasing and perceptually relevant, a procedure known as image colorization is used (Sophia et al. 2022). It is acknowledged as a challenging task that frequently necessitates familiarity with the image content beforehand and manual changes to obtain artifact-free quality. There is no right solution to this task because objects might have multiple colors and there are numerous ways to assign colors to pixels in an image. On the other hand, image restoration entails recovering the original image from its noisy and obscured counterpart. It focuses on the elimination or mitigation

A. Kabeer (✉) · M. Tanna · K. N. Milinda · M. U. Rizwan · P. Agarwal
Department of Computer Science and Engineering, PES University, Bangalore, India
e-mail: aliyahk888@gmail.com

M. Tanna
e-mail: manalitanna29@gmail.com

K. N. Milinda
e-mail: milimilindakn@gmail.com

M. U. Rizwan
e-mail: uzairrizwan15@gmail.com

P. Agarwal
e-mail: poojaagarwal@pes.edu

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes
in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_40

of noise degradation brought on by camera motion blur or misfocus (Nguyen et al. 1604).

Image colorization attempts in the past often incorporated user input. Researchers switched to color transfer, which was based on the resemblance between a provided reference image and the input, to lessen the user workload. Deep learning-based automatic colorization techniques were suggested as a way to further lessen this dependence. These techniques identified the connection between the colorful image from a dataset and the monochrome input. Deep learning-based restoration techniques replaced manual deblurring attempts using editing tools. Autoencoders were frequently utilized for image restoration tasks as they included an encoder–decoder framework and symmetric convolutional–deconvolutional layers to extract the key aspects of the image and ignore the degradations while recovering it. However, substantial advancements have been made in these two tasks individually owing to the development of generative adversarial networks (GANs) (Generative adversarial networks 2020). Using an adversarial network, GANs present a generative model that can produce high-quality natural photos that gradually develop to generate more and more realistic-looking data. In addition to being able to produce extremely high-quality synthetic data, this system may also be used to generate images, colorize, and restore them. The majority of current methods for image restoration and colorization use GANs and their variations to create realistic images.

The proposed GAN aims to utilize loss functions specific to both restoration and colorization and uses an architecture that encompasses the features of both. This explores a new side of GANs and introduces a novel architecture for image enhancement tasks. It also paves the way for a new avenue of research that can exploit GANs where two or more tasks can be achieved together. GAN's are usually used to perform one single task at a time and previous research highlights this point as the GAN models proposed earlier in the image enhancement domain only either restore or colorize, but don't perform both together within the same architecture.

2 Related Work

In this section, the various advancements made in the field of image restoration and colorization are discussed. Traditional methods that aimed to tackle this problem involved some degree of manual intervention or effort for picking the right colors and deblurring and sharpening images using image editing software. With the advent of deep learning techniques, these tasks were automated by training models to learn and generalize color features and restoration functions from a selected distribution of input–output paired images. With the advent of GANs, this was further automated by eliminating the need of a human to evaluate the loss function. Instead, the GAN architecture uses two components—the generator and the discriminator that work together to automatically ensure that the loss function is performing well to produce very realistic images.

For image restoration tasks, Wang et al. (2021) focus on restoring facial images only and apply restoration techniques to extract high-quality images from the corresponding low-quality image counterpart. They attempt to use pretrained face GANs as generative facial priors (GFP) to successfully retain image information and maintain quality. However, the model could not perform well when the degradations were severe or the facial images were of varied poses. One of the main reasons for this was the fact that the model was trained on synthetically degraded data instead of real-world degraded images. Tomosada et al. (2021) propose an image deblurring method using GANs that requires less computational complexity to train and makes use of discrete cosine transform (DCT). The model is trained on a customized dataset including images with large motion blurs and introduces a novel loss function that compares images in the frequency domain. The authors of Pan et al. (2021) utilized image priors from a GAN that was trained on a large dataset. Deep generative priors were employed to replace the missing semantics in the images. Scaling the training images and training an encoder that maps the target image to the latent code helped the architecture's overall performance. This made accurate image reconstruction possible, but it also presupposed known degradation. The model does not generalize well to unknown degradation and performed poorly on real-world blind restoration. The authors in Ledig et al. (2017) introduce a content loss driven by perceptual similarity rather than similarity in pixel space to restore the features of degraded images while training. The deep residual network of the GAN can successfully recover photo-realistic textures from heavily down-sampled images.

Image colorization is an unconstrained problem with no unique solution, and it has been studied as a challenging problem for over many decades. The authors in Thomas et al. (2018) develop a model to efficiently colorize black and white images and videos, and eliminate issues such as—inconsistent colorization within individual objects, undersaturation, and oversaturation of images. The model was developed on the premise that: neighboring image frames with neighboring pixels that have similar intensities should ultimately have similar colors. The authors used a long short-term memory (LSTM) for dealing with grayscale videos and convolutional neural networks (CNNs) for images. The authors in Sarmai and Sadhu (2022) compare the performance of convolutional autoencoders and GANs for the colorization of grayscale images. They observed that the images colorized by the autoencoders lost their sharpness and were blurry in nature, whereas those colorized by the GAN are nearly identical to their ground truth counterparts. This further emphasizes the choice of using GANs in the proposed model. Nazeri et al. (2018) proposed a GAN architecture that included one-sided label smoothing, batch normalization, and RELU activation function. Instead of the RBG color space, they used the CIE Lab color space, where L is the lightness channel and a and b are the color channels. They observed that this prevented any sudden variations in both color and brightness as in RGB. Xiao et al. (2019) were inspired by Cycle-GAN (Zhu et al. 2017) and the concept of image-to-image translation. Their method uses unpaired images for training and direct prediction of color in the RGB color space, which eases the process of training data collection. However, their model was only able to generate reasonable color instead of restoring original color for input grayscale images. The authors of

Mourchid et al. (2020) propose an additional discriminator in their GAN which works in the feature domain. Such an architecture is proposed so that the generator learns to produce high-frequency features instead of noise from the normal distribution. The model colorizes images realistically and requires less artifacts. As future work, they suggested a typical Resnet architecture to retain low-scale features making it shallow. Sankar et al. (2020) study the use of GANs for colorization using the perceptual loss function. The model was trained on the open source CIFAR-100 dataset. Additionally, they did a comparative study on the same dataset using three different models—the U-Net CNN, GAN, and GAN with perceptual loss. After evaluating the results and comparing the three models, the authors' concluded that the GAN with perceptual loss gave the best results for colorization. This further strengthens the choice of using the same loss function as an integral part of the proposed GAN architecture.

Therefore, in their own separate domains, brilliant research has been carried out in restoration as well as colorization of images. As a joint task very few researchers have been able to provide promising results. Compared with early deep learning methods like CNN and LSTM, autoencoders have outperformed in the tasks of image restoration. With the advent of GANs or generative adversarial networks, they have outperformed even autoencoders in both restoration and colorization. GANs can produce more realistic and better-quality images. Although computation in these methods is expensive, overhead is reduced as the loss function is learnt automatically. To conclude this section, it is identified that GAN-based image restoration and colorization has a wide scope, and can contribute significantly to this domain.

3 Proposed Methodology

Most existing systems focus solely on either restoration or colorization of images or on both sequentially. Moreover, limitations on colorization such as over/undersaturation and the inability to capture the different colors of intricate images have been a common recurring problem. The reason why previous algorithms and models for colorization didn't work remarkably well is because there was still a human involved in hand-coding a key step, which is evaluating whether or not the restored or colorized image "looked good." So over the last couple of years, generative adversarial networks (GANs) have increasingly been replacing vanilla U-Net architectures and autoencoders (Sarmai and Sadhu 2022) in order to solve this problem, as it has proven to effectively learn the loss function with the help of its discriminator that can evaluate the generated image and also ensure more realistic feature generation. The proposed system aims to utilize this advantage of GANs in order to both effectively restore and colorize degraded grayscale images within the same architecture. The proposed methodology includes the details of the GAN architecture, the preprocessing stage of the model, followed by the breakdown of the GAN into its two different components—the generator and the discriminator.

The foundation of GANs is built on a generator that learns to generate new images and a discriminator that learns to differentiate between artificial and original images.

A conditional setup is used in GANs, which ensures that the generator and discriminator are both dependent on extra data (such as class labels or information from other modalities). The ideal model is thus capable of learning multi-modal mapping from inputs to outputs by being fed various contextual data. This conditional setup is what is known as a conditional GAN (Zhao et al. 2019). The proposed algorithm follows that of the conditional GAN algorithm and involves two separate modules—the generator and the discriminator, that are trained simultaneously.

3.1 Preprocessing

As part of the preprocessing stage, the images from the dataset are first resized, following which they are converted from the RGB color space to the *L*a*b* color space (Nazeri et al. 2018). The L channel of the degraded image serves as the grayscale image and its *L*a*b* counterpart as the colored image. The images are then split into the train and test datasets before being fed into the generator.

3.2 Generator Module

The generator takes a degraded grayscale image as input and generates a restored, colorized version of the same. The generator consists of a U-Net architecture (Ronneberger et al. 2015) as seen in Fig. 1, which includes a series of encoders and decoders. In order to speed up the training, the proposed model utilizes the pretrained ResNet50 model (He et al. 2016) in the first few layers of the encoder. This ensures the model can identify features faster and take significantly less time for training. This is followed by six decoder blocks that upscale the image to give a three-channeled output or a colored image.

3.3 Discriminator Module

The discriminator primarily serves as a binary classifier and will classify the input image as being either real or fake. It consists of a series of convolutional layers as seen in Fig. 2 with “Relu” activation function, followed by max pooling layers. The final layer utilizes the sigmoid activation function in order to generate a value between 0 and 1, indicating whether the input image is classified as fake or real, respectively.

The final part of the model includes the loss functions used in the training of the generator and discriminator. The first loss function is the GAN loss or adversarial loss. This minimax loss is traditionally used to train the generator and discriminator in sync with one another. The discriminator uses the cross-entropy loss function for training. The generator utilizes the VGG-19 pretrained model (Simonyan and

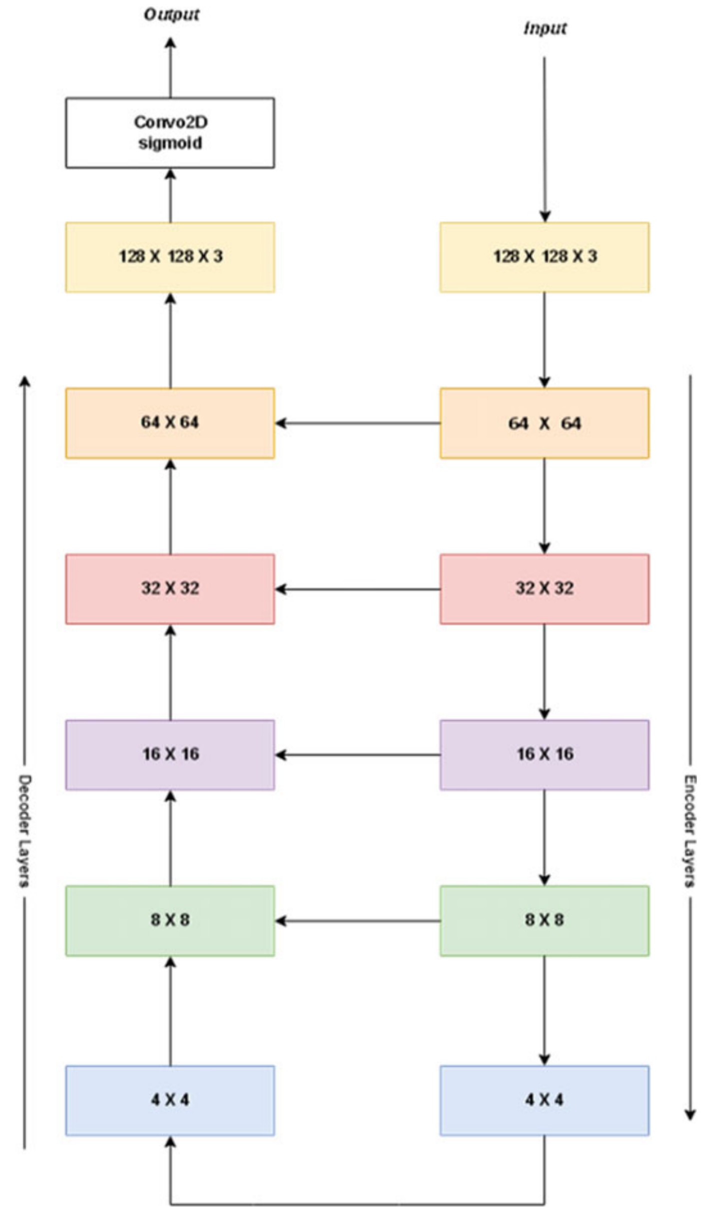


Fig. 1 Proposed generator architecture

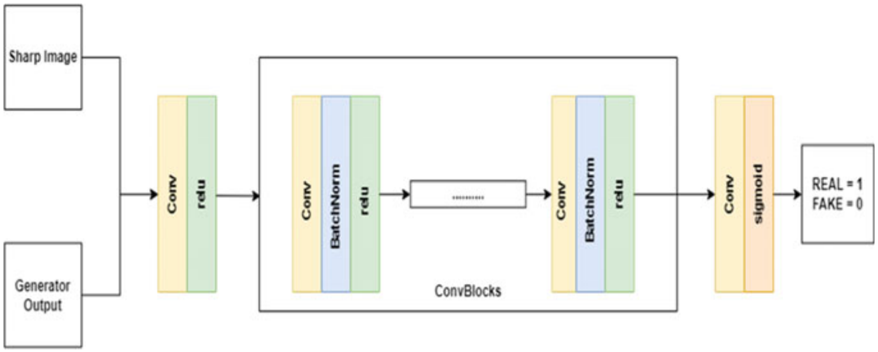


Fig. 2 Proposed discriminator architecture

Zisserman 2015) to obtain the feature-level representations of the generated image and the original image and performs a feature-wise comparison of the two. This loss function is known as the perceptual loss function and is used in combination with the L1 loss and the adversarial loss to update the generator's weights. The Adam optimizer is utilized with a learning rate of 0.005 for both the generator and the discriminator. Because the generator model and discriminator model are trained simultaneously in a zero-sum game, they are challenging to train. In other words, while one model improves, the other model suffers. Finding a point of equilibrium between the two conflicting concerns is the aim of training two models. It also implies that the nature of the optimization problem being tackled changes each time the parameters of one of the models are modified. As a result, a dynamic system is created. The technical difficulty of simultaneously training two competing neural networks is that they might not converge. Plotting the generator and discriminator's losses provides insight into the model's convergence.

