Lecture Notes in Electrical Engineering 959

Anand J. Kulkarni
Seyedali Mirjalili
Siba Kumar Udgata *Editors*

Intelligent Systems and Applications

Select Proceedings of ICISA 2022

Springer

Lecture Notes in Electrical Engineering

Volume 959

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. ****

Anand J. Kulkarni · Seyedali Mirjalili ·
Siba Kumar Udgata
Editors

Intelligent Systems and Applications

Select Proceedings of ICISA 2022

Springer

Editors
Anand J. Kulkarni
Institute of Artificial Intelligence
Dr Vishwanath Karad MIT World Peace University
Pune, India

Seyedali Mirjalili
Torrens University
Adelaide, SA, Australia

Siba Kumar Udgata
School of Computer and Information Science
University of Hyderabad
Hyderabad, India

ISSN 1876-1100 ISSN 1876-1119 (electronic)
Lecture Notes in Electrical Engineering
ISBN 978-981-19-6580-7 ISBN 978-981-19-6581-4 (eBook)
https://doi.org/10.1007/978-981-19-6581-4

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

Artificial intelligence (AI) has been instrumental in almost every fundamental as well as multidisciplinary application-based research. Several intelligent systems have been developed in the numerous domains including robotics, pharmaceutics, baking and finance, agriculture, food processing, crime prevention, smart homes, transportation and traffic control, wild life conservation, disaster management, health care, manufacturing, sports, education, human resource, IoT, pollution control, politics, governance, fashion design, cybersecurity, etc., which are just to mention a few. Moreover, every country in the world is taking lead in developing such systems which is positively affecting the society at large. In order to be a driver and contributor in this direction, the Institute of Artificial Intelligence of the MIT World Peace University, Pune, Maharashtra, India, successfully organized the 1st International Conference on Intelligent Systems and Applications (ICISA) on May 4 and 5, 2022. Several academicians, researchers, scientists, professionals, and students shared their findings, knowledge, and expertise in the fields associated with intelligent systems in agriculture, Internet of Things and cloud computing, image processing and deep learning, cybersecurity and data security, intelligent systems for social welfare, social media, natural language processing, medical and healthcare management, optimization algorithms and applications, finance, and stock market, etc. The associated sessions were chaired by several eminent researchers and academicians including Dr. E. S. Gopi from NIT Trichy, India; Dr. Ishaan R. Kale, Dr. Ratnakar R. Ghorpade, Prof. Apoorva S. Shastri, Dr. Priyanka Gujrati, Dr. Rohit Ghadge, Dr. Anand J. Kulkarni, Dr. Saket R. Yeolekar, Dr. Sushil Kumar Gupta, Dr. Pankaj Dhatrak, Dr. Shubhalaxmi Joshi from MIT World Peace University, Pune, India; Dr. Mangal Singh and Prof. Aniket Nargundkar from Symbiosis International (Deemed University), Pune; Dr. Shahin Jalili from University of Aberdeen, UK; Prof. Pavnesh Kumar from Mahatma Gandhi Central University, India; Dr. Krishna Prasad Ponnekanti from ANITS, Visakhapatnam, India; Dr. Anabel Martinez Vargas from Universidad Politécnica de Pachuca, Mexico; Dr. Parikshit Joshi from IIM Ahmedabad; Dr. Priya Ranjan from Bhubaneswar Institute of Technology; and Dr. Sachin Salunkhe from Vel Tech, Chennai, India.

In addition, three elite keynote talks were delivered by Dr. Seyedali Mirjalili, Torrens University, Australia, Dr. Priya Ranjan from Bhubaneswar Institute of Technology, and Dr. Naoufel Cheikhrouhou, University of Applied Sciences, Western Switzerland. We would like to extend our special thanks to Pro-Vice-Chancellor of the MIT World Peace University Dr. Milind Pande for his support and motivation in seamless organization of the conference. Thanks are due to very distinguished reviewers for their timely support and competent team members for successfully organizing the event. Importantly, we thank Ms. Kamiya Khatter, Associate Editor, Springer, and Ms. Rini Christy, the Project Coordinator, Springer, for all the help they provided.

Pune, India — Anand J. Kulkarni
Brisbane, Australia — Seyedali Mirjalili
Hyderabad, India — Siba Kumar Udgata

Contents

About the Editors

Anand J. Kulkarni holds a Ph.D. in Distributed Optimization from Nanyang Technological University, Singapore, an MS in Artificial Intelligence from the University of Regina, Canada, and a Bachelor of Engineering from Shivaji University, India, and a Diploma from the MSBTE, Mumbai. He worked as a Post Doctorate Research Fellow at Odette School of Business, University of Windsor, Canada. Dr. Kulkarni has worked with Symbiosis International University, Pune, India for over six years. Currently, he is a Professor and Associate Director at the Institute of AI at MITWPU. His research interests include optimization algorithms, multi-agent systems, complex systems, swarm optimization, and self-organizing systems. Anand pioneered socio-inspired optimization methodologies such as Cohort Intelligence, Ideology Algorithm, Expectation Algorithm, and Socio Evolution and Learning Optimization Algorithm. He is the founder and chairman of Optimization and Agent Technology Research Lab and has over 70 research papers in journals and conferences, 04 authored and 08 edited books to his credit. Dr. Kulkarni is the lead editor for the Springer and Taylor and Francis book series. He regularly writes on Artificial Intelligence in several newspapers and magazines. Dr. Kulkarni has delivered expert research talks in many countries such as the USA, Canada, Singapore, Malaysia, India, and France.

Seyedali Mirjalili is a Professor at Torrens University Center for Artificial Intelligence Research and Optimization and is internationally recognized for his advances in nature-inspired Artificial Intelligence (AI) techniques. He is the author of more than 350 publications. With over 50,000 citations and an H-index of 80, he is one of the most influential AI researchers in the world. From Google Scholar metrics, he is globally the most cited researcher in Optimization using AI techniques, which is his main area of expertise. Since 2019, he has been on the list of 1% highly-cited researchers and named as one of the most influential researchers in the world by Web of Science. He is a senior member of IEEE and is serving as an editor of leading AI journals.

Siba Kumar Udgata is currently a Professor of Computer and Information Sciences at the University of Hyderabad, India, where he directs a research group focusing on sensor networks, IoT, wireless communications, and intelligent algorithms. He worked as a Research Fellow at the United Nations University/International Institute of Software Technology (UNU/IIST), Macau. He has published over 100 research papers in peer-reviewed journals and at international conferences. He has edited ten international conference proceedings for Springer LNAI, AISC, and SIST. He is a recipient of the IBM SUR (Shared University Research) award for the project "Mobile Sensor network based rescue management system". He has completed seven Government of India-sponsored research projects in sensor networks, IoT, and cognitive radio networks.

Intelligent Systems in Agriculture

An Efficient Approach for Plant Leaf Species Identification Based on SVM and SMO and Performance Improvement

Sahil Vyas, Manish Kumar Mukhija, and Satish Kumar Alaria

Abstract Plants are necessary for life to exist on this planet. There are several plant species to pick from, and the number of plant species is increasing every year. Programmed species identification has a variety of advantages over traditional species identification. The majority of plant-programmed identification methods today concentrate on leaf morphology, venation, and surface traits, which have shown to be useful in recognising some plant species. Due to their year-round availability, especially in tropical climates, leaves are widely utilised in plant species identification. In terms of shape, texture, breathability, and colour, a single leaf can serve a number of purposes. A variety of methodologies, such as classic moral assessment methods or machine learning, can be employed to extract these functions. For non-specialists who have little or no comprehension of common natural notions, this, on the other hand, demands expert knowledge and becomes a time-consuming and challenging task. Nonetheless, advancements in the fields of machine learning and computer vision may be able to assist in making this labour more doable. This study provided an effective approach for recognising plant leaf species based on SVM and SMO. A random data set was used to select the leaves. The performance is assessed using accuracy, error rate, and other measures. Nonetheless, advances in the domains of machine learning and computer vision can help to make this work more manageable. Based on SVM and SMO, this research proposed an efficient method for identifying plant leaf species. The leaves were chosen at random from a random data set. The accuracy, error rate, and other metrics are used to evaluate the performance.

Keywords Plant · Leaves · Species · SVM · SMO · Accuracy · Error rate

S. Vyas (✉) · M. K. Mukhija
Department of Computer Science and Engineering, AIET, Jaipur, India
e-mail: sahilvyas1988@gmail.com

S. K. Alaria
Department of Electronics and Communication, AIET, Jaipur, India
e-mail: satish.alaria@gmail.com

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_1

1 Introduction

While modern agronomy and plant breeding have increased development production, they also resulted in widespread environmental harm and poor human health. Automated PC vision is critical in this case because it helps repair the damage. The appearance of plant leaves changes over time, making them legitimate elements. It is a challenge to automate the identification and arrangement of plant leaves, because plant leaves are genuine elements. However, scientists use a number of common characteristics to distinguish between various leaf tests, such as the state of the leaf, its shade, its skeleton, and so on. There are many aspects to consider when designing a plant's arrangement, and this project is centred on the surface characteristics of a leaf. A plant leaf's veins are another important feature. Vein extraction and grouping can also be used to determine a plant's type. There are some drawbacks to shape-based leaf recognition [1, 2], especially if there are not enough leaf tests.

In the name of Kumar [1], a multi-facet perceptron with AdaBoosting was used in this study to classify plant species based on their morphological features. The proposed framework includes pre-processing, extraction, highlighting, and order. Different pre-processing methods are used to prepare a leaf image for the element extraction measure. Several morphological parameters, such as centroid, significant hub length, minor hub length, firmness, edge, and orientation, are retrieved from digital photographs of various leaf classes. In the suggested machine learning classifier, 95.42% of the computations were accurate. Tan and Colleagues [2] Morphometric calculations based on Sobel segmented veins were employed as a benchmark for this study. The D-leaf model achieved a testing accuracy of 94.88% when compared to the AlexNet (93.26%) and adjusted AlexNet (95.54%) models. Furthermore, CNN models outperformed traditional morphometric estimations (66.55%). The CNN-derived properties fit well with the ANN classifier. Using computer-aided image processing and gadget vision innovation, Singh [3] outlines a fundamental and computationally restricted technique for plant identification. Leaf data is preprocessed, a large data set is analysed for correlations, and the data is organised into a logical sequence. Pre-preparation is the process of enhancing estimates pix prior to computational treatment. The leaf picture's shade and architecture are used to identify genuine elements by the limit extraction piece. Paulson [4] convolutional neural network (CNN) and pre-trained VGG16 and VGG19 models were compared for the leaf recognition task. The data set has leaf images of 64 medicinal plants. CNN achieved a 95.79% accuracy rate. VGG16 and VGG19 beat fundamental CNN in terms of accuracy, with 97.8% and 97.6%, respectively.

Yang [5] An object's bending and geographic data are represented by a triangular place distance grid, while an object's arched/sunken characteristic is defined by a sign framework. Detailed and global properties of the leaf form are captured while the likeness changes are preserved (interpretation, revolution, and scaling). It is also quite easy to implement and has a modest computational burden. Four well-known data sets of plant leaves, including the Swedish, Smithsonian, Flavia, and Image CLEF 2012 databases, were used to test our method. A comparison of the results shows that our method outperforms other leading edge methods of plant leaf detection based on shape. Using the MPEG-7 shape data set as a second test, we found that our method can be applied to a wide range of shapes. In the words of Zhou [6] because the unique RDN model was first used in image super objective, we need to rebuild the network engineering for grouping activities through modified info image attributes and hyper boundaries. In artificial intelligence, the tomato test data set. Challenging 2018 data sets demonstrate that this model can achieve a best 1 normal recognition accuracy of 95%, which confirms its appetising exhibition. The rebuilt residual dense network model can deliver considerable improvements over a large portion of the best-in-class models, as well as requiring less calculation to attain elite outcomes in crop leaf recognition. Almeida [7] proposes a new technique to plant recognition based on the usage of an unsupervised distance learning strategy instead of pairwise similitudes. They demonstrate a significant increase in performance when using it for time series search workloads. Furthermore, we demonstrate how the late mixing of several time series can affect the identification of plant species. Increases in precision at 10 scores and mean normal precision were both up to 8.21% and 19.39% higher when compared to when time series were used in the disengagement process, respectively.