4 Experimental Setup and Results

4.1 Dataset

The dataset required for the project includes images containing different classes of objects pertaining to nature so that the model is versatile and able to restore and colorize all kinds of scenery images. Colored, sharp images are expected from the dataset as they will be used as the ground truth images while training the model. Since GANs require huge amounts of data to be trained well, a relatively large and vast dataset is required. Previous work in this field implied that working with datasets having too many different classes might result in poor performance of the model. Hence, a dataset with a limited but varied number of classes is preferred, and for this reason a dataset of 5500 images is curated, consisting of images of colorful

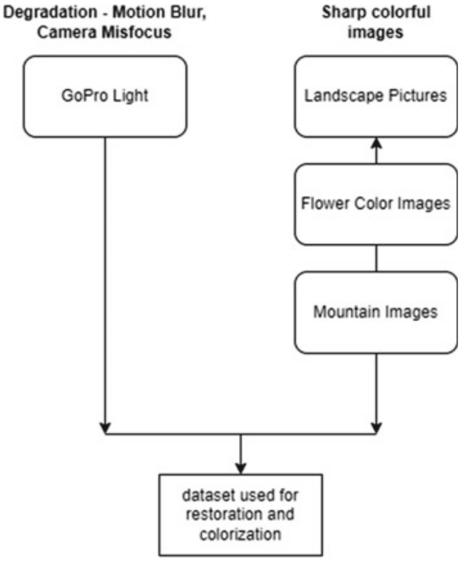

Degradation - Motion Blur, Camera Misfocus
Sharp colorful images
GoPro Light
Landscape Pictures
Flower Color Images
Mountain Images
dataset used for restoration and colorization

Fig. 3 Curation of dataset

scenery (Arnaud 2020), flowers (Belitskaya 2017) and mountains (https://www.kaggle.com/puneet6060/intel-image-classification) to focus on the colorization aspect of the model training. This was combined with the GoPro Light dataset (Tomosada et al. 2021) which consists of 2103 blurred and sharp image pairs for training. The images are captured by a high-speed camera. This allows the model to learn how to restore naturally degraded images affected by blur or motion blur.

The images of the newly curated dataset consists of 6840 training images and 763 testing images. They are resized to 128 × 128 and converted to grayscale as a part of the training phase of the model. As seen in Fig. 3, the final dataset used in training the model comprises of a combination of the GoPro Light dataset which consists of blur images and their clear counterparts, along with the manually curated dataset consisting of only nature-specific images.

4.2 Implementation

The proposed GAN architecture is implemented using tensorflow, keras. In total three models are implemented. The first model COLOR-GAN is the proposed architecture trained on just the colorization dataset and it performs no restoration. The input is a sharp grayscale image and the output is the colorized output. If the input is blurry then it doesn't perform any restoration on it and the colorization is not up to the mark. The second model RESTORE-GAN, uses the architecture trained only on the GoPro dataset and only performs restoration. The input is a blurry image and the output is the corresponding sharp image. This model performs well in terms of restoration, no matter the color of the input. Finally, the third model restore and colorize images GAN (RCI-GAN) uses the proposed architecture on the curated dataset. It takes as input a degraded (blurry) grayscale image and outputs the restored and appropriately colored image. Each of the models was trained for 100 epochs and tested on the test images from the GoPro light dataset (Tomosada et al. 2021). Figure 4 depicts the results of the COLOR-GAN, Fig. 5 depicts the results of RESTORE-GAN, and Fig. 6 depicts the results of RCI-GAN, respectively. The three models are evaluated based on image quality-specific metrics like peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM). The PSNR metric displays the relationship between a signal's maximum achievable power and the power of corrupting noise that compromises the accuracy of its representation. PSNR, a variant of MSE, focuses on pixel-by-pixel comparison. SSIM is related to the quality and perception of the human visual system (HVS color model). The SSIM represents picture distortion as a combination of three elements, namely loss of correlation, luminance distortion, and contrast distortion, as opposed to utilizing conventional error summation techniques. The PSNR and SSIM metrics for all three models are recorded on the test data and tabulated in Table 1.

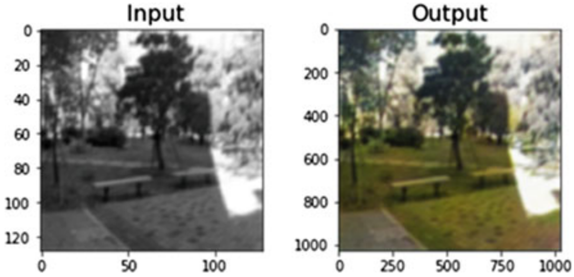


Fig. 4 COLOR-GAN output

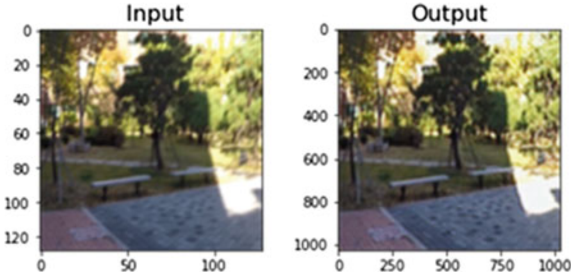


Fig. 5 RESTORE-GAN output

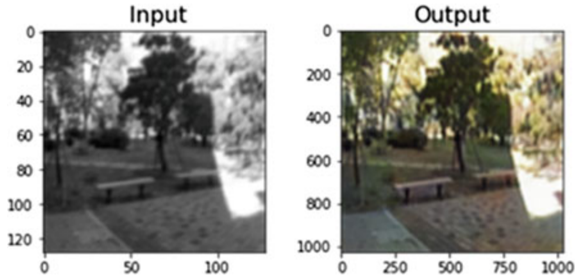


Fig. 6 RCI-GAN output

Table 1 Comparison of proposed models

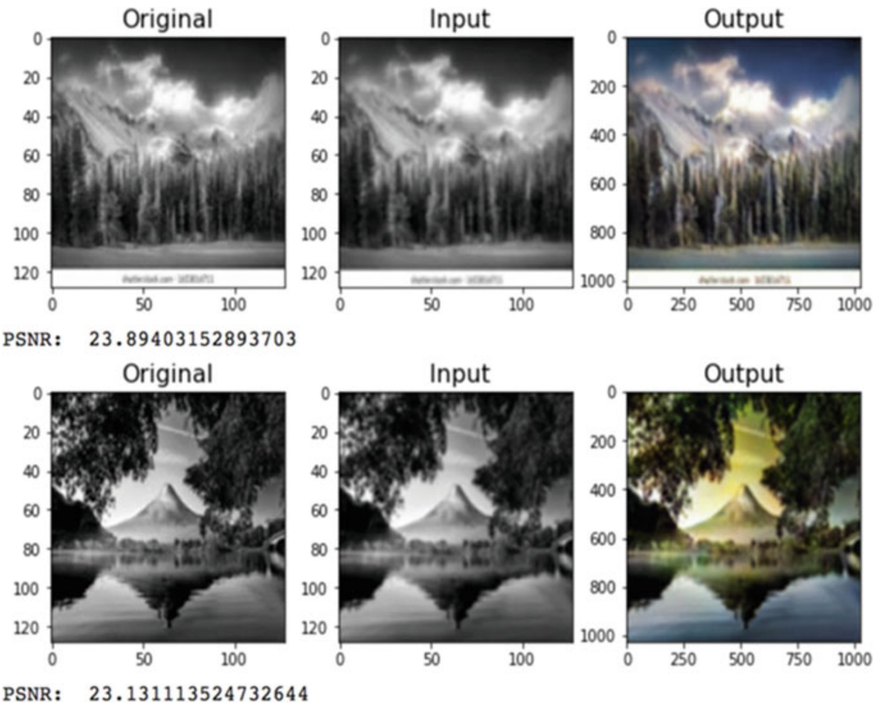
	PSNR	SSIM
COLOR-GAN	22.96	0.74
RESTORE-GAN	25.64	0.93
RCI-GAN	23.22	0.80

4.3 Results

The proposed RCI-GAN trained on the curated dataset is able to effectively remove motion blur and colorize both degraded as well as sharp grayscale images specific to the scenery class of images. As the model has been trained on a dataset containing images of nature and scenery, the model is fine-tuned to such images only at present. The model assigns reasonable color to the different features of the image, and is able to produce perceptually meaningful results. In terms of image quality and restoration, the model gives an average PSNR value of 23 and an average SSIM value of 0.8 when tested on real-world sample images. Figure 7 depicts the outputs obtained on testing the model on originally grayscale real-world samples. Figure 8 shows the outputs obtained on colored real-world samples that were converted to grayscale as a part of preprocessing, and Fig. 9 depicts the model's performance on predominantly green landscape images that were also converted to grayscale before being fed into the model.

4.4 Comparison of COLOR-GAN, RESTORE-GAN, and RCI-GAN Results

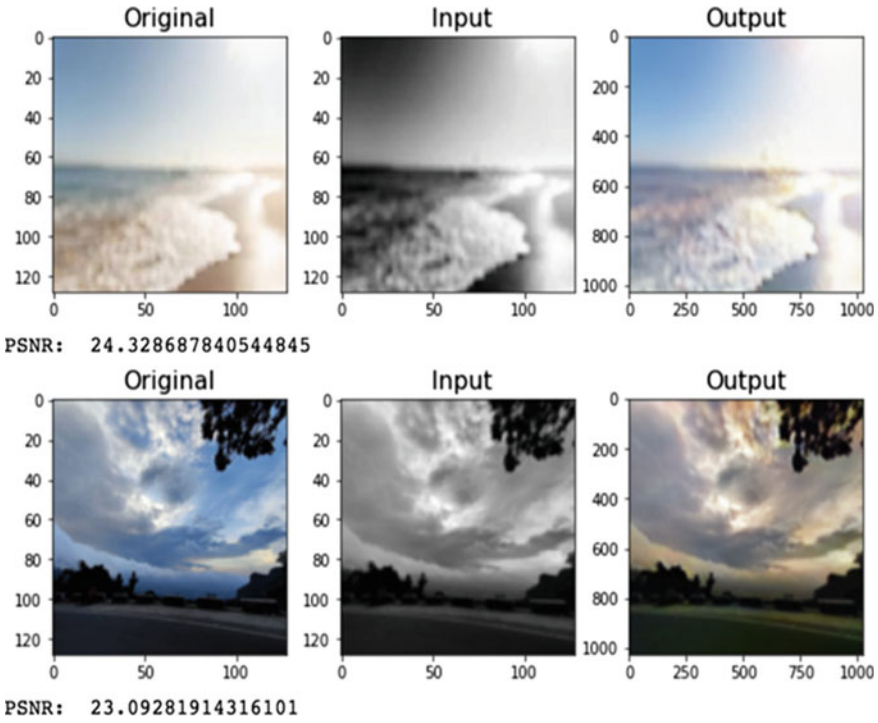
From Table 1 it can be concluded that when trained on the specific dataset, the proposed GAN architecture is able to achieve a good trade-off between image restoration and image colorization with acceptable evaluation metrics. The PSNR and SSIM values obtained for the RCI-GAN lie in between those obtained from COLOR-GAN and RESTORE-GAN, thereby proving that the proposed model can



```
PSNR:  23.89403152893703
PSNR:  23.131113524732644
```

Fig. 7 Visual results on real-world data

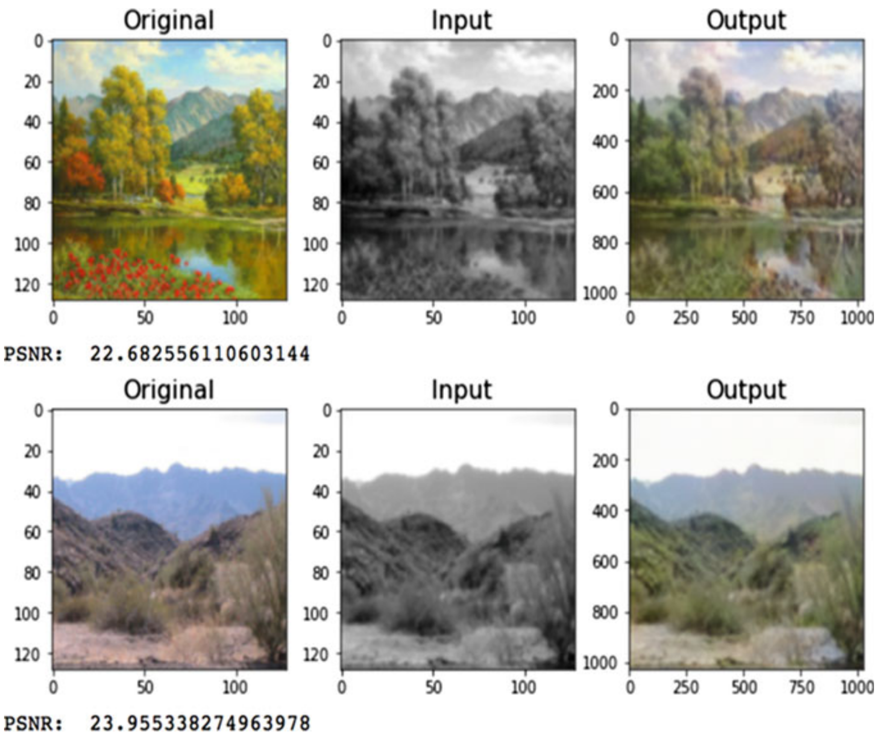
successfully combine both tasks within the same architecture without significant loss in performance.



```
PSNR:  24.328687840544845
```

```
PSNR:  23.09281914316101
```

Fig. 8 Visual results on real-world data



```
PSNR:  22.682556110603144
```

```
PSNR:  23.955338274963978
```

Fig. 9 Visual results on real-world data

4.5 *Comparison of RCI-GAN Results with Existing Models*

Evaluation metrics of the proposed model are compared with existing GAN models as stated in published research works. GFP GAN (Wang et al. 2021) performs blind face restoration and colorization, but is specific only to facial images. DeblurDCT GAN (Tomosada et al. 2021) and ESRGAN (Wang et al. 2018) perform deblurring and restoration of naturally blurred images but do not perform colorization. There are no models that do exactly what the proposed GAN offers and Table 2 provides a good comparison to evaluate results. It compares the PSNR and SSIM values of the proposed GAN with the existing models, performing either image restoration, colorization, or both. The model is successfully able to perform two tasks: image restoration and image colorization within a single GAN architecture, thereby reducing model complexity and eliminating the need for a second GAN. The RCI-GAN compares well with the evaluation metrics of existing models as depicted clearly in Table 2.

Table 2 Comparison of RCI-GAN with existing models

	PSNR	SSIM
RCI-GAN	23	0.8
GFP GAN	25	0.6
Deblur GAN	29	0.9
ESRGAN	23	0.6

5 Conclusion

While GAN architectures can be utilized in a wide variety of applications, it's rare to find a GAN architecture that can perform multiple tasks at once. This paper introduces a new avenue of research where a single GAN model can perform two tasks at the same time: image restoration and image colorization. The proposed restore and colorize images GAN (RCI-GAN) effectively restores and colorizes nature-specific images within a single GAN architecture. This eliminates the need for utilizing two separate GANs for both tasks individually, thereby presenting a novel architecture for training the GAN to focus on both tasks simultaneously. The model utilizes a customized dataset (GoPro, landscape, flowers, and mountains) and loss functions (L1 loss, adversarial loss, and perceptual loss) specific to both restoration and colorization to construct an architecture that encompasses the features of both image enhancement tasks. The image quality of the restored and colorized images have been calculated and compared with the existing models using metrics like peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) and shows promising results.