Using a visual tree to filter through a large number of plant species and determine the relationship between related learning tasks, Fan [8] proposes that a progressive conduct multiple tasks primary learning calculation can be used to enable the identification of a wide variety of plant species. Plants are essential for human survival, according to Azlah et al. [9]. Since the dawn of time, indigenous peoples have relied heavily on herbs as folk medicines. For practitioners, identifying herbs is frequently a matter of years of olfactory or sensory training. Recent advances in analytical technology have substantially aided scientific data-based herbal identification. People who are not familiar with herbal identification can benefit greatly from this. Testing in a laboratory requires expertise in sampling and data analysis in addition to lengthy procedures. In this way, an easy and reliable way to identify herbs is required. Plant identification is anticipated to benefit from a combination of computing and statistical analysis. Identifying herbs rapidly is made easier by this non-destructive method, especially for those who cannot afford pricey analytical equipment. To identify plant species from images of plant leaves, Mahmudul Hassan [10] used the DL approach. By extracting and learning features automatically, DL identification eliminates the need to manually remove created attributes. After each cycle, the recognition rate increases, but the loss decreases. As a result of this research, CNN performs better in plants with comparable leaf shapes and sizes. Aside from that, more study may be done to test the proposed network's performance utilising real time and drone images. A cloud-based architecture will allow smart devices to identify plant types in real time. Plants are difficult to manually identify by botanists, according to Bojamma [11], who found that different plant features used as identification keys take a long time. Inquiries concerning the quality of a candidate search that is constantly modified are answered by these keys. Because the traditional method of plant species identification is a tired method that has the potential to be false and is out of reach for the general public, it poses a challenge for botanists, nature enthusiasts, and people dealing with botanical challenges in their daily lives, such as farmers and conservationists.

2 Methodology

Figure 1 depicts the proposed flow chart. A random collection of plant species is obtained from Google and used to define the input configuration. Listed below are the flow chart's steps:

1. The leaf picture collection is utilised as input. A.jpg or.png file must be used as the source image.
2. The photographs acquired for input are subjected to pre-processing. In the pre-processing stage, we can employ image rescaling.
3. Extracting plant species pattern characteristics is a major part of feature extraction.
4. The spider monkey optimization (SMO) approach is utilised to optimise the features in this process.
5. The support vector machine (SVM) is used to classify the leaf species at this point in the process.
6. The accuracy, sensitivity, and error rate of the system are all estimated in this module.

Algorithm

Input: Random plant leaf data set as input

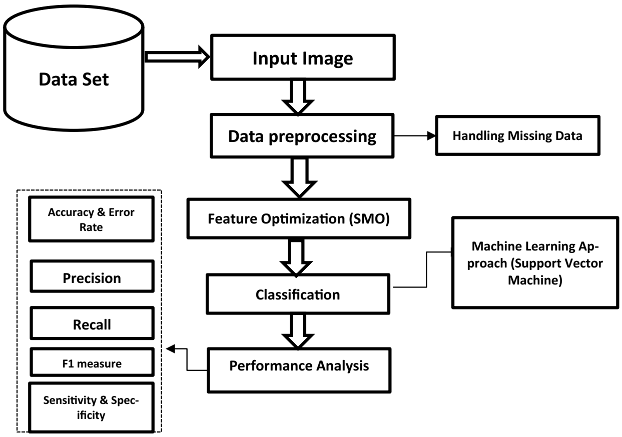


Fig. 1 Flow chart

R, G, B colour transformation, patch segmentation, image resizing, channel separation the null value are being filtered out.

Output: Achieves the highest possible levels of accuracy and precision with the least amount of noise.

Step 1: X train, X test, y train, y test data sets are split into separate training and testing data sets.

Step 2: Extracting features from a picture in flat, edge, and corner.

Step 3: SMO, the SMO class, should be used ()

Calculate Fitness (self, fun1)

smo.initialize ()

smo.GlobalLearning ()

smo.LocalLearning ()

Step 4: Using the machine learning classifier support vector machine, perform the fourth step

class = "train features,labeled" > CLASSES = clf.fit

(clf.predict(Train features); AN = AN)

Step 5: Create a confusion matrix and show the values of TP, FP, TN, and FN in it.

Step 6: Accuracy and other metrics are calculated at this step as well.

3 Spider Monkey Optimization Algorithm (SMO)

There are several similarities between SMO and population-based algorithm. Interaction between a local leader and a global leader is divided into six phases: local leader learning; global leader learning; local leader decision-making; and global leader decision-making. Alternate variants of the Gbest-guided artificial bee colony (ABC) [7] By relying on experimentation, SMO shares some similarities with algorithms and computations that use population data. Local leader, global leader, local leader phase of learning, global leader phase of learning local leader phase of decision, and global leader phase of decision are the six phases of SMO interactivity. During the global leader phase, the Gbest-guided artificial bee colony (ABC) and a modified version of ABC are employed to update the position of the leader. The SMO execution process is explained in greater depth in the following sections:

These sections go into the various steps involved in putting an SMO plan into action.

3.1 Population Initiation

Figure out how much money you have in your bank account, or how much money you have in your bank account, or how much money you have in your bank account and how much money you have in your bank account, use this method (1).

$$SM_{ij} = SM_{\min\ j} + U(0, 1) \times \left(SM_{\max\ j} - SM_{\min\ j}\right) \tag{1}$$

The functions $SM_{\min j}$ and $SM_{\max j}$ provide the boundaries of $U(0, 1)$, a random number with uniform distribution in the range [0, 1].

3.2 Local Leader Phase (LLP)

In the local leader phase, every spider monkey SM changes its position based on local leader experience, as well as that of the local gathering component. This position's happiness quotient is established. When the new position's welfare value exceeds that of the previous position, the SM prefers to move. This step necessitates constant communication with the ith SM (a representative of the kth local gathering).

$$SM_{new\,ij} = SM_{ij} + U(0, 1) \times \left(LL_{kj} - SM_{ij}\right) + U(-1, 1) \times \left(SM_{rj} - SM_{ij}\right) \tag{2}$$

This position has two dimensions: one for each local gathering leader and one for each local gathering leader's role in SM_{iJ}. For each gathering, random dimensions of the R-SM are chosen at random, so that the SM_{rj} dimension contains the $r = i$, $U(0, 1)$ random number between 0 and 1. Calculation 1 shows the local leader phase position updates measure. There are many gathers, MG, and a rate of irritation that affects how uncomfortable a person is in their current position. Pr can only measure values between [0.1, 0], it is a good idea [0.9].

3.3 Global Leader Phase (GLP)

Immediately after the local leader phase comes the global leader phase (also known as GLP). They use their experience to update all SM positions at this time, including the global leader (GL) and the local gathering part (LGP). Thus, the position update condition for this phase can be summarised as the following: $U(0, 1) + U(GL_j - SM_{ij}) + U(0, 1) + U(1, 1) - (SM_{rj} - SM_{ij})$ (3) where GL_j represents the position of the global leader in the jth dimension, and $j - 1, 2, \ldots, D$ is the random index.

$$SM_{new\,ij} = SM_{ij} + U(0, 1) \times \left(GL_j - SM_{ij}\right) + U(-1, 1) \times \left(SM_{rj} - SM_{ij}\right) \tag{3}$$

where GLj represents the jth dimension of the global leader position and $j \in \{1, 2, \ldots, D\}$ is the randomly chosen index.

$$\text{Probi} = 0.9 \times (\text{fitness}/\text{max\_fitness}) + 0.1, \tag{4}$$

4 Support Vector Machine

With the use of a separator, an educational index can be narrowed down into at least two categories using SVM, a type of supervised calculation. Despite the fact that the data are not all that different, SVM sorts them via rearrangement into a high-dimensional segment space (a process known as kernelling) (Fig. 2).

It is the normal vector to the hyperplane, and the input vector that produces a linear SVM output.

It is defined by the distance between the hyperplane and the closest positive and negative examples in the linear case. It can be represented as an optimization problem: Minimise subject to where x_i is the ith preparation model and yi the right yield of SVM for ith preparation. It is determined as follows: where, the input vector, and normal to hyperplane's hyperplane's normal vector distance from hyperplane to nearest positive or negative example determines margin in linear scenario. x_i is the ith preparing model, and y_i is the appropriate yield of the SVM for the ith preparing. Minimise as follows: It is a difficult quadratic optimization issue to get the support vectors x_i (y_i) and their coefficients, as well as a threshold b, in SVM preparation. When solving a simple two-class problem, the support vector order concentrates on the identification of a working choice. A test event is classified by the decision-making that goes along with it.

$$\phi(x) = \mathrm{sgn}\left(\sum_j \alpha_i y_i K(x_i, x_j) + b\right) \tag{5}$$

The kernel function, $K(x_i, x_j)$, determines the functional shape of the support vectors; as a result, different kernels behave differently. In the following list, you can

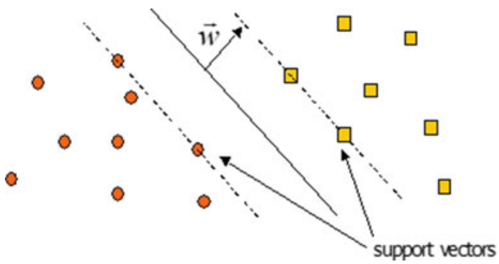


Fig. 2 Support vector machine

find some of the most common kernels:

$$\text{Linear}: K(X, Y) = X.Y$$
$$\text{Polynomial}: K(X, Y) = (1 + X.Y)^d$$
$$\text{Radial Basis function}: K(X, Y) = \exp(-||X - Y||^2/2\sigma^2) \quad (6)$$

5 Simulation Results

Python Spyder 3.7 is used to implement the algorithm under development. With the help of libraries like sklearn and Python, we can apply numerous approaches like decision trees and Naive Bayes in Spyder, as well as the functions available in the Spyder environment.

Figure 3 is showing the plant leaf image input data. Total 32 images taken with 7 different species, which include apple, banana, ginger, mango, Norway maple, onion, and work mulberry (Figs. 4, 5 and 6).

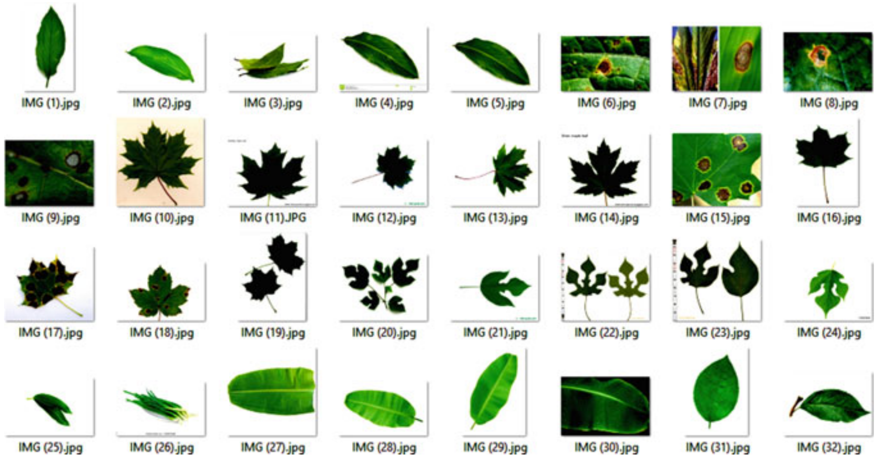
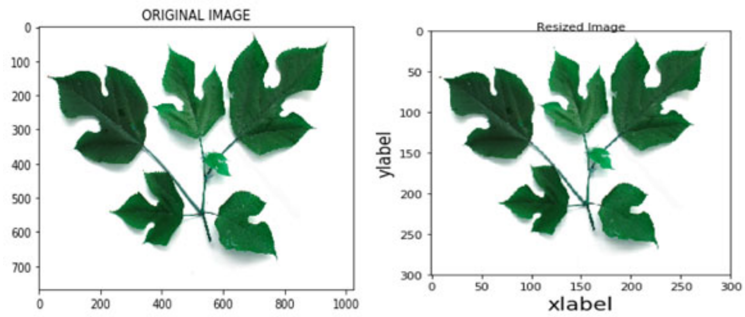


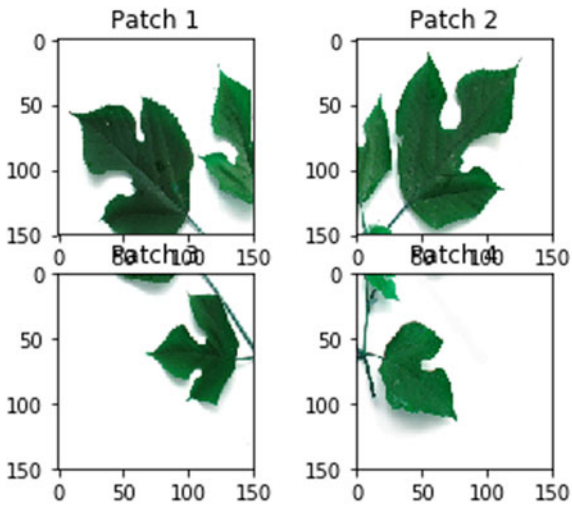
Fig. 3 Input data



(a) (b)

Fig. 4 **a** Original input image **b** resized image

Fig. 5 Patch segmentation subplot



Confusion matrix

[[50 0]

[2 48]]

TP = 50

FP = 0

TN = 2

FN = 48

Identified as work mulberry.

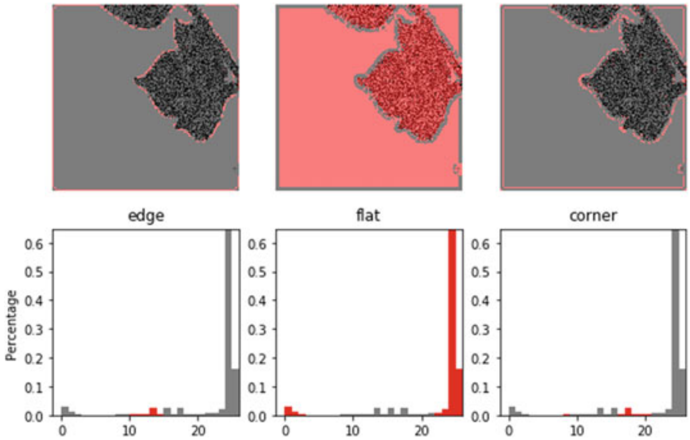


Fig. 6 Image features

Table 1 is presenting all the simulation parameters after calculating by using standard formulas (Table 2).

Table 1 Simulation result

S. No.	Parameters	Values (%)
1	Accuracy	98
2	Classification error	3
3	Precision	100
4	Recall	96
5	F-measure	97
6	Sensitivity	96
7	Specificity	100

Fig. 7 Comparison of the accuracy

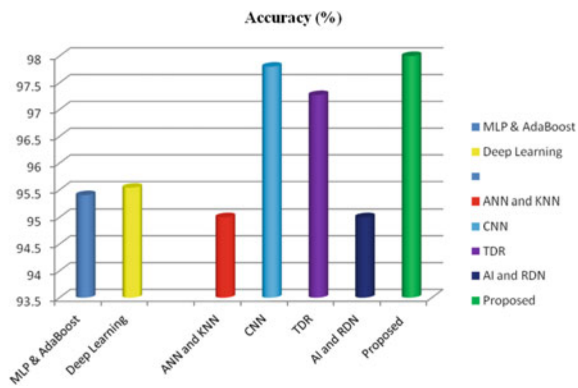


Table 2 Comparison chart of proposed work with literature and existing approach

S. No.	Author name	Method	Precision (%)	Recall (%)	*F*-1 (%)	Accuracy (%)	Error rate (%)
1	M. Kumar et al.	Multilayer perceptron	95.08	NA	NA	95.4	4.6
2	Jing Wei Tan et al.	Deep learning	NA	NA	NA	95.54	4.46
3	G. Singh et al.	ANN and KNN	NA	NA	NA	95	5
4	A. Paulson et al.	CNN	NA	NA	NA	97.8	2.2
5	C. Yang et al.	TDR	97	97	NA	97.27	2.73
6	C. Zhou et al.	AI and RDN	NA	NA	NA	95	5
7	Sahil et al.	Proposed (SVM and SMO)	100	96	97	98	02

Comparing the correctness of the reference with the proposed work is shown in Fig. 7. The accuracy and error rate of the proposed approach were clearly superior than those of the previous approach.

6 Conclusion

This research proposes a plant species identification method that orders plant leaf. In this study, computer vision and machine learning techniques are used to sort photographs of plant leaves in order to develop a method for identifying plant species. Pre-processing, image division/extract/arrangement were all stages, in which the research was carried out. In order to extract picture features such as flat, edge, and corner, a support vector machine (SVM) classifier was used, which was subsequently classified using the spider monkey optimization (SMO). A random data set was used to test the system, and it turned out to be 98% accurate. The image colour and feature space worked admirably in compared to the previous approaches. In addition, SVM beats other classifiers including CNN, ANN, and TDR. The proposed method can be implemented in a relatively short period of time. Take into account additional data sets in future, along with the use of unique techniques like a practical model for real-time prediction.

References

1. Kumar M, Gupta S, Gao X, Singh A (2019) Plant Species recognition using morphological features and adaptive boosting methodology. IEEE Access 7:163912–163918. https://doi.org/10.1109/ACCESS.2019.2952176
2. Tan JW, Chang S-W, Abdul-Kareem S, Yap HJ, Yong K-T (2020) Deep learning for plant species classification using leaf vein morphometric. In: IEEE/ACM transactions on computational biology and bioinformatics, vol 17, no 1, pp 82–90, 1 Jan–Feb 2020. https://doi.org/10.1109/TCBB.2018.2848653
3. Singh G, Aggarwal N, Gupta K, Misra DK (2020) Plant identification using leaf specimen. In: 2020 11th International conference on computing, communication and networking technologies (ICCCNT), pp 1–7. https://doi.org/10.1109/ICCCNT49239.2020.9225683
4. Paulson A, Ravishankar S (2020) AI Based indigenous medicinal plant identification. Adv Comput Commun Technol High Perform Appl (ACCTHPA) 2020:57–63. https://doi.org/10.1109/ACCTHPA49271.2020.9213224
5. Yang C, Wei H (2019) Plant species recognition using triangle-distance representation. IEEE Access 7:178108–178120. https://doi.org/10.1109/ACCESS.2019.2958416
6. Zhou C, Zhou S, Xing J, Song J (2021) Tomato leaf disease identification by restructured deep residual dense network. IEEE Access 9:28822–28831. https://doi.org/10.1109/ACCESS.2021.3058947
7. Almeida J, Pedronette DCG, Alberton BC, Morellato LPC, Torres RDS (2016) Unsupervised distance learning for plant species identification. IEEE J Sel Top Appl Earth Observations Remote Sens 9(12):5325–5338. https://doi.org/10.1109/JSTARS.2016.2608358
8. Pereira CS, Morais R, Reis MJCS (2019) Deep learning techniques for grape plant species identification in natural images. Sensors 19(22):4850
9. Yujian EC, Njuki L, Yingchun S, Liu (2018) A comparative study of fine-tuning deep learning models for plant disease identification. Comput Electron Agric S0168169917313303. https://doi.org/10.1016/j.compag.2018.03.032
10. Fu L et al (2019) Lycoris species identification and infrageneric relationship investigation via graphene enhanced electrochemical fingerprinting of pollen. Sens Actuators B: Chem 298:126836
11. Rangarajan AK, Purushothaman R, Ramesh A (2018) Tomato crop disease classification using pre-trained deep learning algorithm. Procedia Comput Sci 133:1040–1047. https://doi.org/10.1016/j.procs.2018.07.070
12. Huixian J (2020) The analysis of plants image recognition based on deep learning and artificial neural network. IEEE Access 8:68828–68841. https://doi.org/10.1109/ACCESS.2020.2986946
13. Pankaja K, Thippeswamy G (2017) Survey on leaf recognization and classification. In: [IEEE 2017 international conference on innovative mechanisms for industry applications (ICIMIA)—Bengaluru, India (2017.2.21–2017.2.23)], 2017 International conference on innovative mechanisms for industry applications (ICIMIA)—survey on leaf recognization and classification, pp 442–450. https://doi.org/10.1109/ICIMIA.2017.7975654
14. Wäldchen J, Mäder P (2018) Plant species identification using computer vision techniques: a systematic literature review. Arch Computat Methods Eng 25:507–543. https://doi.org/10.1007/s11831-016-9206-z
15. Kaur S, Kaur P (2019) Plant species identification based on plant leaf using computer vision and machine learning techniques. J Multimedia Inf Syst 6:49–60. https://doi.org/10.33851/JMIS.2019.6.2.49
16. Bao TQ, Kiet NTT, Dinh TQ, Hiep HX (2020) Plant species identification from leaf patterns using histogram of oriented gradients feature space and convolution neural networks. J Inf Telecommun 4(2):140–150. https://doi.org/10.1080/24751839.2019.1666625
17. Hassan SM, Maji AK, Jasiński M, Leonowicz Z, Jasińska E (2021) Identification of plant-leaf diseases using CNN and transfer-learning approach. Electronics 10:1388. https://doi.org/10.3390/electronics10121388

18. Xiong J, Yu D, Liu S, Shu L, Wang X, Liu Z (2021) A Review of plant phenotypic image recognition technology based on deep learning. Electronics 10:81. https://doi.org/10.3390/electronics10010081
19. Malarvizhi K, Sowmithra M, Gokula Priya D, Kabila B (2021) Machine learning for plant species classification using leaf vein morphometric. Int J Eng Res Technol (IJERT) 10(04)
20. Kaur S, Kaur P. Plant Species identification based on plant leaf using computer vision and machine learning techniques. J Multimedia Inf Syst 30;6(2):49–60 [Internet]. Available from: https://doi.org/10.33851/JMIS.2019.6.2.49
21. Rzanny M, Mäder P, Deggelmann A et al (2019) Flowers, leaves or both? How to obtain suitable images for automated plant identification. Plant Methods 15:77. https://doi.org/10.1186/s13007-019-0462-4
22. Pärtel J, Pärtel M, Wäldchen J (2021) Plant image identification application demonstrates high accuracy in Northern Europe. AoB Plants 13(4):plab050. https://doi.org/10.1093/aobpla/plab050
23. Yang B, Xu Y (2021) Applications of deep-learning approaches in horticultural research: a review. Hortic Res 8:123. https://doi.org/10.1038/s41438-021-00560-9
24. Hassan E, Shams M, Hikal NA, Elmougy S (2021) Plant seedlings classification using transfer learning. Int Conf Electron Eng (ICEEM) 2021:1–7. https://doi.org/10.1109/ICEEM52022.2021.9480654
25. Dhaka VS, Meena SV, Rani G, Sinwar D, Kavita, Ijaz, MF, Woźniak M (2021). A survey of deep convolutional neural networks applied for prediction of plant leaf diseases. Sensors (Basel, Switzerland) 21(14):4749.https://doi.org/10.3390/s21144749
26. Koul D, Alaria SK (2018) A new palm print recognition approach by using PCA & Gabor filter." *International Journal on Future Revolution in Computer Science & Communication Engineering* 4, no. 4 (2018): 38–45.
27. Khandelwal, Ravi, Mukhija MK, Alaria SK (2021) Numerical simulation and performance assessment of improved particle swarm optimization based request scheduling in edge computing for IOT applications. NEW ARCH-Int J Contemp Archit 8(2):155–169. http://the-new-arch.net/index.php/journal/article/view/221
28. Najneen, Mukhija MK, Kumar S, RAFI: parallel dynamic test-suite reduction for software. In: Publication in the book computational intelligence in analytics and information systems, vol 2. Advances in Digital Transformation published by Apple Academic Press, Co publishing with CRC press, a Taylor Fransis Group
29. Pooja, Mukhija MK, Alaria S (2021) An approach for cloud security using TPA and role based hybrid concept. In: 3rd International conference on computing, communication and cyber security (IC4S-2021), indexed in scopus springer lecture notes 30–31st Oct 2021, at KEC, Ghaziabad
30. Bhargav M, Alaria SK, Mukhija MK (2021) Implementation of sentiment analysis and classification of tweets using machine learning. Turk Online J Qual Inquiry (TOJQI) 12(10):9–21. e-ISSN 1309-6591

Smart IoT-Based Pesticides Recommendation System for Rice Diseases

Sumit Kumar, Rutuja Rajendra Patil, and Ruchi Rani

Abstract India is the land of farming, and agriculture in the country plays the most crucial role in employment and economic development. Crops provide us with food and other sources that help make our lives sustainable. The proposed system is an automated Internet of things (IoT)-based system developed to measure moisture, temperature and humidity levels, and soil pH levels. With the help of this model, best pesticides are predicted for the diseases of rice crops. In this paper, the Random Forest algorithm is applied and found to give good accuracy compared to other machine learning algorithms.