6 Limitations and Future Work

Figure 7 shows an image depicting the reflection of a mountain on a water body, surrounded by trees on the top left and right corners of the image. The model assigns the color green to the top part of the image and blue to the bottom part of the image. This is how the model interprets the image. However, the interpretation of the image could vary. Hence, one of the limitations include accurately colorizing the image according to human perception. Moreover, when the degradation is severe or the image sample is out of distribution, the model produces unsatisfactory results which could be improved by training the GAN on a larger class of images and for higher number of epochs. Loss graphs could be plotted at each instance of training to analyze and optimize convergence. Since the model produces good results on scenery/nature-specific images and generalizes well for real-world samples, it shows good potential for the model to be extended to all classes of images. It also paves the way for a unique area of research that can exploit GANs where two or more tasks can be achieved together.

References

Arnaud R (2020) Landscape pictures, version 1. https://www.kaggle.com/datasets/arnaud58/landscape-pictures. Accessed 2022

Belitskaya O (2017) The dataset of flower images, version 7. https://www.kaggle.com/code/olgabelitskaya/the-dataset-of-flower-images/data. Accessed 2022

Gampala V, Sunil Kumar M, Sushama C, Fantin Irudaya Raj E (2020) Deep learning based image processing approaches for image deblurring. Mater Today Proc. https://doi.org/10.1016/j.matpr.2020.11.076

Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2020) Generative adversarial networks. Commun ACM 63(11):139–144. https://doi.org/10.1145/3422622

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of IEEE conference computer vision pattern recognition (CVPR), pp 770–778

Intel image classification challenge, mountain class. https://www.kaggle.com/puneet6060/intel-image-classification

Ledig C, Theis L, Huszar F, Caballero J, Cunning-ham A, Acosta A, Aitken A, Tejani A, Totz J, Wang Z, Twitter WS (2017) Photo-realistic single image super-resolution using a generative adversarial network

Mourchid Y, Donias M, Berthoumieu Y (2021) Automatic image colorization based on multi-discriminators generative adversarial networks. In: 2020 28th European signal processing conference (EUSIPCO), pp 1532–1536. https://doi.org/10.23919/Eusipco47968.2020.9287792

Nazeri K, Ng E, Ebrahimi M (2018) Image colorization using generative adversarial networks. In: Perales F, Kittler J (eds) Articulated motion and deformable objects. AMDO 2018. Lecture notes in computer science, vol 10945. Springer, Cham. https://doi.org/10.1007/978-3-319-94544-6 9

Nguyen T, Mori K, Thawonmas R (2016) Image colorization using a deep convolutional neural network. arXiv preprint arXiv:1604.07904

Pan X, Zhan X, Dai B, Lin D, Loy CC, Luo P (2021) Exploiting deep generative prior for versatile image restoration and manipulation. In: IEEE transactions on pattern analysis and machine intelligence

Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention. Springer, Cham

Sankar R, Nair A, Abhinav P, Mothukuri SKP, Koolagudi SG (2020) Image colorization using GANs and perceptual loss. In: 2020 international conference on artificial intelligence and signal processing (AISP), pp 1–4. https://doi.org/10.1109/AISP48273.2020.9073284

Sarmai A, Sadhu A (2022) Image colorization using convolutional autoencoders and generative adversarial networks, vol 9. IARJSET, pp 86–99. https://doi.org/10.17148/IARJSET.2022.9115

Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: ICLR

Sophia SGG, Umar S, Visvas N (2022) An efficient method for blind image restoration using GAN. In: 2022 international conference on innovative computing, intelligent communication and smart electrical systems (ICSES), pp 1–8. https://doi.org/10.1109/ICSES55317.2022.9914234

Thomas BS, Dogra R, Dixit B, Raut A (2018) Automatic image and video colourisation using deep learning. In: 2018 international conference on smart city and emerging technology (ICSCET), pp 1–4. https://doi.org/10.1109/IC-SCET.2018.8537308

Tomosada H, Kudo T, Fujisawa T, Ikehara M (2021) GAN-based image deblurring using DCT loss with customized datasets. IEEE Access 9:135224–135233. https://doi.org/10.1109/ACCESS.2021.3116194

Wang X, Yu K, Wu S, Gu J, Liu Y, Dong C, Loy CC, Qiao Y, Tang X (2018) ESRGAN: enhanced super-resolution generative adversarial networks. ECCV Workshops

Wang X, Li Y, Zhang H, Shan Y (2021) Towards real-world blind face restoration with generative facial prior. In: 2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 9164–9174. https://doi.org/10.1109/CVPR46437.2021.00905

Xiao Y, Jiang A, Liu C, Wang M (2019) Single image colorization via modified cyclegan. In: 2019 IEEE international conference on image processing (ICIP), pp 3247–3251. https://doi.org/10.1109/ICIP.2019.8803677

Zhao JLL, Bai H, Zeng AWB, Zhao Y (2019) Simultaneous colordepth super-resolution with conditional generative adversarial networks. arXiv:1708.09105

Zhu J-Y, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision, pp 2223–2232

Novel Adaptive Learning Rate Back Propagation Neural Network-Based Online Rotor and Stator Resistance Estimator for Sensorless Induction Motor Drives

M. K. Ajithanjaya Kumar, Rajkiran Ballal, Sanath Saralaya, and J. Sathyendra Bhat

1 Introduction

Induction motors are the most commonly used motors in industries due to their enormous advantages over other classes of motors. Initially, DC motors were used widely in industries due to their simple speed control. The presence of commutators turned out this advantage and brought induction motors to the forefront of industries. Speed control is the main challenge associated with induction motors. The advancements in power electronics addressed most of the issues associated with speed control. Inverters are widely used for speed control, but the stability of inverter-fed induction moors is of greater concern to be addressed (Sengamai et al. 2022). The accurate rotor flux estimation and control is of the greater concerned issue in both direct and indirect vector control systems (Lascu et al. 2020).

Most of the industrial applications involving high-performance drive systems use indirectly field-oriented vector-controlled induction motor drives. The indirect field orientation control scheme is the feed-forward scheme which utilizes an inherent slip relation, hence sensitive to motor parameters. The accurate estimation of the rotor flux largely depends on the value of the rotor and stator resistances used for control.

M. K. Ajithanjaya Kumar (✉)
Nitte Meenakshi Institute of Technology, Bengaluru, India
e-mail: ajithk@sjec.ac.in

M. K. Ajithanjaya Kumar · S. Saralaya
EEE Department, St Joseph Engineering College, Mangaluru, India

R. Ballal
EEE Department, Nitte Meenakshi Institute of Technology, Bengaluru, India

J. S. Bhat
MCA Department, St Joseph Engineering College, Mangaluru, India

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_41

The accuracy of the speed estimation depends on the estimated rotor resistance (Karanayil et al. 2007). During the operation of the induction motor drives, stator and rotor resistances can increase up to 100% as a result of temperature changes. The resistance estimation is very difficult through temperature sensors or thermal models, especially for induction motor drives (Tuan et al. 2020).

Several algorithms for rotor and stator resistance estimation have been proposed in the past. A few of the conventional approaches are: based on model reference adaptive system (MRAS) (Basnet and Lee 2018), extended Kalman filters (EKF) (Outhrouche 2002; Lin and Wan 2012), sliding mode control (Tak et al. 2017), fuzzy logic technique (Karanayil et al. 2003), and Luenberger observer (Jouli et al. 2015). The application of artificial neural networks (ANNs) for the identification and control of nonlinear dynamic systems in power electronic systems and AC drives has been increased as a wide range of nonlinear functions can be approximated with a high degree of accuracy by ANNs.

Accurate knowledge of machine parameters is needed for accurate speed estimation and also for stable operation. In this paper, an ANN-based model reference adaptive approach is proposed for stator and rotor resistance. In the proposed approach, two models of the state variable estimations are used, one model gives the actual states of the induction motor and another gives the neural network model states as shown in Fig. 1. The error obtained between the actual state variable and estimated state variable is back propagated by adjusting the weights of the neural network model in such a way that output of neural network model traces the actual output. Two-layer ANN with a back propagation algorithm with an adaptive learning rate and added momentum is proposed in this paper for the faster response of the system.

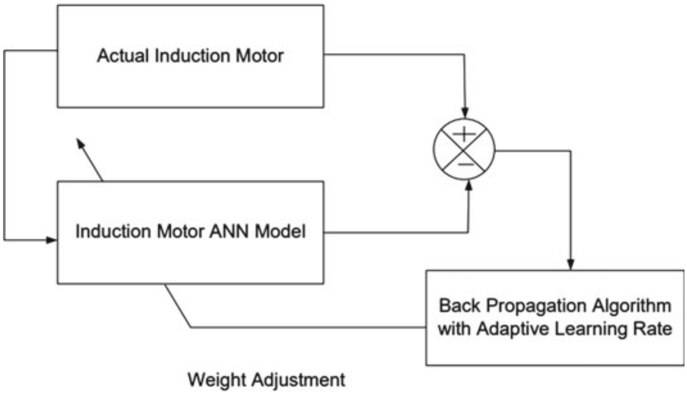

Actual Induction Motor
+
−
Induction Motor ANN Model
Back Propagation Algorithm with Adaptive Learning Rate
Weight Adjustment

Fig. 1 ANN-based model reference adaptive system for parameter identification

2 ANN-Based Rotor Resistance Estimator

2.1 Voltage Model of Induction Motor

From the stationary reference frame model of the induction model, considering the stator side of the equivalent circuit, d, and q-axis rotor flux equations are obtained as (Sengamai et al. 2022)

$$\begin{bmatrix} \dot{\psi}^s_{dr} \\ \dot{\psi}^s_{qr} \end{bmatrix} = \frac{L_r}{L_m}\left\{\begin{bmatrix} v^s_{ds} \\ v^s_{qs} \end{bmatrix} - R_s\begin{bmatrix} i^s_{ds} \\ i^s_{qs} \end{bmatrix} - \sigma L_s\begin{bmatrix} \frac{\mathrm{d}}{\mathrm{d}t} i^s_{ds} \\ \frac{\mathrm{d}}{\mathrm{d}t} i^s_{qs} \end{bmatrix}\right\}, \tag{1}$$

where $\sigma = 1 - \frac{L_m^2}{L_s L_r}$.

2.2 Adaptive Model—Current Model

From the stationary reference frame model of the induction model, considering the rotor side of the equivalent circuit, d and q-axis rotor flux equations are obtained as (Lascu et al. 2020)

$$\overrightarrow{\frac{d}{dt}\psi_r^{s,im}} = \left[-\frac{1}{T_r}I + \omega_r J\right]\overrightarrow{\psi_r^{s,im}} + \frac{L_m}{T_r}\overrightarrow{i^s_{dq}}, \tag{2}$$

where rotor time constant $T_r = \frac{L_r}{R_r}$

$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$; $J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$; $\overrightarrow{i^s_{dq}} = \begin{bmatrix} i^s_{ds} \\ i^s_{qs} \end{bmatrix}$; $\overrightarrow{\psi_r^{s,im}} = \begin{bmatrix} \psi^{s,im}_{dr} \\ \psi^{s,im}_{qr} \end{bmatrix}$; $\overrightarrow{\psi_r^{s,vm}} = \begin{bmatrix} \psi^{s,vm}_{dr} \\ \psi^{s,vm}_{qr} \end{bmatrix}$.

The sample data model of Eq. (2) is

$$\overrightarrow{\psi_r^{s,nm}(k)} = (W_1 I + W_2 J)\overrightarrow{\psi_r^{s,nm}(k-1)} + W_3\overrightarrow{i^s_{dq}} \tag{3}$$

$W_1 = 1 - \frac{T_s}{T_r}$; $W_2 = \omega_r T_s$; $W_3 = \frac{L_m}{T_r} T_s$.

T_s—the sampling time.

Equation (3) can be written as

$$\overrightarrow{\psi_r^{s,nm}} = W_1 X_1 + W_2 X_2 + W_3 X_3 \tag{4}$$

where $X_1 = \begin{bmatrix} \psi^{s,nm}_{dr}(k-1) \\ \psi^{s,nm}_{qr}(k-1) \end{bmatrix}$; $X_2 = \begin{bmatrix} \psi^{s,nm}_{qr}(k-1) \\ \psi^{s,nm}_{dr}(k-1) \end{bmatrix}$; $X_3 = \begin{bmatrix} i^s_{ds}(k-1) \\ i^s_{qs}(k-1) \end{bmatrix}$.

Equation (4) can be represented by the two-layered neural network model as shown in Fig. 2. X_1, $X_{2,}$ and X_3 are the inputs and W_1, $W_{2,}$ and W_3 are the weights

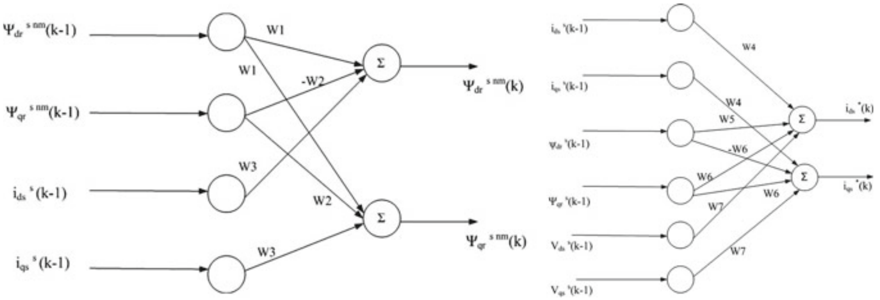


Fig. 2 Implementation of current model for rotor flux and stator flux estimation by two-layered neural networks

of the network. Since W_2 is independent of rotor time constant, only W_1 and W_3 are to be updated for the estimation of rotor resistance R_r.

The square function of the rotor flux error between the reference model and adaptive model is expressed as

$$E_1 = \frac{1}{2}\varepsilon_1^2(k) = \frac{1}{2}\left\{\overrightarrow{\psi_r^{s,vm}} - \overrightarrow{\psi_r^{s,nm}}\right\}^2. \tag{5}$$

The weights W_1 and W_3 are updated during training as

$$W_1(k) = W_1(k-1) + \eta_1 \Delta W_1(k) \tag{6}$$

$$W_3(k) = W_3(k-1) + \eta_3 \Delta W_3(k). \tag{7}$$

The weight adjustment factors $\Delta W_1(k)$ and $\Delta W_3(k)$ are obtained from the error of the rotor flux as

$$\Delta W_1(k) = -\frac{\partial E_1}{\partial W_1} = \left[\overrightarrow{\psi_r^{s,vm}}(k) - \overrightarrow{\psi_r^{s,nm}}(k)\right]^T \overrightarrow{\psi_r^{s,nm}}(k-1) \tag{8}$$

$$\Delta W_3(k) = -\frac{\partial E_1}{\partial W_3} = \left[\overrightarrow{\psi_r^{s,vm}}(k) - \overrightarrow{\psi_r^{s,nm}}(k)\right]^T \overrightarrow{i_{dq}^{s}}(k-1). \tag{9}$$

η_1 and η_2 are the learning rates which are selected based on trial-and-error process. The learning rate may be maintained constant throughout the training of the neural network (Lascu et al. 2020).