Keywords Internet of Things · Pesticides · Crop · Random forest algorithm

1 Introduction

Agriculture is a primary sector in India because it contributes 18% to our GDP and even employs more than 50% of the Indian workforce. Taking care of agriculture should be our primary objective because our country is also the biggest exporter in the current Asian market in terms of agricultural goods. Taking care of agriculture does not only mean that we should keep growing them and just watering them because even plants/crops are living organisms who need additional good sources of energy and nutrients for themselves to exist. Things required by plants are to be observed from the condition of plants/crops or the soil in which it is grown. The most important characteristics observed to identify plant diseases are the leaves of plants, the roots, and the soil parameters such as temperature, humidity, moisture, and soil nutrients. We can change all of it given above if required for the growth of plants except for a few parameters like humidity and moisture. However, the most critical parameter

S. Kumar (✉) · R. R. Patil
Symbiosis Institute of Technology (SIT), Symbiosis International (Deemed University), Pune, Maharashtra, India
e-mail: er.sumitkumar21@gmail.com

R. Rani
Department of Computer Science, Indian Institute of Information Technology, Kottayam, Kerala, India

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_2

is the nutrients present in its soil because that would help us determine whether the soil requires more nutrients according to the type of crops. If proper nutrients are not given to plants, we will suffer huge losses like scarcity, inflation, etc. According to The United Nations Food and Agriculture Organization (FAO), more than 40% of food produced is wasted in India, and its costs could be as high as US$14 billion (12.42 billion euros) every year. India is a country which goes through all kinds of climates, winter, summer, rainy, etc., and primarily because most of the time has summer with high hot temperatures, which increases the amount of workload on farmers to overlook their crops manually and take care of them by watering them and providing additional nutrients to them for their better well-being. The average size of farming land over India is over 1.2 ha, which are to be governed and taken care of by very few farmers manually, which is practically impossible, leading to wastage of crops and damaging soil. Now, this is where industrialization is required, and modern methods are needed to be taken to tackle these problems.

2 Literature Survey

The authors in [1] presented a study on rice's most grown staple food of all time overall worldwide. Their system is used to predict diseases that could affect the rice crops, mainly contributed by paddy diseases. Due to so many different variants of paddy diseases, it is practically challenging to remove or cure them physically. In this paper [2], the researchers present a system for suitable yield for the farmer and detect pests that can affect and promote pest control strategies. In our agriculture section, most of the crops are getting lost due to the wrong choice of crop to be planted in a particular area. The researchers in [3] present paper's objectives to explore several methods in machine learning and conventional deep neural network to identify a given set of diseases in rice plants. In this proposed system, plants images are used to analyze whether the plant is infected or not. This paper also shows its research by comparing and analyzing other networks and algorithms used to achieve successful results in detecting rice crop diseases. The article [4] focuses on combining all advanced approaches for precision agriculture use. People worldwide use technologies like Internet of Things (IoT), artificial learning, and deep learning for modern agriculture methods. The authors in [5] present a work of their own regarding a particular disease known as rust, which is very common in plants and crops formed due to not taking proper care of it. The authors in [6] propose spraying pesticides over a farm using emerging technologies instead of manually. The reason is supported because manually excess or lack of pesticides could be given or spread, and in both cases, humans and plants are suffering loss.

3 Methodology

The proposed system has five different layers: an input layer, sensing layer, artificial intelligence layer, a data visualization layer, and an application interface, as shown in Fig. 1.

First, for the beginning of working of the system, a farming ground is selected so that different data could be taken out from it like the surrounding soil, pH, humidity, temperature, and the moisture content of the soil which is going to be used for sowing seeds of plants [7]. Then the second part comes in, where all these data are accumulated and collected over a cloud database and then could be used to analyze the derived data. These data are then exported over to the algorithm made by using Random Forest as it provides a great analysis based on the weighted average. These data are then explored, and after analyzing of different correlations found between different parameters, we were able to develop a scientific model that could be used to predict whether a plant or a crop has any disease or not or whether it could get infected by some sort of disease in the future based on the current characteristics collected over by the help of knowing the nature of the soil, its surrounding temperature, humidity, and especially its pH and soil moisture content [8].

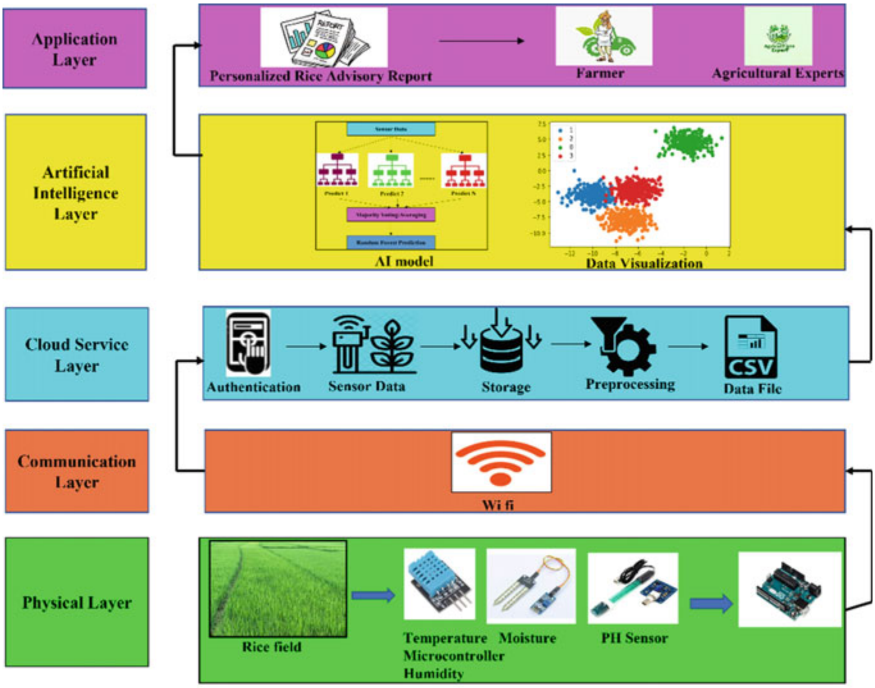


Fig. 1 Block diagram of proposed system

This feature also helps to know whether the water given or provided is less or much more than the substantial amount required for that crop. This proposed system is basically built up now for rice crops as they are heavily sown in India and all over the world. After creating a model used now for the interface environment, Spyder environment is used, an application to deploy models to have an interface with the user so that a person could use the system in real time [9]. Python model was deployed and used without any interference. For the interfacing layout, HTTP was used as they allow the user to access it from anywhere, and its customization can be further done.

3.1 Materials and Methodology

The proposed system basic electronics engineering approach is used for combining it with IoT and artificial intelligence. Apart from just working, we have also taken care of its power efficiency by using such components that will not dissipate as much power as compared to many other components in the market [10–14]. For this study, many sensors were required to fulfill the requirements of getting data based on real time which consisted of the real-time temperature value, the moisture content of the soil, the most important factor of our project the pH of the soil so that acidity or alkalinity of the soil could be determined and by using it system could recommend the soil for the required pesticides [15–17]. The proposed system has five major components shown in (Figs. 2 and 3).

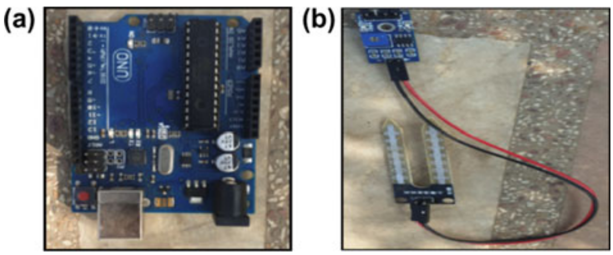


Fig. 2 **a** Arduino UNO and **b** soil moisture sensor

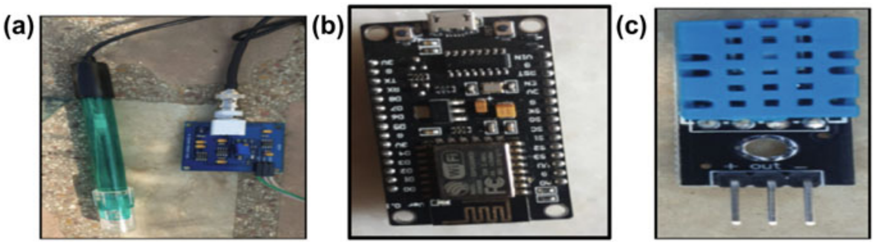


Fig. 3 **a** PH sensor, **b** Wi-Fi module ESP8266, and **c** DHT 11

4 Results

The result obtained was highly effective as the proposed model's accuracy is high and it predicted most of the pesticides correctly as per the data collected. The data collected over 100 data samples with their pH values, temperature, moisture, and humidity level at different times like in daylight; it was observed that the temperature of the surroundings and the soil were almost from 25 to 27 °C. These data did not show many variations from nearby places. These data were then used by Random Forest algorithm to get our model ready, which could predict the kind of diseases that could be there or could possibly affect plants or crops health in the future and could also recommend which kind of pesticides are needed to be used for the cause. Now, these were the major points where the data were collected from sensors. All the data collected were similar and took an average of data collected and used the Random Forest algorithm to test and train the dataset. The crops in this region are most of the rice and are grown over here from June to July and harvested over the month of November and December. So, by getting the real data from the fields, we predicted highly accurate results that were useful during the growth and harvesting of this crop (Fig. 4).

4.1 Evaluation Metrics

The major evaluation metrics used were accuracy and losses the model suffered through the entire training and testing. The model our proposed system has is based

Fig. 4 Collection of readings from rice farms

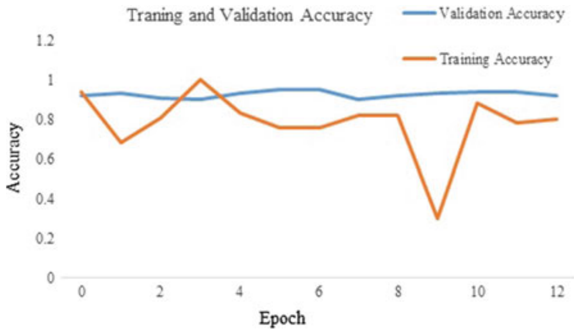


Fig. 5 Training and validation accuracy

on the Random Forest algorithm, which performs a weighted evaluation to give a metric-based performance and a high accuracy for the working of our proposed model.

The graph in Fig. 5 depicts the overall training accuracy, and the validation accuracy is obtained using the Random Forest algorithm. The validation accuracy does not have a standardized value; however, the obtained accuracy was always more than 93 percent, which was good in measuring the amount and type of pesticide required in the training accuracy. In this model testing, the training and validation loss was also tested. We observed that the training loss was never more than 0.25 and the validation loss was a little high; however, it was a lot less than two throughout the entire testing and performance.