The constant learning rates can be replaced by adaptive learning rate to obtain fast converges and reduced estimated resistance error. The weights W_1 and W_3 can be updated to reduce the error as a function $\varepsilon_i(k) = \Delta W_i(k)\Delta W_i(k-1)$, where $\Delta W_i(k)$ and $\Delta W_i(k-1)$ are the weight adjustment factors during k and $(k-1)$ iteration during learning (Tuan et al. 2020). The learning rate is updated based on the function $\varepsilon_i(K)$. The learning rate for the kth iteration during learning can be estimated by adopting the Sigmoid Function:

$$f(\varepsilon_i(k)) = \frac{\alpha_i}{1 + e^{-s*\varepsilon_i(k)}}. \tag{10}$$

s is the steepness parameter which may vary between − 1 and + 1. α_i is the positive constant.

The learning rates are updated for kth iteration as

$$\eta_i(k) = \eta_i(k-1) * [1 + f(\varepsilon_i(k))]. \tag{11}$$

The learning rate for kth iteration is updated as per Eq. (11).

The learning rate obtained as in Eq. (11) is used to update the weights W_1 and W_3. The rotor resistance R_r can be determined either considering W_1 or W_3 as given by Eq. (12) or (13) as

$$R_r = \frac{L_r W_3}{L_m T_s} \tag{12}$$

$$R_r = \frac{L_r(1 - W_1)}{T_s}. \tag{13}$$

2.3 Training Neural Network with Momentum

To avoid struck at local minima during training, pre-selected positive constant momentum term can be included with the weight adjustment terms. The weights W_1 and W_3 are updated during training as

$$W_1(k) = W_1(k-1) + \eta_1 \Delta W_1(k) + \beta_1 \Delta W_1(k-1) \tag{14}$$

$$W_3(k) = W_3(k-1) + \eta_3 \Delta W_3(k) + \beta_2 \Delta W_3(k-1). \tag{15}$$

The β_1 and β_2 are the pre-selected positive constant whose value lies between 0 and 1.

3 ANN-Based Stator Resistance Estimator

The d^s and q^s axes currents of the Induction motor can be represented by considering the voltage model and current model equations considered for rotor resistance estimation as

$$\frac{d}{dt}i_{ds}^{s} = \frac{L_m}{\sigma L_s L_r T_r}\psi_{dr}^{s} + \frac{L_m}{\sigma L_s L_r}\omega_r \psi_{qr}^{s} - \frac{L_m^2}{\sigma L_s L_r T_r}i_{ds}^{s} + \frac{1}{\sigma L_s}v_{ds}^{s} - \frac{1}{\sigma L_s}R_s i_{ds}^{s} \tag{16}$$

$$\frac{d}{dt}i_{qs}^{s} = \frac{L_m}{\sigma L_s L_r T_r}\psi_{qr}^{s} - \frac{L_m}{\sigma L_s L_r}\omega_r \psi_{dr}^{s} - \frac{L_m^2}{\sigma L_s L_r T_r}i_{qs}^{s} + \frac{1}{\sigma L_s}v_{qs}^{s} - \frac{1}{\sigma L_s}R_s i_{qs}^{s}. \tag{17}$$

Expressing Eq. (16), (17) in discrete form,

$$i_{ds}^{s*}(k) = W_4 i_{ds}^{s}(k-1) + W_5 \psi_{dr}^{s}(k-1) + W_6 \psi_{qr}^{s}(k-1) + W_7 v_{ds}^{s}(k-1) \tag{18}$$

$$i_{qs}^{s*}(k) = W_4 i_{qs}^{s}(k-1) + W_5 \psi_{qr}^{s}(k-1) - W_6 \psi_{dr}^{s}(k-1) + W_7 v_{qs}^{s}(k-1) \tag{19}$$

where $W_4 = \left[1 - \left(\frac{T_s}{\sigma L_s}\right)\left(\frac{L_m^2}{L_r T_r}\right) - \left(\frac{T_s}{\sigma L_s}\right)R_s\right]$; $W_5 = \left(\frac{T_s}{\sigma L_s}\right)\left(\frac{L_m}{L_r T_r}\right)$; $W_6 = \left(\frac{T_s}{\sigma L_s}\right)\left(\frac{L_m}{L_r}\right)\omega_r$; $W_7 = \frac{T_s}{\sigma L_s}$.

Equations (18) and (19) can be represented by two-layered neural networks. The weights W_5, W_6 and W_7 are determined from machine parameters, speed ω_r, and sampling interval T_s. As W_4 only depends on stator resistance, it is the adjustment weight during the training of the neural network. The neural network is trained by a standard back propagation algorithm.

The error square function is given by

$$E_2 = \frac{1}{2}\overrightarrow{\varepsilon}_2^{2}(k) = \frac{1}{2}\left\{i_s(k) - i_s^*(k)\right\}^2. \tag{20}$$

Weight W$_4$ is updated during training as (Karanayil et al. 2007)

$$W_4(k) = W_4(k-1) + \eta_4 \Delta W_4(k), \tag{21}$$

where η_4 is the learning rate. The $\Delta W_4(k)$ is the weight adjustment factor which is obtained as

$$\Delta W_4(k) = -\frac{\partial \varepsilon_2}{\partial W_4} = \left[\overrightarrow{i_s(k)} - \overrightarrow{i_s^*(k)}\right]^T - \overrightarrow{i_s^*(k-1)}. \tag{22}$$

The learning rate η_4 is selected based on the trial-and-error process. The learning rate can be maintained constant throughout the training of the neural network (Tuan et al. 2020) or the constant learning rate can be replaced by adaptive learning rate. The learning rate can be updated during the learning as

$$\eta_i(k) = \eta_i(k-1) * [1 + f(\varepsilon_i(k))], \tag{23}$$

where $f(\varepsilon_i(k))$ can be determined by adapting the activation function as explained for the rotor resistance estimation, in Sect. 2.

The stator resistance can be calculated by considering the adjusted weight W_4 as

$$R_s = \left\{1 - W_4 - \frac{T_s}{\sigma L_s}\frac{L_m^2 R_r}{L_r^2}\right\}\frac{\sigma L_s}{T_s}. \tag{24}$$

With the added momentum, W_4 is updated during training as

$$W_4(k) = W_4(k-1) + \eta_{41}\Delta W_4(k) + \beta_3 \Delta W_4(k-1). \tag{25}$$

The β_3 is the pre-selected positive momentum constant, which lies between 0 and 1.

4 Results and Analysis

The block diagram of the induction motor drive with online rotor and stator resistance estimation is shown in Fig. 3. The entire system is simulated using MATLAB/ Simulink software (R2022a, USA). The induction motor parameters are obtained from no load and blocked rotor tests performed in the laboratory. Induction motor parameter details are given in Table 1.

The performance parameters of the motor are determined for the slip of 0.2466 at which the maximum electromagnetic torque of 68.82 N-m is obtained. The proposed back propagation neural network algorithm for stator and rotor resistance estimation is implemented using MATLAB programming with different conditions as constant learning rate with and without momentum, and adaptive learning rate using binary sigmoid function.

The d and q axes rotor flux obtained from the voltage and current model, d and q axes currents of the stator reference frame induction motor model are sampled with sampling time T_s of 2 ms as shown in Figs. 4 and 5, respectively.

The sampled values are given as the input to the MATLAB program of the proposed back propagation neural network algorithm. The sampled values are obtained for actual stator and rotor resistance values, 25, 50, 75, and 100% increase in the respective resistances. The developed neural network is trained to estimate the resistances. Both stator and rotor resistances for all the above mentioned are estimated with conventional neural network back propagation algorithm having constant learning rate without and with added momentum, back propagation algorithm having adaptive learning rate implemented through binary sigmoid function without and with added momentum.

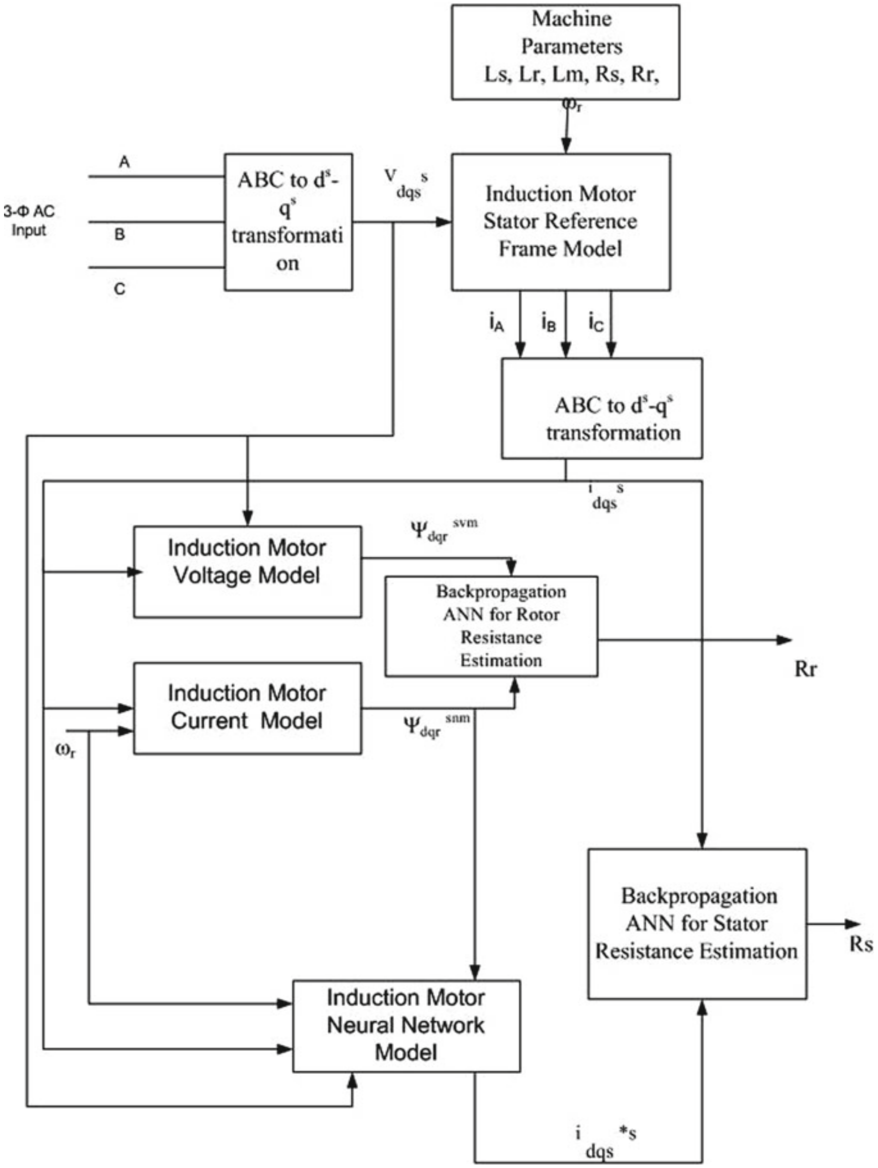

Machine Parameters Ls, Lr, Lm, Rs, Rr, ω_r
3-Φ AC Input
A
B
C
ABC to d^s-q^s transformation
V_{dqs}^{s}
Induction Motor Stator Reference Frame Model
i_A
i_B
i_C
ABC to d^s-q^s transformation
i_{dqs}^{s}
Induction Motor Voltage Model
Ψ_{dqr}^{svm}
Backpropagation ANN for Rotor Resistance Estimation
Rr
Induction Motor Current Model
Ψ_{dqr}^{snm}
ω_r
Backpropagation ANN for Stator Resistance Estimation
Rs
Induction Motor Neural Network Model
i_{dqs}^{*s}

Fig. 3 Block diagram of induction motor drive for online rotor and stator resistance estimation

Table 1 Induction motor parameter details

Sl No.	Parameters	Values
1	Rated power	3.68 kW
2	Rated voltage	415 V
2	Rated current	7.5 A
3	Rated frequency	50 Hz
4	No load speed	1498 RPM
5	Number of poles	4
6	Type of stator connection	Delta
7	Stator resistance	5.7 Ω
8	Rotor resistance	4.11 Ω
9	Stator reactance (L_s)	0.5634 H
10	Rotor reactance ($L_r R_r$)	0.5634 H
11	Magnetizing inductance (L_m)	0.5379 H

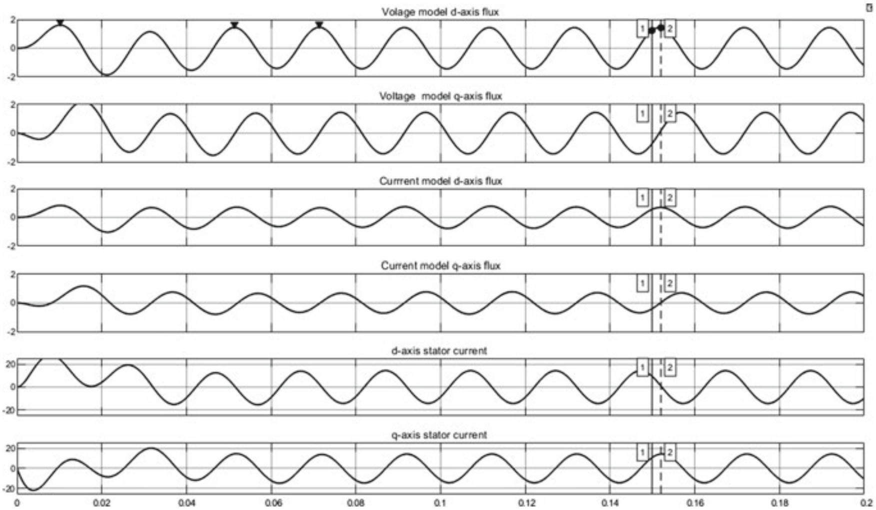


Fig. 4 *d*, *q* axes rotor flux from voltage and current model, *d*, *q* axes stator current from stator reference frame induction motor model

4.1 Rotor Resistance Estimation

The percentage error and numbers of iterations for the convergence of error to minimum with all the cases of back propagation algorithm are presented in Table 2.

The simulation results obtained for weight adjustment factors of W_1, W_3 with iterations during the learning process for all four back propagation algorithms are shown in Figs. 6, 7, 8, and 9. In case of conventional algorithm, learning rate for the weight W_1 and W_3 is maintained constant at 0.015 and 40e−6, respectively,

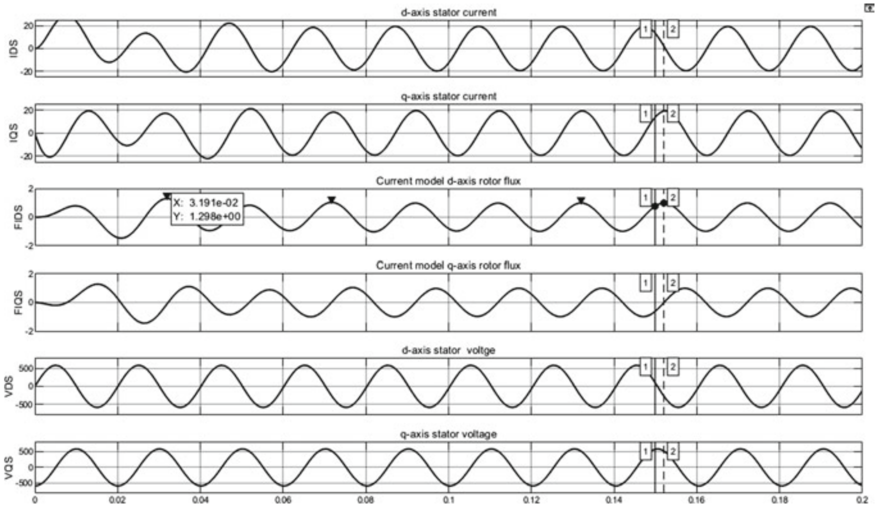


Fig. 5 *d* and *q* axes stator currents from stator reference frame induction motor model and from current model, *d* and *q*-axis stator voltage waveforms

throughout the learning process. As the learning rate is constant throughout the learning process, the weight adjustment factors of W_1 and W_3 decrease by a small amount during successive iteration which leads to small changes in weights W_1 and W_3 in each step during learning. The large number of learning steps is needed for the convergence of the error to the minimum value.

With added momentum, weight adjustments factors of W_1 and W_3 change at higher rate and are nearly double that of without momentum. Thus, both W_1 and W_3 change at higher rate. The error convergences to minimum with reduced number of iterations. With adaptive learning rate using sigmoid function, as the learning rates of W_1 and W_3 increase during training, from the initial values of 0.015 and 40e−6, respectively, weight adjustment factors of W_1 and W_3 increase at higher rate corresponding to that of conventional algorithm. The error convergences to minimum with reduced number of iterations as compared to conventional algorithm. The added momentum in addition to the increasing learning rates gives weight adjustments at larger rate during learning, evident in the results obtained from simulations as shown in Fig. 9, resulting in quick estimation of resistance.

Table 2 Percentage error obtained in the estimation of rotor resistance, number of iterations (i) for the convergence of error to minimum

Actual rotor resistance	Conventional-without momentum		Conventional-with momentum		Sigmoid-without momentum		Sigmoid-with momentum	
	% Error	No. of iterations	% Error	No. of iterations	% Error	No. of iterations	% Error	No. of iterations
4.11	0	0	0	0	0	0	0	0
5.1375	– 0.397	16	– 1.679	9	– 3.168	13	0.727	2
6.165	0.334	19	– 0.141	10	2.621	14	– 1.750	3
7.1925	– 0.949	23	– 1.464	12	1.974	17	– 1.270	4
8.22	– 6.307	30	– 8.535	16	– 2.222	22	0.417	5

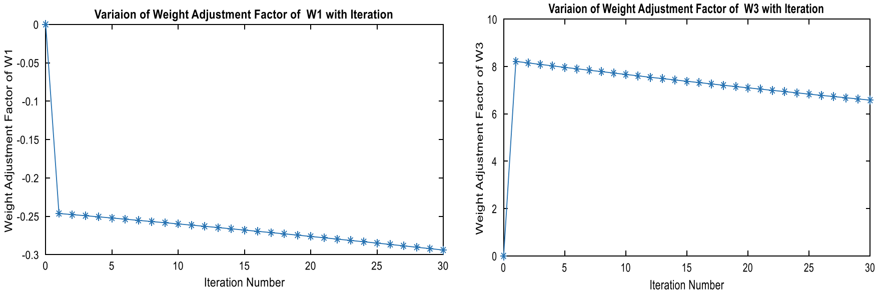


Fig. 6 Variations of weight adjustment factors W_1 and W_3 with conventional back propagation algorithm

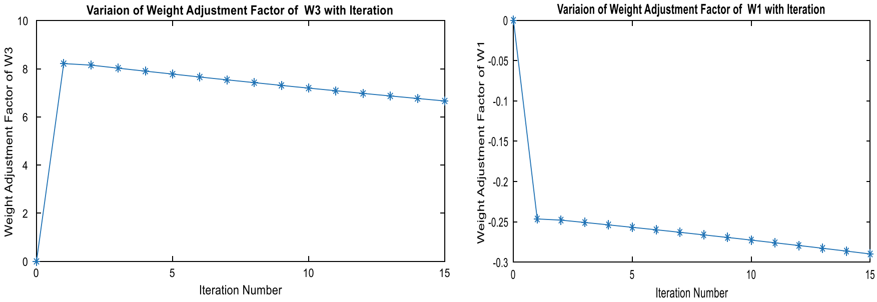


Fig. 7 Variations of weight adjustment factors for W_1 and W_3 with iteration for conventional back propagation algorithm with added momentum

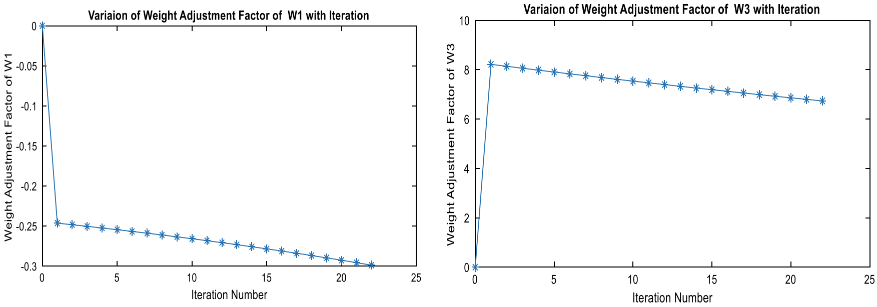


Fig. 8 Variations of weight adjustment factors for W_1 and W_3 with iteration having adaptive learning rate by sigmoid function

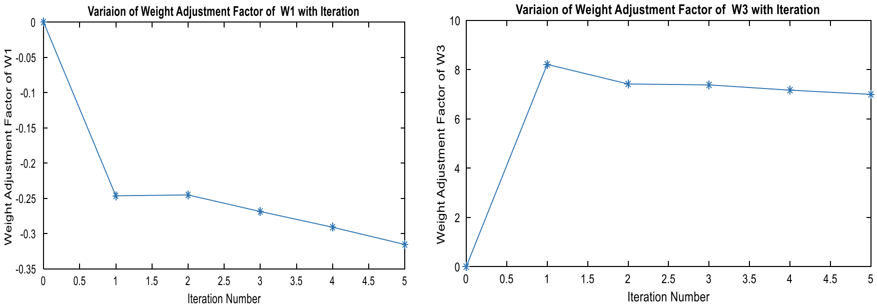


Fig. 9 Variations of weight adjustment factors for W_1 and W_3 with iteration by sigmoid function with added momentum

4.2 Stator Resistance Estimation

The percentage error with number of iterations required for the convergence of error to minimum for all the cases of back propagation algorithms are presented in Table 3.

The simulation results obtained for variation in weight adjustment factor of W_4 with iterations during the learning process for all the four back propagation algorithms are shown in Figs. 10, 11, 12 and 13. The learning rate of 0.00015 and momentum of 0.00025 are obtained by training the network. In the conventional algorithm, learning rate for the weight W_4 is maintained constant at 0.00015. Since the learning rate is constant, weight adjustment factor of W_4 increases in small step during the learning resulting in large number of iterations for the convergence of error to the minimum level. The added momentum with constant learning rate leads to faster convergence of error to the minimum level, resulting in reduced number of iterations. With adaptive learning rate using sigmoid functions, the number of iterations required remains same as that with conventional algorithm but reduces the error in estimation. The added momentum during learning with adaptive learning rate makes the weight adjustment factor to rise at faster rate leading to less number of iterations with reduced error in estimation of the resistance.

Table 3 Percentage error obtained in estimation of stator resistance and number of iterations for the convergence of error

Actual stator resistance	Conventional-without momentum		Conventional-with momentum		Sigmoid-without momentum		Sigmoid-with momentum	
	% Error	No. of iterations	% Error	No. of iterations	% Error	No. of iterations	% Error	No. of iterations
5.7	0	0	0	0	0	0	0	0
7.125	− 2.0368	4	− 5.2407	2	− 2.1080	4	− 1.9087	2
8.55	− 0.1543	7	− 2.7941	3	− 0.3614	7	− 0.7836	3
9.975	1.3518	10	− 0.7177	4	0.9363	10	0.4130	4
11.4	2.1763	13	0.7403	5	1.6640	13	1.3482	5

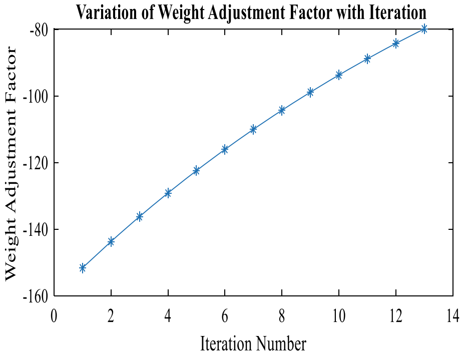


Fig. 10 Variations of weight adjustment factor with iteration for conventional algorithm

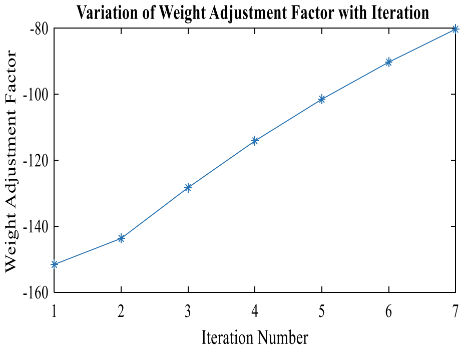


Fig. 11 Variations of weight adjustment factor with iteration for conventional algorithm with added momentum

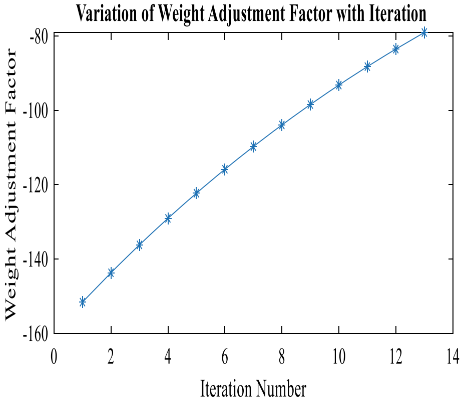
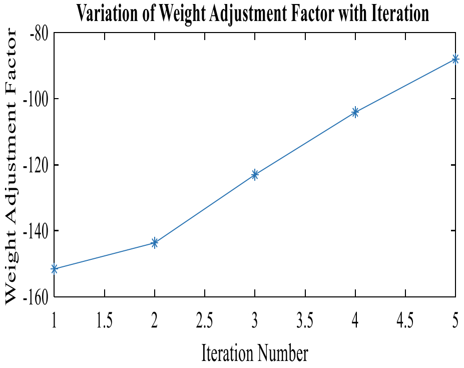


Fig. 12 Variations of weight adjustment factor with iteration for adaptive learning rate algorithm

Fig. 13 Variations of weight adjustment factor with iteration for adaptive learning rate with added momentum



5 Conclusion

The modified back propagation algorithm for the online estimation of stator and rotor resistances for sensorless induction motor drives is proposed. The performance of the proposed back propagation algorithm for the feed-forward neural network with adaptive learning rate determined by sigmoid function and added momentum is compared with conventional algorithms by simulation. The results of the simulation show that the proposed rotor and stator online resistance estimator tracks the resistances more closely and quickly as compared with the conventional back propagation neural network-based resistance estimators and with adaptive learning rate-based back propagation neural network without added momentum.

References

Basnet B, Lee HH (2018) Rotor flux estimation for low speed induction motor sensorless drives with MRAS. Lecture notes in electrical engineering. Springer, Singapore, pp 509–518

Jouli M, Agrebi Y, Koubaa Y, Bousak MA (2015) Luenberger state observer for simultaneous estimation of speed and stator resistance in sensorless IRFOC induction motor drives. In: 2015 16th international conference on sciences and techniques of automatic control and computer engineering (STA), Monastir, Tunisia

Karanayil B, Fazlur M, Grantham C (2007) Online stator and rotor resistance estimation scheme using artificial neural networks for vector controlled speed sensorless induction motor drive. IEEE Trans Ind Electron 54(1):2007

Karanayil B, Rahman MF, Grantham C, "Stator and Rotor Resistance Observer for Induction Motor Drive Using Fuzzy Logic and Artificial Neural Networks", 38th IAS Annual Meeting on Conference Record of the Industry Applications Conference, 2003.

Lascu C, Argeseanu A, Blaabjerg F (2020) Supertwisting sliding mode direct torque and flux control of induction machine drives. IEEE Trans Power Electron 35(5):5057–5065

Lin G, Wan Q (2012) Estimation of rotor resistance of induction motor based on extended Kalman filters. Advances in intelligent and soft computing. Springer, Berlin, Germany, pp 193–198

Outhrouche MA (2002) Estimation of speed, rotor flux and rotor resistance in cage induction motor using the EKF algorithm. Int J Power Energy Syst 22:103–109

Sengamai U, Abazhagan G, Thamizh Thetral TM, Vishnuram P, Khurshaid T, Kamel S (2022) Three phase induction motor drive: a systematic review on dynamic modelling, parameter estimation, and control schemes. Energies 15:8260

Tak R, Kumar SY, Rajpurohit BS (2017) Estimation of rotor and stator resistance for induction motor drives using second order sliding mode controller. J Eng Sci Technol Rev 9–15

Tuan PV, Tien DV, Leonowicz Z, Jasinki M, Sikorski T, Chakrabarti P (2020) Online rotor and stator resistance estimation based on artificial neural network applied in sensorless induction motor drive. Energies 13:4946

An Efficient *K*NN Algorithm for the Mental Health Performance Assessment Using *K*-means Clustering

Ravita Chahar, Ashutosh Kumar Dubey, and Sushil Kumar Narang

1 Introduction

Mental health plays a pivotal part in one's overall health (Rashid and Calhoun 2020). Mental health comprises of a person's all aspects of well-being like emotional, cerebral, and societal. All across the world, people have mental health issues, which can include personality changes, mood swings, and an inability to handle stress or daily challenges, creating voids from friends and maintaining distance in various activities, and more. In 2010, mental health issues accounted for the majority of years lived with disabilities worldwide, with conditions like anxiety and depression being among the most common disorders (Garcia-Ceja et al. 2018). In order to forecast any health-related anomalies, it is crucial to keep track of the mental health assessments of various groups. The population can be largely divided into three groups: high school students, college students, homemakers, and working adults. It is a widely held belief that stress and sadness affect all demographic groups equally (Srividya et al. 2018). Due to lack of public awareness, various people who endure mental disease are not able to get proper societal support (Mannarini and Rossi 2019). The techniques provided by artificial intelligence and the various models of machine learning in medical research may be viewed as a new paradigm in recent years. However, only one significant group of the population is addressed by these methods, which tend to corroborate clinical ideas rather than generate wholly new knowledge (Glaz et al. 2021). People suffering from mental illness are living under the constant dominance of social media. As per various studies, social media posts

R. Chahar · A. K. Dubey (✉) · S. K. Narang
Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India
e-mail: ashutosh.dubey@chitkara.edu.in

R. Chahar
e-mail: ravita.chahar@chitkara.edu.in

S. K. Narang
e-mail: sushilk.narang@chitkara.edu.in

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_42

available on different social sites such as Twitter and Reddit can be used to diagnose mental illness of the person's mental disease (Thorstad and Wolff 2019; Hollis et al. 2018; Low et al. 2020). Leveraging the rise of collection of data related to health by utilizing sensor devices (such as webcams, wearable devices, smartphones) together with advancements in machine learning technology is one promising way to improve the objectivity of psychiatric diagnoses and the accessibility of services. Numerous mental diseases, including psychotic, depressive, and anxiety disorders, have been identified in text taken from recorded audios, blogs, or social media using morphological, syntactic, semantic, and discursive aspects (Magson et al. 2021; Kilbourne et al. 2018; Salari et al. 2020; Rashmi and Shantala 2022). Thus, these technologies offer chances for improved mental health assessment (Low et al. 2020). Worldwide social isolation as a result of the COVID-19 virus quarantines is detrimental to mental health. As per World Health Organization, depression is among the most prevalent behavioral disorders that are connected to low demeanor, lack of interest, remorse, and valuelessness, as well as problems with sleep and eating, low energy, and concentration difficulties. Depression is also one of the major mental illnesses (Salari et al. 2020; Rashmi and Shantala 2022; Levis et al. 2019).