4.2 Sensors Results and Readings

In this, the data were recorded from the actual soil of the farm of rice. The number of readings collected was more than 100 and was a peculiar pesticide recommendation source. This data were recorded with the help of a soil moisture sensor and has been able to see through the Arduino at a baud rate of 9600. As per the sensor data, the pH of the soil recorded from the soil sample taken from the rice farm is always between 7.4 and 7.5 range, which was ideally an excellent soil for the growth of rice crops, and the soil had an essential alkaline nature according to the pH observed. Approximately 100 readings were observed and calculated using the sensor. The data from the temperature sensor show the temperature and humidity observed throughout day and night from the soil of the rice crop. The temperature varied from 27 to 25 °C, and the humidity of the soil varied from 49 to 63 levels. Approximately 100 readings were observed and calculated using the sensor.

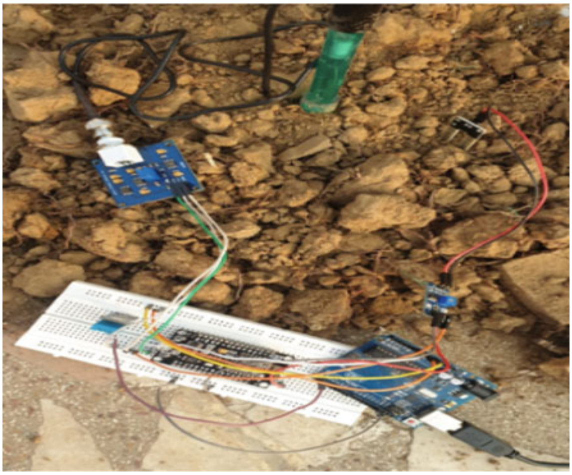


Fig. 6 Hardware prototype of the circuit

Figure 6 is the entire final circuit which is used to take the valuable data from temperature, pH, moisture, humidity and transfer the same data to the IoT network through Wi-Fi module ESP8266 and then these datasets are used to make a model based on Random Forest algorithm, and an application interface is provided to know whether the plant would need any pesticide or not.

Figure 7 demonstrates the system's working by giving the respective inputs as pH, humidity, crop week, temperature, and moisture. When the predicted key is pressed, the recommendations are provided, as shown in the image. In the final prediction part of the proposed work, an application was designed to transform the model made in Python and offer it with an actual user interface. The application was coded to take input as the number of weeks of crops, humidity of the soil, the pH of the ground, and the temperature of the earth in which harvest is sown. When the predict option is clicked, the results are shown to know which pesticide to use for the predicted disease.

5 Conclusion

In India, the significant crops sown are rice and used to feed many people. The proposed system is a working application that is beneficial for the farmers. Using data collected over rice farms, based on real time, the proposed approach for farmers to protect crops and create an environment with less scarcity of foods. An application layered platform has been built to predict the condition of rice crops that are sown in July and harvested over December and January. This was a successful approach for knowing the condition of a rice crop by using data like pH, the moisture level of the soil it is sown in, the temperature of its surroundings, and the region's humidity. From the working of the system, it was inferred that the accuracy provided by the

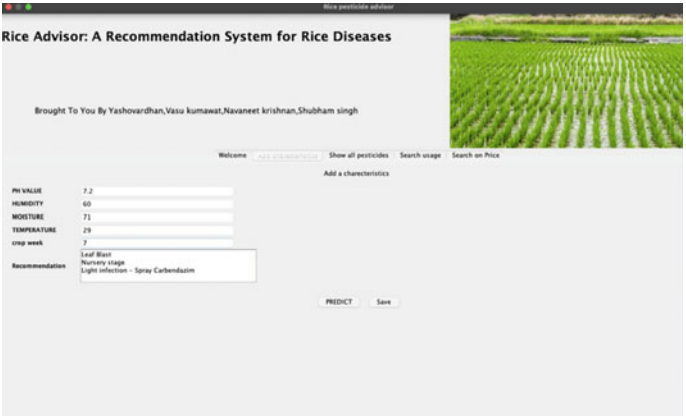


Fig. 7 The application for user interface

system was 95.4% and was able to detect most of the diseases and pesticides required to cure it. Our future work aims to develop this model with more soil attributes and a more extensive dataset.

References

1. Pinki FT, Khatun N, Islam SMM (2018) Content based paddy leaf disease recognition and remedy prediction using support vector machine. In: 20th International conference of computer and information technology ICCIT 2017, vol 2018-January, pp 1–5. https://doi.org/10.1109/ICCITECHN.2017.8281764
2. Kumar A, Sarkar S, Pradhan C (2019) Recommendation system for crop identification and pest control technique in agriculture. In: Proceeding of 2019 IEEE international conference on communication and signal processing ICCSP 2019, no April 2019, pp 185–189. https://doi.org/10.1109/ICCSP.2019.8698099
3. Agrawal MM, Agrawal S (2020) Rice plant disease detection and classification using deep learning models: a systematic review. J Crit Rev 7(11):4376–4390
4. Garg S, Pundir P, Jindal H, Saini H, Garg S (2021) Towards a multimodal system for precision agriculture using IoT and machine learning. https://doi.org/10.1109/icccnt51525.2021.9579646
5. Sood S, Singh H (2020) An implementation and analysis of deep learning models for the detection of wheat rust disease. In: Proceeding of 3rd international conference on intelligence sustainability system ICISS 2020, no December 2020, pp 341–347. https://doi.org/10.1109/ICISS49785.2020.9316123
6. Shaw KK, Vimalkumar R (2020) Design and development of a drone for spraying pesticides, fertilizers and disinfectants. Int J Eng Res V9(05):1181–1185.https://doi.org/10.17577/ijertv9is050787
7. Talaviya T, Shah D, Patel N, Yagnik H, Shah M (2020) Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides. Artif Intell Agric 4:58–73. https://doi.org/10.1016/j.aiia.2020.04.002

8. Kalimuthu M, Vaishnavi P, Kishore M (2020) Crop prediction using machine learning. In: Proceeding 3rd international conference on smart system inventive technology ICSSIT 2020, no ICSSIT, pp 926–932. https://doi.org/10.1109/ICSSIT48917.2020.9214190
9. Patil R, Kumar S (2020) A bibliometric survey on the diagnosis of plant leaf diseases using artificial intelligence. Libr Philos Pract 2020:1–25
10. Patil RR, Kumar S (2021) Predicting rice diseases across diverse agro-meteorological conditions using an artificial intelligence approach. Peer J Comput Sci 7:1–25. https://doi.org/10.7717/peerj-cs.687
11. Kumar S, Patil RR, Kumawat V, Yashovardhan N, Krishnan, Singh SK (2021) A bibliometric analysis of plant disease classification with artificial intelligence using convolutional neural network. Libr Philos Pract 2021:1–14
12. Patil RR, Kumar S (2022) Priority selection of agro-meteorological parameters for integrated plant diseases management through analytical hierarchy process. Int J Electr Comput Eng 12(1):649–659. https://doi.org/10.11591/ijece.v12i1.pp649-659
13. Andrianto H, Suhardi, Faizal A et al (2021) Performance evaluation of IoT-based service system for monitoring nutritional deficiencies in plants. Inf Process Agric. https://doi.org/10.1016/j.inpa.2021.10.001
14. Kumar S, Mehfuz S (2019) A PSO based malicious node detection and energy efficient clustering in wireless sensor network. In: 2019 6th International conference on signal processing and integrated networks (SPIN), Noida, India, pp 859–863
15. Mehfuz S, Kumar S (2015) Energy aware probabilistic broadcasting for mobile adhoc network. In: 2015 2nd International conference on computing for sustainable global development (INDIACom), New Delhi, pp 1028–1033
16. Kumar S, Singh AP, Ghosh A (2021) An efficient algorithm for increasing modularity in IoT based automation systems. J Phys Conf Ser 1964:062114
17. Patil RR, Kumar S (2022) Rice-Fusion: a multimodality data fusion framework for rice disease diagnosis. IEEE Access 10:5207–5222

Classification of Tomato Leaf Diseases: A Comparison of Different Optimizers

Arun M. Patokar and Vinaya V. Gohokar

Abstract A plant disease of any kind has a significant impact on the yield and quality of a harvest. The tomato is an important crop with a high commercial value on the global market. Early disease detection is critical for a successful crop yield. Plant disease has recently been the subject of a slew of studies. This research proposes the use of leaf pictures to classify tomato illnesses. Deep learning outperforms machine learning in a number of ways, and this is an important step forward in terms of categorization accuracy and the global reach of applications. ResNet50 and Xception architecture were used in this paper for comparison. Three different optimizers Adam, Nadam, and RMSProp have been used with learning rates of 0.0001, 0.002, and 0.04. All tests were done with data that was available to the public. Rotation, zooming, height shifting, breadth shifting, and other data augmentation techniques are used. Xception Architecture with Adam optimizer and Learning rate of 0.0001 gives greater accuracy, recall, precision, and F-score values of 99% on average when compared to Nadam and RMSProp optimizers and learning rates. These values are higher than those obtained with any other combination of learning rates and optimizers.

Keywords Optimizers · Learning rate · Overfitting · Accuracy · Deep learning

1 Introduction

Agriculture has an important role in the economies of many countries.

There are numerous illnesses that affect a wide range of fruits and vegetables; therefore, plant disease identification is receiving more attention these days. To identify several diseases is difficult task because diseases are changing, to detect them through one disease control method to another is extremely challenging for farmers.

A. M. Patokar (✉)
School of Engineering and Research Technology, STC, Shegaon, Maharashtra, India
e-mail: arunpatokar@gmail.com; arun.patokar@stc.org.in

V. V. Gohokar
School of Electronics and Communication Engineering, MIT World Peace University, Kothrud, Pune, Maharashtra, India
e-mail: Vinaya.gohokar@mitpune.edu.in

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_3

Expert observation with the naked eye is the traditional method for detecting and identifying crop diseases. Farming products must be resistant to illnesses in order to increase their yields. Antagonizing biotic stresses and environmental and meteorological conditions are common challenges for trees and plants presented by Patokar et al. [1]. Pesticides, fertilizers, irrigation policies, and other methods are among the tools farmers employ to address these issues. Agriculture of fruits and vegetables supplies human beings with food and the opportunity to experience a variety of flavors. There are millions of dollars lost each year in apple cultivation because of problems caused by the environment, both natural and man-made. Every stage of the growing season is fraught with pests and pathogens that can harm apple trees. Some of the most common diseases that affect apple leaves include leaf spot, mildew, Alternaria, and leaf spot. In temperate areas, the fungus Venturia causes Apple scab, which is one of the deadliest fungi infections.

Crop diseases diagnosis in large Indian farms is the primary goal of this research, which aims to find best fit optimizer for disease detection that could save crops from crop getting destroyed and even increase yield by protecting crops from different infections. The rapidity, efficiency, reliability, and economy of a Predictive Deep Learning model make it an excellent tool for early diagnosis of plant disease. This study is able to distinguish between healthy and unhealthy plant photos captured by different media or sensors, drones, mobile devices, etc. High-resolution images from [2] are used to train the model, which is then tested on other datasets. Farmers will be able to determine the percentage of their crops affected by illnesses and pests as a result of this research, and they may use our software application's recommendations to limit disease spread and thereby increase agricultural productivity. Algorithms and methods called" optimizer" are used to cut down on an error function (also called a loss function) or make the output process faster. Optimizers, which are mathematical functions, use weights and biases to make them better, and thus, an optimizer can tell you how to modify the weights and learning rate of neural networks. In terms of generating acceptable results in minutes, hours, or days, the optimization approach you select for your deep learning model can make all the difference. The performance Adam optimizer seems to be better at classifying images.

2 Related Work

Deep learning research has advanced over time by presenting novel approaches for achieving astonishing results. Work presented in [3] proposed random search method [4] for tuning the neural network's hyperparameters in order to minimize prediction error. In the same way, several modern studies recommended optimization algorithms for determining the best DL architecture with hyperparameters [5]. Additionally, deep learning necessitates the use of an optimization to continuously upgrade the weight and minimize loss. As a consequence, the research and study community has developed the variety of deep learning optimizers to improve performance on image classification tasks. The work proposed in [6] evaluated a Transfer learning model to detect plant leaf disorder, in which a pre-trained model was used for large datasets. The VGG and Inception models were evaluated using this method. The work presented by Bharali et al. [7] used DCNN to identify and classify leaf diseases. 1400 images were used in the training, and accuracy of 96.6% is achieved. A number of different DL methods are used by Khatoon et al. [8] to identify and predict diseases that are affected by nutritional deficiency, pests, or pathogens.