There are several researchers which are already working in the direction of mental health assessment. There are several machine intelligence methods like K-nearest neighbor (KNN), decision tree (DT), naïve Bayes (NB), random forest (RF), support vector machine (SVM), logistic regression (LR), etc. were used in a variety of ways in terms of mental health assessment (Thakur and Prediction 2022; Dubey et al. 2021; Dunn et al. 2018; Gedam and Paul 2021; Chahar and Dubey 2022; Ibrahim et al. 2021). These algorithms were used as they are efficient in terms of prediction, detection, extraction, and analysis.

The major problem with the previous research work is as follows:

1. The large-scale dataset is required for the proper analysis of the attribute impact, and it may be helpful in the improvement of the performance measures like accuracy (Thakur et al. 2022).
2. Analytical changes and sociocultural variables are important, and they should be incorporated into the results as they may affect the mental health assessment (Samuelson et al. 2022).
3. There is a need for cross-platform which should be the hybridization of intelligent approaches for the proper classification at each stage (Chahar et al. 2021).

To investigate the applicability of the combination of KNN and k-means algorithm for the mental health assessment is the objective of this paper. This paper presents the literature review in Sect. 2, and methodology part is explored in Sect. 3. Results and discussion have been investigated in Sect. 4, and finally, it is concluded in Sect. 5.

2 Related Works

In 2019, Deepa and Marseline (2019) discussed the applications of various machine learning methods to identify the autism spectrum disorder. They applied NB, SVM, and DT trees on the dataset considered from the UCI repository. They considered the total of 704 instances. SVM obtained the best result among the three methods. In 2019, Chancellor et al. (2019) conducted an analysis that unveiled three distinct areas of conflict: an ethical gap in social media research, concerns related to information, and considerations involving stakeholders. In 2019, Menzies et al. (2019) examined the associations between anxiety and death. They considered the population of 200 patients. Out of 200 participants, 126 were females. The various numbers of scales were applied like depression anxiety stress scale-21, 42, adult attachment styles, the meaning in life questionnaire, agoraphobia scale, etc. The results indicate the same predicate. In 2019, Masino et al. (2019) used a consumer-grade wearable device for the heart rate assessment. They used LR and SVM. The accuracy of LR and SVM models was 93% and 87%, respectively. In 2019, Pereira et al. (2019) examined various papers from scholarly journals and conferences. Their analysis revealed that the majority of these works utilized signal-based data, often collected through sensors. In 2019, Li et al. (2019) considered twenty-eight volunteers who participated in an experiment. It is based on facial task. SVM and a deep forest were used for the classification and the feature extraction. Using the activity, the deep learning method's greatest accuracy was 84.75%. In 2019, Burdisso et al. (2019) applied multi-NB, SVM, neural network, etc. as the various machine learning classifiers. Even though their classifier was less computationally expensive and had the capacity to justify its decisions, experimental data demonstrate that it was still able to outperform these models and traditional classifiers. In 2019, Bucci et al. (2019) examined the topic of digital health technology evaluation and intercession, focusing especially on supplement services provided for mental health treatment, as well as the obstacles and enablers to incorporating interventions delivered through digital media into service delivery. In 2019, Al Asad et al. (2019) suggested a data-analytic methodology to identify depression in any human. They gathered the information from posts made by individuals on the well-known social media platforms like Twitter and Facebook. Data scraped from SNS users was processed using machine learning. The accuracy, precision, and recall achieved by NB were 74%, 100%, and 60%. In 2019, Lovejoy (2019) discussed the successful use of artificial intelligence in healthcare, which might significantly raise the standard of care among people. In 2020, Ransing et al. (2020) developed a framework which might direct the creation, application, and assessment of mental health therapies. This conceptual framework was created by early career psychiatrists from 16 different countries, which represented every region under the World Health Organization. In 2020, Raj and Masood (2020) considered three different datasets of children, adolescents, and adults suffering from autism spectrum disorder. They have the instances of 292, 704, and 104 with 21 attributes of each. Among all the considered datasets and applied machine learning algorithms, convolutional neural network achieved the higher accuracy of 99.53, 98.30, and 96.88%. In 2020, Priya et al. (2020) used the depression, anxiety, and stress scale questionnaire-21 (DASS-21). They further applied the RF tree, NB, SVM, *K*NN, and DT on these datasets. The number of participants was 348 from the age group of 20–60 years. Among all the applied machine learning algorithms, RF achieved the best F1 score. In 2020, Kumar et al. (2020) considered the dataset named DASS-21 and DASS42. Multilayer perceptron obtained the highest accuracy of 98.85, 93.10, and 96.55% in anxiety, depression, and stress illnesses. In 2020, D'Alfonso (2020) discussed the artificial intelligence's role in the mental health. They discussed the modern streams of plentiful data which were allowed for the development of prediction and detection models for mental health disorders using data-driven artificial intelligence techniques. In 2020, Júnior et al. (2020) considered the different patients for mental health assessments. Bipolar disorder was identified in 61 patients, schizophrenia in 55, anxiety disorders in 41, and major depressive disorder in 43. In 2020, Wang et al. (2020) used the electronic medical records, and they classified 103 participants in a sample of 275 schizophrenia patients. They have considered different machine learning algorithms. They found SVM with different combination outperforms. In 2020, Aguirre et al. (2020) explored the five databases having mental health problems in adolescents whose age was between 10 years 19 years. They included ninety studies who qualified for the inclusion criteria. The bulk of therapies depended on psychoeducation, highlighting self-harm, depression, stigma, suicide, and all-around mental health awareness. In 2020, Bauer et al. (2020) discussed the use of apps of smartphones, which can be helpful in reviewing the mental health status. In 2021, Sonnweber et al. (2021) identified 370 offenders who suffered from schizophrenia spectrum disorder and they were admitted at Zurich University Hospital from 1982 to 2016. They applied LR, RF, gradient boosting, *K*NN, NB, and SVM as the machine learning methods. It was found that gradient boosting method achieved the best accuracy of 70.21%. In 2021, William and Suhartono (2021) performed the systematic review of the literature. They found that it is possible to detect depression early by considering social media impacts. In 2021, Liu et al. (2021) performed a retrospective study in Western Canada for 955 participants. Machine learning was used for performing the screening of applications. Their results confirmed the same. Their study found to have the potential bias of self-reporting in the depressed patients. In 2021, Shi et al. (2021) used fMRI and sMRI data to construct a new method known as multimodal imaging and multilevel characterization with multiclassifier (M3). They discovered that their method produced 83.49 as the classification accuracy and 68.69% and 93.75% as the sensitivity and specificity, respectively. When they employed the Brainnetome 246 atlas and without doing global signal regression (GSR), they achieved 0.8491 AUC. In 2021, Rajabimajd et al. (2021) conducted a comprehensive review of the information on the consequences of the pandemic on the workplace. The work and social adjustment scale scores were correlated with COVID-19 anxiety syndrome levels. In 2021, Chiong et al. (2021) used social site (Twitter) datasets for the experimentation. They used on other social websites also for the same. They developed a framework to train their machine learning classifiers. Among all the machine learning classifiers, RF was found to be best in the detection of depression. In 2021, Richter et al. (2021) outlined the basic key machine learning-based techniques used to diagnose depression and other mental diseases using behavioral data. In 2021, Lalik et al. (2021) introduced a novel acoustic system, which uses acoustic wave frequency measurements to assess the state of concrete structures. A real-time classifier was suggested as the approach to enable the self-acoustic system (SAS) system. In 2021, Ceccarelli et al. (2021) considered well-being and bipolar disorder dataset. They suggested an approach which combines text, audio, and video modes. The adaptive classifier achieved the accuracy of 91.67% and F1-score of 88.31%. In 2021, Lin et al. (2021) collected 17 different datasets from Physionet base. They suggested a fuzzy-assisted approach for stress evaluation on heart rate and blood pressure monitoring. The results show the numerical validation of 93.55%, 89.01%, 89.50%, and 89.901% as the performance rate, precision, recall, and adaptation in stress management, respectively. In 2021, Nemesure et al. (2021) considered the sample of 4184 undergraduate samples. They found this approach to be prominent considering different performance measures. In 2022, Revathi et al. (2022) used a two-stage categorization technique to increase the accuracy of their cognitive impairment prediction. They have used SVM and RF algorithms. In 2022, Loh et al. (2022) employed a dataset comprising individuals with major depressive disorder (MDD) and control participants. This dataset was constructed by initially applying the short-time Fourier transform (STFT) to the EEG data. Subsequently, a CNN algorithm utilized these spectrogram images to automatically distinguish between MDD patients and healthy individuals. In 2022, Varshney et al. (2022) discussed DT, RF, NB, ensemble models, *K*NN, maximum entropy, etc., a portion of sentimental analysis. In 2022, Amanat et al. (2022) endeavored to predict depression from text. They employed the long short-term memory (LSTM) model and achieved an accuracy of 99%.

The overall analysis discussed in this section indicated that the combination of intelligent algorithms is capable in the mental health assessment, but there is the need to analyze the work in the direction of attribute analysis, hybridization of approaches, variations in parameter, and controlled hyperparameter tuning.

3 Methods

In this paper, we have applied the *K*NN algorithm for the mental health assessment. For the experimentation, open-sourcing mental illness (OSMI) in tech survey 2019 was considered from Kaggle repository. The attributes are age, gender, country, state, and the list of questionaries. There are a total of 352 respondents with 80 column categories. Data preprocessing has been performed first considering auto-encoding to recreate the input. This data is then inputted for the *K*NN classification. We have computed the K value in random order and compared the error rates. It is found that the small K value leads to an unstable decision area so sustainable K value is required which should be higher. Then the final K value has been selected which shows the less error rates. We have applied the k-means algorithm before the assignment of top k rows. K-means algorithm helps in the assignment of more appropriate class labels. It is shown below. Based on the distance measures, k-means is capable in the assignment of the data point in the best cluster group. It is considered as the threshold for the class label assignment. The k-means-based KNN (k-KNN) is shown in Fig. 1. The experimentation has been performed on the Intel (R) Core (TM), i5-10210U CPU @ 1.60 GHz, 2.11 GHz processor and 8GB RAM. Python libraries have been used for the implementation.

Assumptions: {d: Data point distance; *Xi, Yi*: data point coordinate; *n:* Total number; *N:* Total pairs of data point}

Step 1: Initialize the data for the model

Step 2: Calculate the distance considering each data point

Three distance measures have been considered

Euclidean distance: $d = \sqrt{\sum_{i=1}^{n}(X_i - Y_i)}$

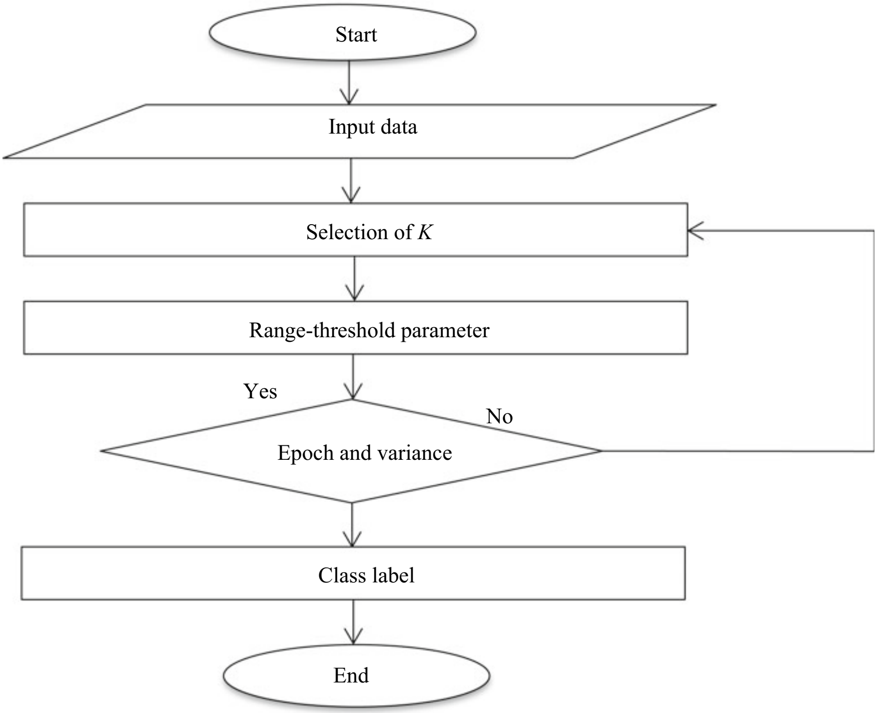


Fig. 1 Flowchart of k-KNN approach

Manhattan distance: $d = \sum_{i=1}^{n} |X_i - Y_i|$

Pearson Correlation: $r = \frac{\sum XY - \frac{\sum X \sum Y}{N}}{\sqrt{\sum X^2 - \frac{(\sum X)^2}{N}}\sqrt{\sum Y^2 - \frac{(\sum Y)^2}{N}}}$

Step 3: Based on the distance measures arrange it

Step 4: Threshold assignment for the nearest neighbors

The data point cluster center is calculated with variant epochs to find the variance: $c_i = \left(\frac{1}{n_i}\right) \sum_{j=1}^{n_i} d_i$

Minimum variance has been considered for the final threshold assignment

The assignment of the data point has been performed

Step 5: Top k rows have been selected for the assignment of the final class label

Step 6: End

4 Results and Discussion

The following performance measures have been considered for the experimentation and result analysis.

- Accuracy $= \frac{\text{TP+TN}}{\text{TP+FP+FN+TN}}$ [It shows the correct prediction numbers in the complete predictions]
- Sensitivity $= \frac{\text{TP}}{\text{TP+FN}}$ [It shows the total positive correct prediction numbers in the complete positive cases]
- Specificity $= \frac{\text{TN}}{\text{TN+FP}}$ [It shows the total negative correct prediction numbers in the complete negative cases]
- F1-score $= 2 \times \frac{\text{precison} \times \text{recall}}{\text{precison} + \text{recall}}$.

Where TP indicates true positive, TN indicates a true negative, FP indicates false positive, and FN indicates false negative.

The cases and parameters considered for the experimentation are shown in Table 1.