Zhang and colleagues Zhang et al. [9] proposed a modified Faster R-CNN to detect and classify four diseases, including mosaic virus, leaf mold, powdery mildew, and blight, in tomato crops. The work presented [10] came up with a way to quickly and accurately detect citrus disease from very little data. They used deep metric learning to do this. Also, they came up with a patch-based classifier, which is made up of simple NN classifiers, cluster prototypes, and embedding modules, to help them find accurate citrus disease diagnoses [11]. CNN algorithm was used to find features that match the intensities of image to the correct classes, and then, they compared them to the images that had been trained on a set of other images [12]. This was accomplished by fine-tuning and optimizing various leaf parameters, resulting in a lower classification inaccuracy. The image classification was used to compare images and classify them. Jia Shijie and her colleagues [13] discussed about how to improve the dataset for classification using CNNs and also discussed various ways to improve the dataset, like flip, shift, noise reduction, and GAN. They demonstrate that it was good at predicting the diseases after the application of different augmentations techniques and that worked well. Transfer Learning Convolutional Neural Networks can be trained and validated on a variety of dataset subsets using multiple algorithms proposed by [14]. Overfitting was found to be a problem for AlexNet and GoogleNet in plant disease classification tasks on different sets of data. According to the results presented in [15], the test demonstrated that the B4 and B5 models of the EfficientNet achieved the high accuracy and precision of 99.91% and 99.97%, respectively, When the results were compared to certain other deep learning architectures using the existing and augmented datasets, the accuracy and precision were 98.42 and 99.39%, respectively. Tomato leaf disease classification can be automated using a transfer learning model, as demonstrated in [16]. From the result of comparison between SGD, Adam, and RMSprop, the Adam optimizer found more accurate. The ICRMBO algorithm was used to optimize convolutional models, Vgg16 and Inception V3, in this study [17]. When the ICRMBO algorithm was used to optimize the VGG16 architecture and Inception V3 architecture, the overall test accuracy was 99.98 and 99.94%, respectively.

To classify images from the PlantVillage and Kaggle dataset, the DenseNet-121 architecture has been used [18]. 29 different classes of diseases of 7 different plants have been taken into account in this study. After converting the original image into the HSV color form, they created a masked image and used it for training and classification with 0.002 learning rate and 50 epochs, which resulted in an average accuracy of 94.96%. PlantVillage's maize leaf images were used to test the CNN architectures, which were then cross-validated five times on the training and test sets. Data augmentation in pre-trained models has an effect on correlation between maize leaf classes, according to the findings of this study. Classification of maize leaf disease was found to be 97% accurate across all CNN models tested [19]. To improve the ability to classify plant leaf diseases, four cutting-edge convolutional neural networks were tested: VGG16, InceptionResNetV2, MobileNetV2, and DenseNet201. Thus, DenseNet201 outperformed InceptionResNetV2 with an accuracy of 66.78% for the CNN model's baseline CNN model, which is a significant improvement over the previous model [20].

In this research [21], researchers compare and contrast several different methodologies and find that deep learning networks (VGG16, ResNet34, AlexNet, MobileNetV2, and EfficientNet) [22] outperform machine learning algorithms (SVM, kNN, and random forest). Furthermore, in the classification task, the ResNet34 network achieved the best results [23], with an accuracy of 99.7%, precision of 99.6%, recall of 99.7%, and F1 score of 99.7%. CNNs were used in this paper [12] to make the leaf spot feature extraction and classifier, and use of two-way parallel CNNs to make these things easier. VGG, DenseNet, and ResNet were chosen for comparison, and local optimization has been done with spatial structure optimizer [24]. On average, the classification accuracy of S-DenseNet is 7.7% better than DenseNet. This is true for different number of steps. Thus, by using SSO, each of these neural networks can perform better.

The comparison and impact of VGG19 and Mobile Net were discussed in this paper [25]. A CNN model, VGG19, and MobileNet are used to train a potato and a grape. Each of these models has an accuracy rate as: 93.78% for a potato and grape leaf 93.90% for CNN model, 77.33% for potato and 93.90 for grape using VGG19, and in MobileNet 96.30%, and 80.00%, respectively, for potato and grape. Using performance metrics like validation accuracy, F_1 score, and epochs number, Saleem et al. [4] compared various CNNs. The dataset contains a total of 26 plant diseases, each of which is associated with one of 14 different species. The Xception architecture is shown to outperform when trained using the Adam optimizer.

3 Methodology

The effectiveness of deep learning models with different optimizers and learning rates on Images of tomato leaves has been validated in this paper. The dataset [2, 26] and was used. This study employed a total of 19,372 single leaf photographs. The class distribution such as denoted by numbers from 0 to 10 is shown in the image class distribution Table 1.

All of the data are obtained from a publicly accessible repository [2, 26]. In future studies, the quantity of photos obtained may be different because datasets are always being updated. Pretrained models use various input sizes. It is necessary for Xception model to have an input shape (299, 299, 3). Three inputs are required, and the width and height must be never less than 71 pixels in each direction, respectively. ResNet50 requires a specific input shape (224, 224, 3). Input channels should be limited to three, and the device's width and height should never be less than 32. In this experiment,

Table 1 Image class distribution for tomato leaf images

Image class	Leaf disease	Number of Images
0	Bacterial spot	1702
1	Early blight	1920
2	Healthy	1926
3	Late blight	1851
4	Leaf mold	1882
5	Septoria leaf spot	1745
6	Spider mites	1741
7	Target spot	1827
8	mosaic virus	1790
9	Yellow leaf curl virus	1961
10	Background	1027

the input photos were downsized to 128 × 128 × 3 pixels. All images were scaled to meet the pre-trained model's training requirements.

In the proposed technique, there are three steps. The first steps include data collection from an open-and data augmentation. It is not possible to train a model using the dataset as it acquired, since each image had a different amount of channels. As a starting point, we had converted all images to 3 channels. Data augmentation techniques, such as rotating, width shift, height shift, zooming, and image flip, have been employed to ensure the best possible results. Various training parameters, such epochs, model selection, optimizer, epoch, learning rate, and number of folds per epoch, and batch size, were determined during the second stage. In order to accommodate both models, data was resized to a 128 × 128 × 3 shape.

The classification is carried out in the third step, during which the network determines whether a diseased leaf or a healthy leaf or background has been detected in the scans. A large number of test runs have been carried out to ensure that the trained models were valid, due to the fact that disease detection is a highly sensitive case. Both models were trained with different optimizers with learning rate (Lr) 0.0001, 0.0002, and 0.0004. Table 2 presents architecture, optimizers, and learning rates used in this study, to find the superlative combination and minimize the factor of over fitting in the data training process. Xception and Resnet50 pre-trained models were used in this research, and the three-step implementation flow is represented in Fig. 1. The PlantVillage and Kaggle dataset has been combined and used to train the deep learning models from scratch. Summary of the technical details for all of the DL architecture optimizers with different learning rates is mentioned in Table 2.

Early detection and correct identification of plant leaf disease are critical to halting the disease's spreading and ensuring the health of the crops. This method's output must be checked for accuracy. Using a variety of means. Model performance is evaluated using metrics like as Accuracy, Precision, Recall, and F_1 Scores. To train our deep learning models, Jupyter notebook has been used for programming. The

Table 2 Xception/Resnet50 architecture with 3 optimizers and 3 learning rate

S. No.	Architecture	Optimizer	Lr (learning rate)
1	Xception/ResNet50	Adam	0.0001
2	Xception/ResNet50	Adam	0.0002
3	Xception/ResNet50	Adam	0.0004
4	Xception/ResNet50	Nadam	0.0001
5	Xception/ResNet50	Nadam	0.0002
6	Xception/ResNet50	Nadam	0.0004
7	Xception/ResNet50	RMSProp	0.0001
8	Xception/ResNet50	RMSProp	0.0002
9	Xception/ResNet50	RMSProp	0.0004

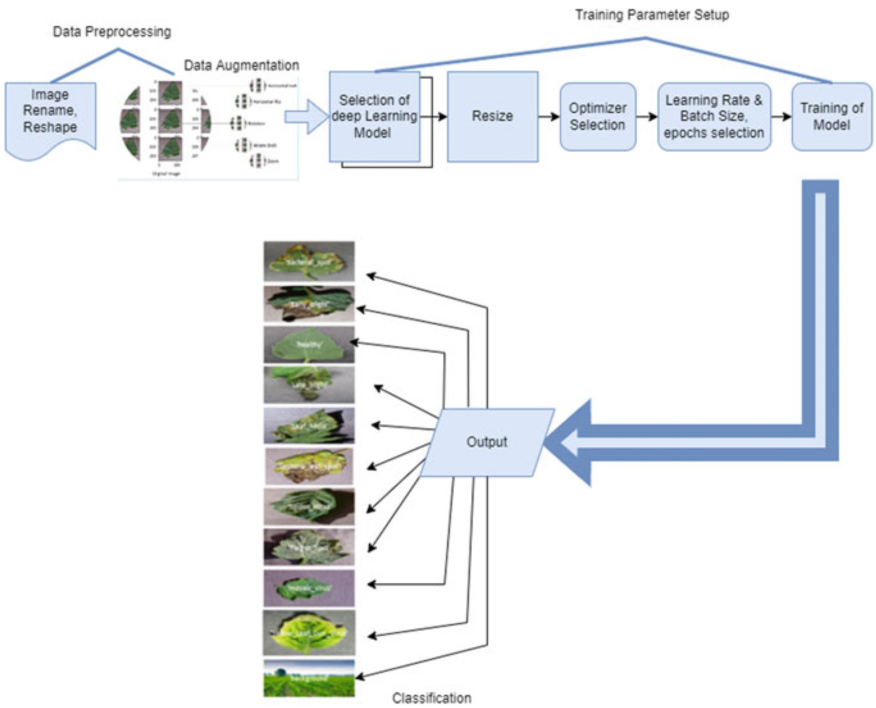


Fig. 1 Implementation flow

NVIDIA GeForce GT 720 GPU and an Intel Core i5-6500 CPU were used for all training and prediction. Base formulas for performance evaluation of accuracy, Precision, Recall Protik et al. [27], and F_1-score are used.

4 Results and Discussion

This study is designed to detect and classify 11 different classes of tomato leaf images. Multiple optimizers and a varying learning rate have been applied during training the two deep learning models. This research aims to find the best model, optimizer, and learning rate combination for the input data.

The model Xception with 3 different optimizers and learning rates with accuracy is presented in Fig. 2. For Adam with learning rate(Lr) 0.001, Nadam and RMSProp with Lr 0.0002 give accuracy of 99%, and for rest combination of Learning rate, it provides accuracy of 98%. The "Adam" optimizer's top outcomes were obtained with a learning rate of 0.0001, "Nadam" optimizer's top outcomes were obtained with a learning rate of 0.0002, for RMSProp with Lr 0.0002. The model ResNet50 with 3 different optimizers and learning rates with accuracy is presented in Fig. 3. For Adam with Lr 0.002, Nadam with Lr 0.0002 and RMSProp with Lr 0.0001 and Lr 0.0004 give accuracy of 98%, 97%, and 96%, respectively. For rest combination of learning rate, it provides accuracy of 79%; for "Adam" optimizer's and for Nadam and RMSProp, it provides 89% and 88%, respectively.

After the training, it is observed that the learning rate is a hyperparameter which governs how much we adjust our network's weights in relation to the loss gradients. The lesser the value, the slower we move down the hill. While adopting a low learning rate may be a good option in terms of ensuring that we do not miss any premature convergence, it may also mean that convergence will take a lengthy time. Thus, we can say Adam and Nadam optimizers perform well with minimum learning rate. The Xception model generated after training is light weight and can be easily ported to hardware like Raspberry-Pi for making real-time leaf disease identification and classification system.