Table 1 Cases for the result evaluation

S. No	Cases	Ranges
1	Variable epochs (E) and K	$E = 100$ to 550; $K = 3$ to 7
2	Performance measures	Aggregate values
3	Training ratio	Split ratio (70:30; 75:25; 80:20; 85:15)

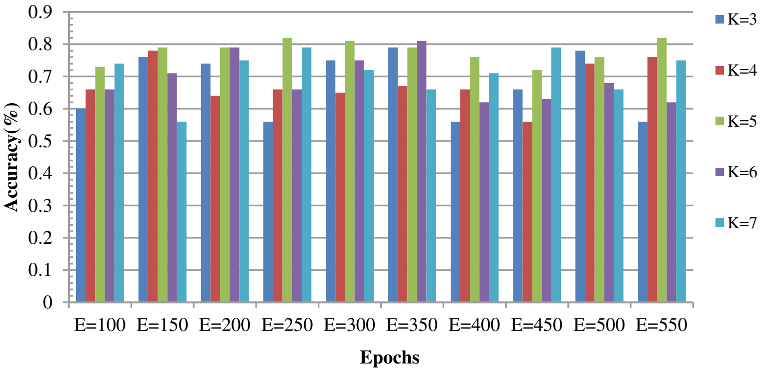


Fig. 2 Results of accuracy based on epochs and K

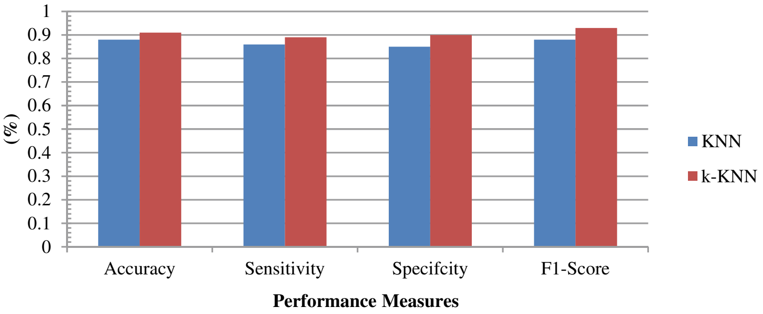


Fig. 3 Results of performance measures for KNN and k-KNN

Figure 2 shows the results of accuracy based on epochs and K. It clearly shows the error is high if the K value is low. Better results have been observed in the case of $K = 5$ and $K = 6$. The values are stable beyond $K = 7$. It has been observed in case of different epochs. Figure 3 shows the results of performance measures for KNN and k-KNN. It clearly shows the improvement due to the best selection of k along with the threshold range selection mechanism. Figure 4 shows the results of accuracy based on training ratios. It clearly depicted that the results variations are minor. It shows the average accuracy of variable epochs. The results depict the effectiveness of the combined approach. It shows the improvement in terms of K, variable epochs, and considering varying split ratios. The overall average accuracy obtained by k-KNN is approximately 92%, and it is 88% by KNN alone. Considering different epochs, the average accuracy is approximately 90% and 87% in the case of k-KNN and KNN, respectively.

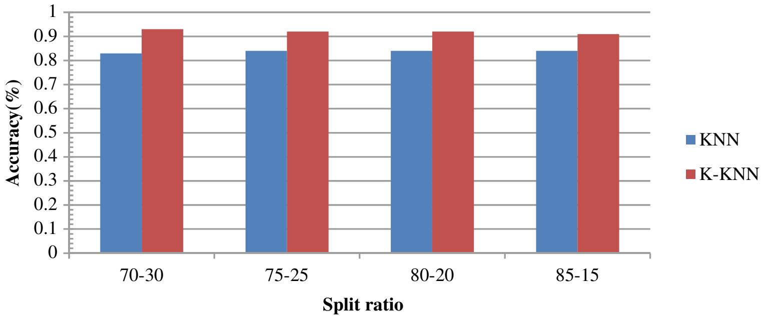


Fig. 4 Results of accuracy based on training ratios

5 Conclusion

In this paper, a combination of *K*NN and *k*-means has been applied to the mental health dataset. It has been applied considering the threshold range of *k*-means. The selection is completely dependent on the threshold so that to avoid the issues of overfitting. Three different distance measures, namely Euclidean, Manhattan, and Pearson Correlation coefficients, were considered. Euclidean distance was found to be efficient in all these measures. The variations found in different epochs were also considered for the average accuracy to show the randomness if any. The results were found to be efficient considering different cases and parameters. The overall average accuracy obtained by *k*-*K*NN is approximately 92%.

References

Aguirre Velasco A, Cruz IS, Billings J, Jimenez M, Rowe S (2020) What are the barriers, facilitators and interventions targeting help-seeking behaviours for common mental health problems in adolescents? A systematic review. BMC Psychiatry 20(1):1–22

Al Asad N, Pranto MA, Afreen S, Islam MM (2019) Depression detection by analyzing social media posts of user. In: 2019 IEEE international conference on signal processing, information, communication and systems. IEEE, pp 13–17

Amanat A, Rizwan M, Javed AR, Abdelhaq M, Alsaqour R, Pandya S, Uddin M (2022) Deep learning for depression detection from textual data. Electronics 11(5):676

Bauer M, Glenn T, Geddes J, Gitlin M, Grof P, Kessing LV, Monteith S, Faurholt-Jepsen M, Severus E, Whybrow PC (2020) Smartphones in mental health: a critical review of background issues, current status and future concerns. Int J Bipolar Disord 8(1):1–9

Bucci S, Schwannauer M, Berry N (2019) The digital revolution and its impact on mental health care. Psychol Psychother Theor Res Pract 92(2):277–297

Burdisso SG, Errecalde M, Montes-y-Gómez M (2019) A text classification framework for simple and effective early depression detection over social media streams. Exp Syst Appl 133:182–197

Ceccarelli F, Mahmoud M (2021) Multimodal temporal machine learning for bipolar disorder and depression recognition. Pattern Anal Appl 1–2

Chahar R, Dubey AK (2022) A review and analysis of IoT and machine learning algorithms in the brain disease diagnosis and detection. ECS Trans 107(1):6641

Chahar R, Dubey AK, Narang SK (2021) A review and meta-analysis of machine intelligence approaches for mental health issues and depression detection. Int J Adv Technol Eng Explor 8(83):1279

Chancellor S, Birnbaum ML, Caine ED, Silenzio VM, De Choudhury M (2019) A taxonomy of ethical tensions in inferring mental health states from social media. In: Proceedings of the conference on fairness, accountability, and transparency, pp 79–88

Chiong R, Budhi GS, Dhakal S, Chiong F (2021) A textual-based featuring approach for depression detection using machine learning classifiers and social media texts. Comput Biol Med 135:104499

D'Alfonso S (2020) AI in mental health. Curr Opin Psychol 36:112–117

Deepa B, Marseline KJ (2019) Exploration of autism spectrum disorder using classification algorithms. Procedia Comput Sci 165:143–150

Dubey A, Gupta U, Jain S (2021) Medical data clustering and classification using TLBO and machine learning algorithms. Comput Mater Continua 70(3):4523–4543

Dunn J, Runge R, Snyder M (2018) Wearables and the medical revolution. Pers Med 15(5):429–448

Garcia-Ceja E, Riegler M, Nordgreen T, Jakobsen P, Oedegaard KJ, Tørresen J (2018) Mental health monitoring with multimodal sensing and machine learning: a survey. Pervasive Mob Comput 51:1–26

Gedam S, Paul S (2021) A review on mental stress detection using wearable sensors and machine learning techniques. IEEE Access 9:84045–84066

Hollis C, Sampson S, Simons L, Davies EB, Churchill R, Betton V, Butler D, Chapman K, Easton K, Gronlund TA, Kabir T (2018) Identifying research priorities for digital technology in mental health care: results of the James Lind Alliance Priority Setting Partnership. Lancet Psychiatry 5(10):845–854

Ibrahim EN, Jamali N, Suhaimi AI (2021) Exploring gamification design elements for mental health support. Int J Adv Technol Eng Explor 8(74):114

Júnior ÉD, Passos IC, Scott J, Bristot G, Scotton E, Mendes LS, Knackfuss AC, Gerchmann L, Fijtman A, Trasel AR, Salum GA (2020) Decoding rumination: a machine learning approach to a transdiagnostic sample of outpatients with anxiety, mood and psychotic disorders. J Psychiatr Res 121:207–213

Kilbourne AM, Beck K, Spaeth-Rublee B, Ramanuj P, O'Brien RW, Tomoyasu N, Pincus HA (2018) Measuring and improving the quality of mental health care: a global perspective. World Psychiatry 17(1):30–38

Kumar P, Garg S, Garg A (2020) Assessment of anxiety, depression and stress using machine learning models. Procedia Comput Sci 171:1989–1998

Lalik K, Kozek M, Dominik I (2021) Autonomous machine learning algorithm for stress monitoring in concrete using elastoacoustical effect. Materials 14(15):4116

Le Glaz A, Haralambous Y, Kim-Dufor DH, Lenca P, Billot R, Ryan TC, Marsh J, Devylder J, Walter M, Berrouiguet S, Lemey C (2021) Machine learning and natural language processing in mental health: systematic review. J Med Internet Res 23(5):e15708

Levis B, Benedetti A, Thombs BD (2019) Accuracy of Patient Health Questionnaire-9 (PHQ-9) for screening to detect major depression: individual participant data meta-analysis. BMJ 365

Li X, Zhang X, Zhu J, Mao W, Sun S, Wang Z, Xia C, Hu B (2019) Depression recognition using machine learning methods with different feature generation strategies. Artif Intell Med 99:101696

Lin Q, Li T, Shakeel PM, Samuel R (2021) Advanced artificial intelligence in heart rate and blood pressure monitoring for stress management. J Ambient Intell Humaniz Comput 12(3):3329–3340

Liu Y, Hankey J, Cao B, Chokka P (2021) Screening for major depressive disorder in a tertiary mental health centre using EarlyDetect: a machine learning-based pilot study. J Affect Disord R 3:100062

Loh HW, Ooi CP, Aydemir E, Tuncer T, Dogan S, Acharya UR (2022) Decision support system for major depression detection using spectrogram and convolution neural network with EEG signals. Exp Syst 39(3):e12773

Lovejoy CA (2019) Technology and mental health: the role of artificial intelligence. Eur Psychiatry 55:1–3

Low DM, Bentley KH, Ghosh SS (2020) Automated assessment of psychiatric disorders using speech: a systematic review. Laryngoscope Invest Otolaryngol 5(1):96–116

Magson NR, Freeman JY, Rapee RM, Richardson CE, Oar EL, Fardouly J (2021) Risk and protective factors for prospective changes in adolescent mental health during the COVID-19 pandemic. J Youth Adolesc 50(1):44–57

Mannarini S, Rossi A (2019) Assessing mental illness stigma: a complex issue. Front Psychol 9:2722

Masino AJ, Forsyth D, Nuske H, Herrington J, Pennington J, Kushleyeva Y, Bonafide CP (2019) M-Health and autism: recognizing stress and anxiety with machine learning and wearables data. In: 2019 IEEE 32nd international symposium on computer-based medical systems. IEEE, pp 714–719

Menzies RE, Sharpe L, Dar-Nimrod I (2019) The relationship between death anxiety and severity of mental illnesses. Br J Clin Psychol 58(4):452–467

Nemesure MD, Heinz MV, Huang R, Jacobson NC (2021) Predictive modeling of depression and anxiety using electronic health records and a novel machine learning approach with artificial intelligence. Sci Rep 11(1):1–9

Pereira CR, Pereira DR, Weber SA, Hook C, De Albuquerque VH, Papa JP (2019) A survey on computer-assisted Parkinson's disease diagnosis. Artif Intell Med 95:48–63

Priya A, Garg S, Tigga NP (2020) Predicting anxiety, depression and stress in modern life using machine learning algorithms. Procedia Comput Sci 67:1258–1267

Raj S, Masood S (2020) Analysis and detection of autism spectrum disorder using machine learning techniques. Procedia Comput Sci 167:994–1004

Rajabimajd N, Alimoradi Z, Griffiths MD (2021) Impact of COVID-19-related fear and anxiety on job attributes: a systematic review. Asian J Soc Health Behav 4(2):51

Ransing R, Adiukwu F, Pereira-Sanchez V, Ramalho R, Orsolini L, Teixeira AL, Gonzalez-Diaz JM, da Costa MP, Soler-Vidal J, Bytyçi DG, El Hayek S (2020) Mental health interventions during the COVID-19 pandemic: a conceptual framework by early career psychiatrists. Asian J Psychiatr 51:102085

Rashid B, Calhoun V (2020) Towards a brain-based predictome of mental illness. Hum Brain Mapp 41(12):3468–3535

Rashmi CR, Shantala CP (2022) EEG artifacts detection and removal techniques for brain computer interface applications: a systematic review. Int J Adv Technol Eng Explor 9(88):354

Revathi A, Kaladevi R, Ramana K, Jhaveri RH, Rudra Kumar M, Sankara Prasanna Kumar M (2022) Early detection of cognitive decline using machine learning algorithm and cognitive ability test. Sec Commun Netw

Richter T, Fishbain B, Richter-Levin G, Okon-Singer H (2021) Machine learning-based behavioral diagnostic tools for depression: advances, challenges, and future directions. J Personalized Med 11(10):957

Salari N, Khazaie H, Hosseinian-Far A, Khaledi-Paveh B, Kazeminia M, Mohammadi M, Shohaimi S, Daneshkhah A, Eskandari S (2020) The prevalence of stress, anxiety and depression within front-line healthcare workers caring for COVID-19 patients: a systematic review and meta-regression. Hum Resour Health 18(1):1–4

Samuelson KW, Dixon K, Jordan JT, Powers T, Sonderman S, Brickman S (2022) Mental health and resilience during the coronavirus pandemic: a machine learning approach. J Clin Psychol 78(5):821–846

Shi D, Li Y, Zhang H, Yao X, Wang S, Wang G, Ren K (2021) Machine learning of schizophrenia detection with structural and functional neuroimaging. Disease markers

Sonnweber M, Lau S, Kirchebner J (2021) Violent and non-violent offending in patients with schizophrenia: exploring influences and differences via machine learning. Compr Psychiatry 107:152238

Srividya M, Mohanavalli S, Bhalaji N (2018) Behavioral modeling for mental health using machine learning algorithms. J Med Syst 42(5):1–2

Thakur B, Kumar N, Gupta G (2022) Machine learning techniques with ANOVA for the prediction of breast cancer. Int J Adv Technol Eng Explor 9(87):232

Thakur B, Kumar N (2022) Prediction, detection and recurrence of breast cancer using machine learning based on image and gene datasets. In: Recent innovations in computing: proceedings of ICRIC 2021, vol 1, 10 Mar 2022, pp 263–73

Thorstad R, Wolff P (2019) Predicting future mental illness from social media: a big-data approach. Behav Res Methods 51(4):1586–1600

Varshney T, Gupta S, Goel L (2022) Literature survey on depression detection using machine learning. In: Proceedings of the international conference on cognitive and intelligent computing. Springer, Singapore, pp 313–321

Wang KZ, Bani-Fatemi A, Adanty C, Harripaul R, Griffiths J, Kolla N, Gerretsen P, Graff A, De Luca V (2020) Prediction of physical violence in schizophrenia with machine learning algorithms. Psychiatry Res 289:112960

William D, Suhartono D (2021) Text-based depression detection on social media posts: a systematic literature review. Procedia Comput Sci 179:582–589

Classification of Skin Cancer Using Deep Learning

A. Nagarjun, N. Manju, V. N. Manjunath Aradhya, N. Shruthi, and S. Malapriya

1 Introduction

A tumor is formed when healthy cells begin to change and grow out of control. Both cancerous and noncancerous tumors are conceivable. Malignant tumors are those that have the potential to grow and spread to other areas of the body (Lee et al. 2018). This number exceeds the sum of cases of lung, bone, and colon cancers. In actuality, a melanoma victim is lost every 57 s. Skin cancer early detection is still a prized possession. It's difficult to tell if a skin lesion is benign or malignant because they all seem similar (Jamil et al. 2016). The sun's harmful ultraviolet (UV) rays and the usage of UV tanning beds are the two most common causes of skin cancer. It is particularly difficult for dermatologists to distinguish between melanoma and non-melanoma lesions because of the low degree of difference between lesions and skin. The main problem of similar opinion is largely dependent on private judgment and is scarcely reproducible (Pai and Giridharan 2019). With the help of robotization using operation of deep literacy helps the case to get the early opinion report and grounded on the report case can consult dermatologists for treatment.