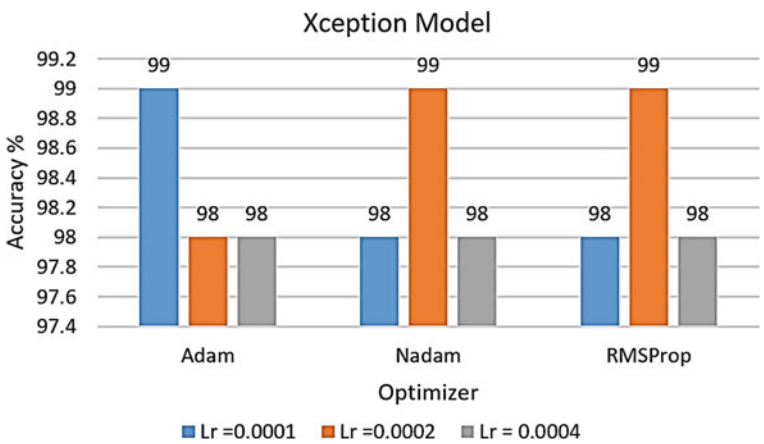


Fig. 2 Xception model with different optimizers and learning rate

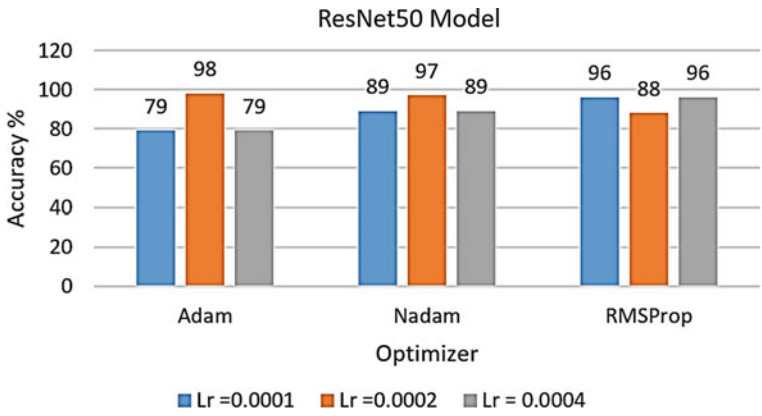


Fig. 3 ResNet50 model with different optimizers and learning rate

After training the Xception model with the Adam optimizer and learning rate 0.0001, prediction was done on random 20 images and prediction is presented as shown in Fig. 4. Predicted images present 8 different classes of leaf infections, one healthy class and one background class image.

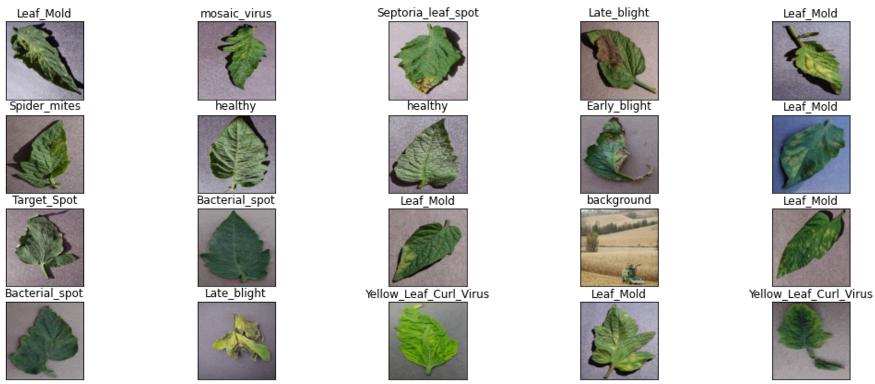


Fig. 4 Predicted 20 images with Xception model, adam optimizer, and Lr = 0.0001

5 Conclusion

Diseased leaves are one of the causes in yield reduction, which causes reduction in chlorophyll content [28]. The decline in photosynthesis appears to be caused by the build-up of starch in diseased leaves. The trained model for disease detection and classification can be used for automatic leaf disease detection and classification in early stages of leaf disease. By comparing the performance of three optimizers with two different architecture Adam optimizer work well with minimum learning rate 0.0001 with Xception model without over fitting. Model can be simply translated to low-cost hardware to identify leaf disease automatically. In future, IoT-enabled autonomous robotic platform can be deployed with this system so that farmers can take monitor remotely. Also, necessary preventive action can be taken to reduce yield loss due to disease. We believe that this study will be of great assistance in the early recognition of leaf diseases, thereby preventing significant crop yield loss.

References

1. Patokar AM, Gohokar VV (2020) Plant leaf disease identification and classification using transfer learning | solid state technology. http://solidstatetechnology.us/index.php/JSST/article/view/2087
2. Huang M, Chang Y (n.d.) Dataset of Tomato leaves—mendeley data. Mendeley Data. https://data.mendeley.com/datasets/ngdgg79rzb/1
3. Torres JF, Gutiérrez-Avilés D, Lora AT, Martínez-Álvarez F (2019) Random hyper-parameter search-based deep neural network for power consumption forecasting. IWANN
4. Saleem MH, Potgieter J, Arif KM (2020) Plant disease classification: a comparative evaluation of convolutional neural networks and deep learning optimizers. Plants 9(10):1319. https://doi.org/10.3390/plants9101319
5. Martínez-Álvarez F, Asencio-Cortés G, Torres JF, Gutiérrez-Avilés D, Melgar-García L, Pérez-Chacón R, Rubio-Escudero C, Raquel J, Troncoso A (2020) Coronavirus optimization algorithm: a bioinspired metaheuristic based on the COVID-19 propagation model. Big Data 8:308–322
6. Chen J, Chen D, Zhang Y, Sun YA (2020) Nanehkaran, Using deep transfer learning for image-based plant disease identification. Comput Electron Agric 173:105393. https://doi.org/10.1016/j.compag.2020.105393
7. Bharali P, Bhuyan C, Boruah A (2019) Plant disease detection by leaf image classification using convolutional neural network. Springer, Singapore, pp 194–205
8. Khatoon S, Hasan MM, Asif A, Alshmari M, Yap YK (2021) Image-Based automatic diagnostic system for tomato plants using deep learning, CMC 67(1):595–612, https://doi.org/10.32604/cmc.2021.014580
9. Zhang Y, Song C, Zhang D (2020) Deep learning-based object detection improvement for tomato disease. IEEE Access 8:56607–56614. https://doi.org/10.1109/ACCESS.2020.2982456
10. Janarthan S, Thuseethan S, Rajasegarar S, Lyu Q, Zheng Y, Yearwood J (2020) Deep metric learning based citrus disease classification with sparse data. IEEE Access 8:162588–162600. https://doi.org/10.1109/ACCESS.2020.3021487

11. Ashwinkumar S, Rajagopal S, Manimaran V, Jegajothi B (2021) Automated plant leaf disease detection and classification using optimal MobileNet based convolutional neural networks. ISSN 2214–7853. https://doi.org/10.1016/j.matpr.2021.05.584
12. Liang X, Liu W, Zhang Y, Yu J, Qu H (2018) Dimensionality- Varied convolutional neural network for hyperspectral image classification with small-sized labeled samples. In: 2018 9th workshop on hyperspectral image and signal processing: evolution in remote sensing (WHISPERS), Amsterdam, Netherlands, pp 1–5.https://doi.org/10.1109/WHISPERS.2018.8747243
13. Shijie J, Ping W, Peiyi J, Siping H (2017) Research on data augmentation for image classification based on convolution neural networks. In: 2017 Chinese automation congress (CAC), Jinan, pp 4165–4170. https://doi.org/10.1109/CAC.2017.8243510
14. Bhimavarapu S, Vinitha PJ (2020) Analysis and characterization of plant diseases using transfer learning. In: 2020 International conference on communication and signal processing (ICCSP), Chennai, India, pp 1293–1296. https://doi.org/10.1109/ICCSP48568.2020.9182451
15. Atila Ü, Uçar M, Akyol K, Uçar E (2021) Plant leaf disease classification using EfficientNet deep learning model. Ecol Inform 61. https://doi.org/10.1016/j.ecoinf.2020.101182
16. Thangaraj R, Anandamurugan S, Kaliappan VK (2021) Automated tomato leaf disease classification using transfer learning-based deep convolution neural network. J Plant Dis Prot 128(1):73–86. https://doi.org/10.1007/s41348-020-00403-0
17. Nandhini S, Ashokkumar K (2021) Improved crossover based monarch butterfly optimization for tomato leaf disease classification using convolutional neural network. Multimedia Tools Appl 80(12):18583–18610. https://doi.org/10.1007/s11042-021-10599-4
18. Vellaichamy AS, Swaminathan A, Varun C, Kalaivani S (2021) Multiple plant leaf disease classification using densenet-121 architecture. Int J Electr Eng Technol 12(5). https://doi.org/10.34218/ijeet.12.5.2021.005
19. Da Rocha EL, Rodrigues L, Mari JF (2021) Maize leaf disease classification using convolutional neural networks and hyperparameter optimization. Sociedade Brasileira de Computacao, SB. pp 104–110. https://doi.org/10.5753/wvc.2020.13489
20. Puangsuwan T, Surinta O (2021) Enhancement of plant leaf disease classification based on snapshot ensemble convolutional neural network. ICIC Exp Lett 15(6):669–680. https://doi.org/10.24507/icicel.15.06.669
21. PB G, Prabhakar V, GS A (2021) An evaluation of multiclass leaf classification using transfer learning techniques | SpringerLink. https://link.springer.com/chapter, https://doi.org/10.1007/978-3-030-84760-9_3
22. Alsabei A, Alsayed A, Arif M (2021) Classification of apple tree leaves diseases using deep learning methods. Int J Comput Sci Netw Secur. https://www.koreascience.or.kr/artcle/JAKO202123563872609.jsp1ff8%3Fkj=OJOOBS&py=2014&vnc=v31n1ff1ff882&sp=141 .
23. Tan L, Lu J, Jiang H (2021) Tomato leaf diseases classification based on leaf images: a comparison between classical machine learning and deep learning methods. AgriEngineering 3(3):542–558. https://doi.org/10.3390/agriengineering3030035
24. Liang X (2021) Few-shot cotton leaf spots disease classification based on metric learning. Plant Methods 17(1). https://doi.org/10.1186/s13007-021-00813-7
25. Wongbongkotpaisan J, Phumeechanya S (2021) Plant leaf disease classification using local-based image augmentation and convolutional neural network. In: ECTI-CON 2021—2021 18th international conference on electrical engineering/electronics, computer, telecommunications and information technology: smart electrical system and technology, proceedings. Institute of Electrical and Electronics Engineers Inc., pp 1023–1027 https://doi.org/10.1109/ECTI-CON51831.2021.9454672

26. PlantVillage Dataset (n.d.) PlantVillage Dataset | Kaggle (2018). https://www.kaggle.com/emmarex/plantdisease
27. Protik A, Rafi AH, Siddique S (2021) Real-Time personal protective equipment (PPE) detection using YOLOv4 and TensorFlow. In: 2021 IEEE Region 10 symposium (TENSYMP), pp 1–6. https://doi.org/10.1109/TENSYMP52854.2021.9550808
28. Mandal K, Saravanan R, Maiti S et al (2009) Effect of downy mildew disease on photosynthesis and chlorophyll fluorescence in Plantago ovata Forsk. J Plant Dis Prot 116:164–168. https://doi.org/10.1007/BF03356305

Optimization of Rainfall Intensities Classification Based on Artificial Intelligence Using Recurrent Neural Network

Mourad Lazri, Karim Labadi, Fethi Ouallouche, and Soltane Ameur

Abstract In this paper, a classification of precipitation intensities is performed using the recurrent neural network (RNN). The latter is used to take into account temporal information in order to verify the contribution of the previous states to future states in the prediction of precipitation classes. The structure of an RNN introduces a mechanism of memory of the previous entries which persists in the internal states of the network and can thus impact all its future exits. The RNN is learned by using the mappings between Meteosat Second Generation (MSG) data as network inputs and radar data as network outputs. With the RNN, the outputs are also recombined at the inputs. To classify a precipitation scene at time t, the results depend on the MSG inputs at time t and the classification results at time $t - 1$. The model was evaluated by making comparisons with radar data considered as reference data. To see the contribution of RNN, we have also implemented the ANN, and a comparison between RNN results and ANN results is performed. Thus, all the comparisons show very interesting performances in terms of good classification rate obtained by RNN.