A. Nagarjun · N. Manju (✉) · N. Shruthi · S. Malapriya
Department of Information Science and Engineering, JSS Science and Technology University, Mysuru, India
e-mail: manjun007@jssstuniv.in

V. N. M. Aradhya
Department Computer Applications, JSS Science and Technology University, Mysuru, India

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_43

2 Literature Review

The methodology for classification of skin lesions with fine-tuned neural networks is proposed in Lee et al. (2018). For the purpose of balancing the dataset, the skin lesion photos are resampled. A combination of DenseNet and U net is trained for segmentation and then used to fine-tune the following classifiers (Jamil et al. 2016). The encoder element of the segmentation model's extracted architecture is then trained to categorize the seven different skin disorders (Pai and Giridharan 2019; Badrinarayanan et al. 2017). The classification model's average balanced accuracy was 0.836 in the test set and 0.840 in the validation set. An innovative strategy is put forth in that can pre-process the image automatically before segmenting the lesion (Shete et al. 2021). Hair, gel, bubbles, and specular reflection are just a few of the undesirable artefacts that the system filters (Vidya and Karki 2020). An innovative method for identifying and painting the hairs visible in cancer images is given using the wavelet idea (Mhaske and Phalke 2013). Utilizing an adaptive sigmoidal function that manages the localised intensity distribution within the pictures of a specific lesion, the contrast between the lesion and the skin is improved (Nahata and Singh 2020; Wighton et al. 2011). A segmentation method to precisely separate the lesion from the surrounding tissue. On the European dermoscopic image database, the proposed method is tested. The proposed system is focusing on classifying skin lesions in deep learning with specific implementation of CNN approach (Saeed and Zeebaree 2021). The methods in this paper includes the screening of a dataset from MNIST: HAM10000 which consists of seven different types of skin lesions with the sample size of 10,015. The methods in this paper included training the model with the help of CNN and obtained an accuracy of 78%. In the study, a deep fully convolutional neural network is presented for semantic pixel-by-pixel segmentation. A pixel-wise classification layer, a corresponding decoder network, and an encoder network make up the trainable network (Li et al. 2016). Convolution layers are identical to those in VGG-16's 13 convolution layers. As its name implies, the decoder network's primary goal is to convert encoder feature maps into full input resolution feature maps. The method is focusing on classifying skin lesions in deep learning with specific implementation of CNN approach (Jamil et al. 2016). The methods in this paper includes the screening of a dataset from MNIST: HAM10000 which consists of seven different types of skin lesions with the sample size of 10,015. The methods in this paper included training the model with the help of CNN and obtained an accuracy of 88% and used transfer learning methods like the Resnet model (Jamil et al. 2016).

3 Methodology

Detecting skin cancer at an early stage is challenging for dermatologists, with extensive use of deep learning as shown in Fig. 1 helps to classify the seven types of skin cancer images. Deep learning methods like Convolutional Neural Networks (CNN), has surpassed all others in object detection and classification tests.

3.1 Dataset Description

The dataset contains images of skin lesions from seven different classes. HAM10000 and ISIC from the National Institutes of Standards and Technology (NIST) include 10,015 dermatoscopic images of all significant diagnostic categories of pigmented skin lesions.

3.2 Data Pre-processing

The visual presentation of information and data is known as data visualization. Data visualization tools make it easier to spot trends and outliers in data by using visual elements like graphs, charts, and maps. Figure 2 shows the dataset of the skin lesions to the respective classes they belong to.

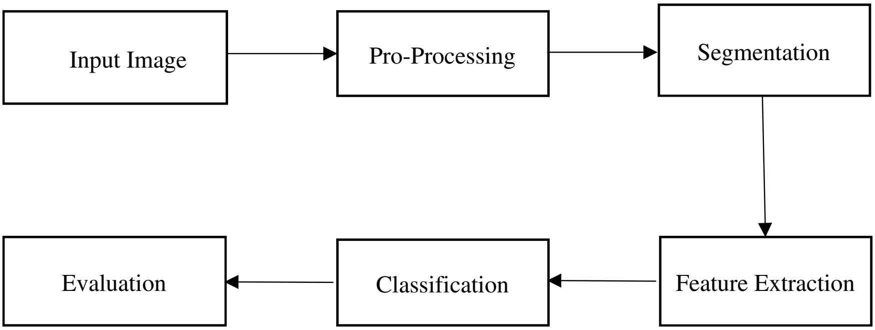


Fig. 1 Methodology

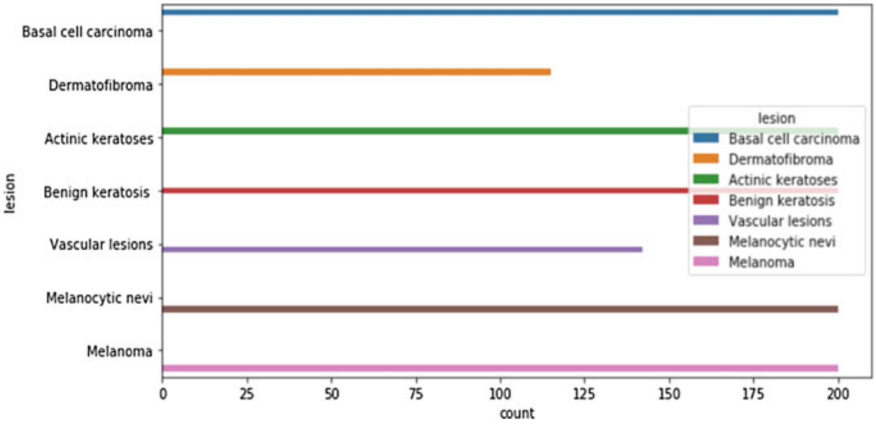


Fig. 2 Data visualization

3.3 *Segmentation of Image Using Encoder and Decoder*

The technique of segmenting an image involves dividing the foreground from the background or grouping pixels according to how similar they are in terms of color or shape.

Encoder: Convolutions and max pooling are done at the encoder. From the original fully connected layers, 13 convolutional layers were eliminated. The associated max pooling indices (locations) are stored while executing 2 × 2 max pooling.

$$\begin{aligned} &\phi : X \rightarrow F \\ &\psi : F \rightarrow X \\ &\phi, \psi = \arg\min \| X - (\psi \circ \phi) X \|^2 \end{aligned} \tag{1}$$

In the encoder function, indicated by ϕ, a latent space F is used to map the original data X. To get the data out of the bottleneck, we use the decoder function, which we indicate by ψ. Here, the result is the same as what you got when you put it in. After some generic nonlinear compression, we attempt to restore the original image which is given in Eq. (1). Standard neural network function is transferred via an activation function, which represents the latent dimension z in the encoding network is given in Eq. (2)

$$Z = \sigma(Wx + b) \tag{2}$$

It is also possible to describe the decoding network this way, but with a wide range of weights, biased and activation functions is given in Eq. (3).

$$X^{|} = \sigma^{|}\left(W^{|}z + b^{|}\right) \tag{3}$$

The loss function may then be defined in terms of these network functions, and the neural network will be trained using the traditional backpropagation method with this loss function is given in Eq. (4).

$$L\left(X, X^{|}\right) = \left\|X - X^{|}\right\|^2 = \left\|X - \sigma^{|}\left(W^{|}(\sigma(Wx + b)) + b^{|}\right)\right\|^2 \tag{4}$$

Decoder: Up sampling and convolutions are done at the decoder. Each pixel has a softmax classifier at the end. Prior to sampling, encoder layers remember their highest possible maximum pooling indices. In the end, a K-class softmax classifier is used to predict each pixel's class.

4 System Implementation

Convolutional Neural Networks (CNNs) are built in such a way that they can account for the input's spatial structure. An input layer, an output layer, and several hidden layers make up a convolutional neural network. Convolutional, RELU (activation function), pooling, fully connected, and normalization layers are often seen in a CNN's hidden layers. The features are extracted automatically. Figure 3 shows the CNN architecture of skin lesion images.

Pre-trained models like Densenet169 is used to train the model. The model was trained for 30 epochs. At each iteration, the training accuracy improves but the validation loss is detected more often, resulting in a reduction in the training accuracy at each iteration of the plot and Fig. 4 shows the learning graph of training and validation accuracy plotted for 30 epochs.

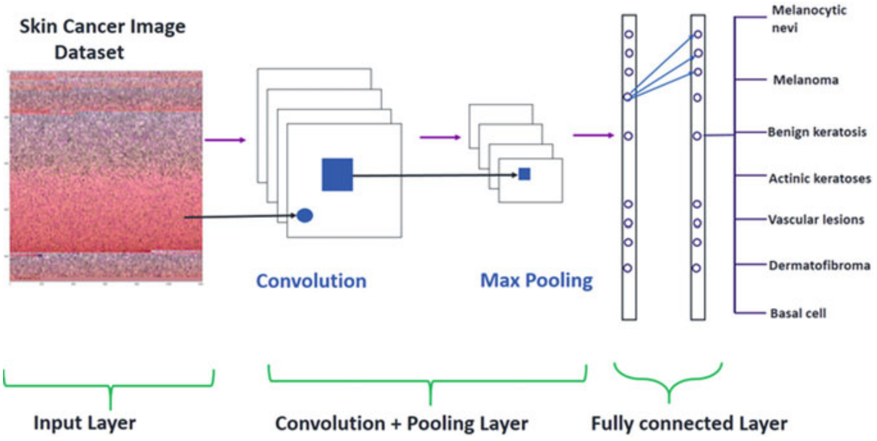


Fig. 3 CNN architecture of skin lesion images

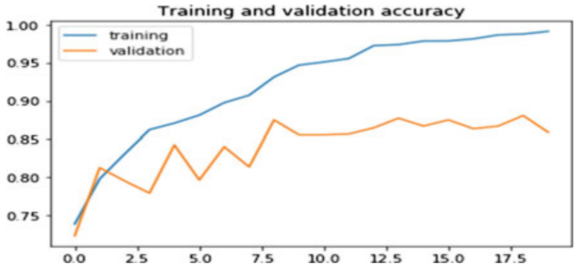


Fig. 4 Learning graph of training and validation accuracy for 30 epochs

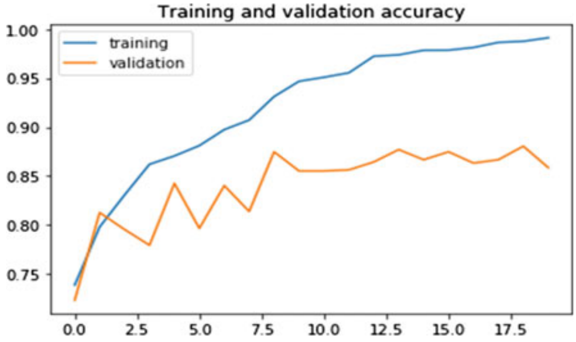


Fig. 5 Learning graph of training and validation accuracy for 20 epochs

Table 1 Comparison of training the model with different split

Architecture	Split ratio	Accuracy in %
Densenet169	80:20	91.20
	70:30	87.70

The accuracy of the model obtained is 91.20%. The training and testing ratio were set to different ranges like 80:20, 70:30. For 70:30 the model is trained for 20 epochs with a learning rate of 0.000001. The accuracy of the model obtained is 87.70%. Figure 5 shows the training and validation accuracy graph plot.

The model is trained with densenet169 architecture. The accuracy obtained is 91.20% and f1 measure of 91.70%. The training and testing ratio were conducted on different ranges like 80:20, 70:30 as shown in Table 1.

5 Conclusion and Future Scope

Skin cancer is one of the illnesses that is spreading the quickest on the earth. Skin cancer is mostly brought on by a person's vulnerability to the sun's UV radiation. Given the limited resources available, early identification of skin cancer is essential. Accurate diagnosis and identification viability are generally essential for skin cancer prevention strategies. Additionally, dermatologists have trouble seeing skin cancer in its early stages. Convolutional Neural Networks (CNNs) are one of these models that have excelled in object identification and classification tasks (CNN). The dataset is filtered from MNIST: HAM10000 and ISIC, which has a sample size of 10,015 and includes seven different types of skin lesions. Data pre-processing method like segmentation using an autoencoder and decoder is used. The model was trained using transfer learning methods such as DenseNet169. Different ratios of 80:20, 70:30 were used for the training and assessment. DenseNet169's architecture produced an accuracy of 91.20% and 87.70 respectively.

References

Badrinarayanan V, Kendall A, Cipolla R (2017) Segnet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell 39(12):2481–2495

Jamil U, Akram MU, Khalid S, Abbas S, Saleem K (2016) Computer based melanocytic and nevus image enhancement and segmentation. BioMed Res Int

Lee YC, Jung SH, Won HH (2018) WonDerM: skin lesion classification with fine-tuned neural networks. arXiv preprint arXiv:1808.03426

Li Y, Esteva A, Kuprel B, Novoa R, Ko J, Thrun S (2016) Skin cancer detection and tracking using data synthesis and deep learning. arXiv preprint arXiv:1612.01074

Mhaske HR, Phalke DA (2013, December) Melanoma skin cancer detection and classification based on supervised and unsupervised learning. In: 2013 international conference on circuits, controls and communications (CCUBE). IEEE, pp 1–5

Nahata H, Singh SP (2020) Deep learning solutions for skin cancer detection and diagnosis. In: Machine learning with health care perspective. Springer, Cham, pp 159–182

Pai K, Giridharan A (2019, October) Convolutional Neural Networks for classifying skin lesions. In: TENCON 2019–2019 IEEE region 10 conference (TENCON). IEEE, pp 1794–1796

Saeed J, Zeebaree S (2021) Skin lesion classification based on deep convolutional neural networks architectures. J Appl Sci Technol Trends 2(01):41–51

Shete AS, Rane AS, Gaikwad PS, Patil MH (2021) Detection of skin cancer using CNN algorithm. Int J 6(5)

Vidya M, Karki MV (2020, July) Skin cancer detection using machine learning techniques. In: 2020 IEEE international conference on electronics, computing and communication technologies (CONECCT). IEEE, pp 1–5

Wighton P, Lee TK, Lui H, McLean DI, Atkins MS (2011) Generalizing common tasks in automated skin lesion diagnosis. IEEE Trans Inf Technol Biomed 15(4):622–629

Author Index

N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1