Keywords Artificial intelligence · RNN · Classification · Satellite MSG · MLP

1 Introduction

Precipitation is considered a crucial parameter because of its importance for the ecological balance of the planet and for the functioning of the climate system. In the recent years, due to the growing need for water resources, precipitation has even become a worrying factor because of its plummet and decline for some regions of the world. It is therefore very urgent to define a management plan to ration water

M. Lazri (✉) · F. Ouallouche · S. Ameur
Laboratoire LAMPA (Laboratoire d'Analyse et de Modélisation des Phénomènes Aléatoires), Faculty G.E.I, University Mouloud MAMMERI, Tizi-Ouzou, Algeria
e-mail: m_lazri@yahoo.fr

K. Labadi
Laboratoire LR2E, ECAM-EPMI/Quartz-Lab, Cergy-Pontoise, France
e-mail: k.labadi@ecam-epmi.com

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_4

resources. The management plan must necessarily take into account the spatial–temporal distribution maps of precipitation. However, measuring precipitation is one of the most complexes because of its randomness in time and space.

Traditional means for measuring precipitation such as radars and rain gauges are no longer sufficient; areas are completely inaccessible. To circumvent these deficiencies, data from meteorological satellites are used extensively for the classification and estimation of precipitation at large spatio-temporal scales and at all points of the globe. However, the indirect nature between satellite information and precipitation requires the implementation of techniques that make it possible to relate the optical and microphysical properties of clouds to precipitation rates [1–4].

Recently, artificial intelligence has made a very remarkable breakthrough in this context. Indeed, techniques based on machine learning are widely used in meteorology and particularly in the classification of precipitation intensities [5, 6]. Among these techniques, we can cite artificial neural networks which are one of the models that have given satisfaction in image processing and particularly in automatic classification [3]. Thus, a large number of techniques using neural networks, in particular multilayer perceptron (MPL) are implemented for the classification of precipitation intensities from meteorological satellite images [7] without however testing other neural network architectures. However, neural networks are well documented in the literature and several architectures have been developed. Architectures have their strengths and weaknesses and each can be adapted to a type of processing. The choice of architecture is thus crucial and depends mainly on the objective.

Precipitation events, despite their randomness, persist in one state before changing state. This characteristic was never taken into account during the classification. Indeed, two clouds can have the same characteristics without belonging to the same class. For example, one cloud is growing and another is decaying. The separation between the two clouds can be done by introducing the previous state.

In this context, our contribution in this paper is to integrate the sequential notion on precipitation events using recurrent neural networks (RNNs). These use the previous outputs as additional inputs to the model and are suitable for processing sequential data. Unlike a classic artificial neural network (ANN) where the output depends only on the input values, the calculation at time t is based on the information provided from time $t - 1$, itself calculated from the information provided from $t - 2$, etc. The methodology therefore consists in classifying the precipitation intensities from the MSG images by introducing the notion of time. RNN training is performed by comparing the MSG data to the corresponding data from the radar.

The rest of the paper is organized as follows. The methodology is discussed in Sect. 2. The results and interpretation will be presented in Sect. 3. Section 4 contains the conclusion and perspectives.

1.1 Data and Methodology

To classify precipitation intensities, MSG images and radar image correspondences are used. The model used for classification is recurrent neural networks (RNNs). To see the contribution of the RNN model to the classification, we compared the classification results obtained by RNN to the classification results of MLP. We present the steps of our procedure; MSG data and radar data, selection of input parameters, RNN model, choice of RNN architecture.

1.2 MSG Data and Radar Data

The study area is located in the northeast of Algeria (Fig. 1).

These are therefore pairs of images (MSG image/radar image) used initially for the training of RNN, that is, the images collected during the period from October 2009 to April 2010. In a secondly, other pairs are used for the evaluation of the classification by RNN. They are recorded during the period from October 2010 to April 2011.

MSG satellites provide multispectral images (12 channels) at temporal and spatial resolutions of 15 min and 3 * 3 km^2 (VSH at 1 * 1 km^2), respectively. A pixel is coded on 10 bits giving numerical values between 0 and 1023 which will be converted into radiance then into brightness temperature for the infrared channels and into reflectance for the visible channels [8]. MSG images taken in infrared and visible channels, such as IR10.8, IR8.7, IR3.9, IR12.1, WV6.2, WV7.3, VIS0.6 and VIS1.6 are selected. The spatial resolution reaches 4 * 5 km^2 in the study region.

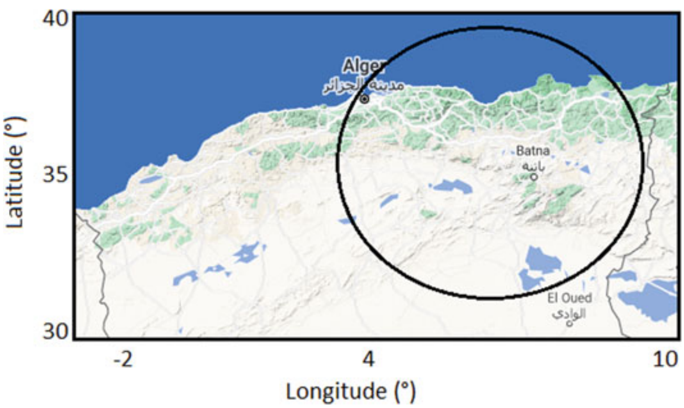


Fig. 1 Study area indicated bay the radar coverage of Sétif (northeast of Algeria). *Source* google maps

Table 1 Input parameters, possible values and the corresponding information

Input parameters	Possible values		Information on
	Daytime	Nighttime	
$T_{10.8}$ (K)	207.2 k to 283.9 k	205.3 k to 282.4 k	CTT and CTH
$\Delta T_{10.8–12.1}$ (K)	− 0.3 k to 7.4 k	− 0.3 k to 7.1 k	CP
$\Delta T_{8.7–10.8}$ (K)	− 4.6 k to 1.3 k	− 4.8 k to 1.7 k	CP
$\Delta T_{6.2–10.8}$ (K)	50.1 k to 6.4 k	− 51.8 k to 5.1 k	CTT and CTH
$\Delta T_{7.3–12.1}$	− 50.3 k to 6.6 k	− 52.0 k to 5.7 k	CTT and CTH
$R_{0.6}$ (μm)	0.02 μm to 1 μm	No used	CWP
$R_{1.6}$ (μm)	0.03 μm to 1 μm	No used	CWP
$\Delta T_{3.9–10.8}$ (Kelvin)	No used	− 10.3 k to 15.1 k	CWP
$\Delta T_{3.9–7.3}$ (Kelvin)	No used	− 4.9 k to 25 k	CWP

The correspondences in radar images are taken by the Sétif radar every 15 min with a spatial resolution of 1 * 1 km^2. A pixel has 16 classes (dBZ).

In order to bring the MSG image and radar image correspondence as closely as possible, the radar images were brought to the same spatial resolution of the MSG images by replacing the windows of 4 * 5 pixels by the mean rounded to the nearest class over the 16 classes. Temporarily, the two sources do not require processing because the time shift is very small.

1.3 Selection of Input Parameters

As stated earlier, a set of channels are used. They are used to provide information on the temperature of the cloud tops (CTT), the cloud vertical extension (CTH), the thermodynamic phase of clouds (CP) and the cloud water path (CWP) (Table 1).

1.4 RNN Model

In the recent years, a significant number of RNN architectures have been developed. Here, we are interested in the simpler version of RNNs [9]. This version consists of adding links to an MLP of output layers to the previous layers. The previous outputs are therefore combined with current input parameters to give the current outputs.

To describe the RNN, we will first discuss some parameters about the MLP. An MLP is defined by the n layers which compose it and which follow one another. The layer $j \in [1, N]$ of a MLP is given by a triplet: $Lj = (nj, \sigma j, aj)$ where:

- $n_j \in N$ represents the number of neurons in layer j,
- a_j: R^{nj-1} to R^{nj}: is the affine transformation given by the matrix $W_j \in R^{\mathrm{nj}-1 \times nj}$ and the vector $bj \in R^{nj}$.
- σj: R^{nj} to R^{nj}: is the transfer function of layer j.
- The matrix W_j is called the weight matrix or interconnection matrix between layer $j - 1$ and layer j.
- The vector b_j represents the bias vector of layer j.

The function σ_j is chosen among the functions from R to R, bounded, nonlinear and derivable such as the sigmoid or the hyperbolic tangent.

The transfer function is applied to each element of the vectors of dimension n_j.

To develop such an RNN, a weight matrix (interconnection) $V_j \in R^{nj \times nj}$ is added to the triplet of a layer $L_j = (n_j, \sigma_j, a_j)$ in the case of a MLP to give a layer L'_j of the recurring network. In the case of a series of input vectors $x(t)$ with $t \in [1, t_f]$, we obtain a series of output vectors $z(t)$ with $t \in [1, t_f]$ whose internal state vectors are initialized with $h_0(t) = x(t)$, $\forall t \in [1, t_f]$ and $h_j(0) = 0$, $\forall j \in [1, N]$.

All the layers are applied recursively for each time step $L'j$, $j \in [1, N]$ from the network to the output vector of the previous layer as follows:

The affine transformation a_j is applied to the output vector of the previous layer, to which we add the linear transformation given by V_j and applied to its output vector at the previous time step:

$$g_j(t) = W_j * h_{j-1}(t) + V_j * h_j(t-1) + b_j \tag{1}$$

Then, we apply the transfer function to the result:

$$h_i(t) = \sigma_j\left(g_j(t)\right) \tag{2}$$

$$\text{With } \forall t \in \left[1, t_f\right] \text{et} \, \forall j \in [1, N], h_j(t), g_j(t) \in R^{n_j}, z(t) = h_N(t) \tag{3}$$

In the case of back-propagation of errors, the reader can refer to [10].

1.5 Choice of RNN Architecture

In our case, this is a modification of MLP. This simple change in the MLP architecture has important consequences for the model, and as we will see, for the results. Indeed, the outputs at time t will be combined with the inputs at time $t + 1$ to generate outputs at time $t + 1$ (Fig. 2).

The two models are therefore learned on a database made up of data pairs (MSG data (inputs)/radar data (outputs)). The database is collected during the rainy period from October 2009 to April 2010. Once the training is finished, the coefficients obtained are used for the classification of the precipitation scenes observed from October 2010 to April 2011.

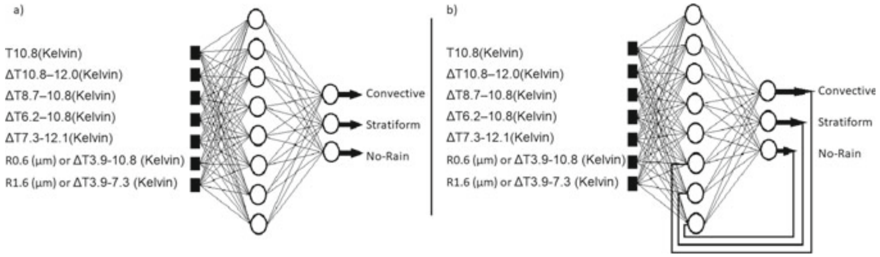


Fig. 2 Artificial neural networks, **a** MLP with three layers (input layer, hidden layer, output layer, **b** modified MLP where the outputs of the output layer are connected to a few neurons of the hidden layer to give an RNN

2 Application and Classification Results

We therefore applied the RNN model for the classification of MSG images after the training step. To show the contribution of the RNN method, the MLP model has also been implemented. We applied the two models first on an instantaneous scene and secondly on all the scenes of the rainy season.

2.1 Application to a Scene

We tested both RNN and MLP models on an instantaneous high pressure precipitation scene taken on December 29, 2010. Large convective cells are embedded in a stratiform system. The classification results are shown in Fig. 3.

Visually, the number of misclassified pixels is greater in the case of the MLP classification than in the case of RNN. For this scene, we also determined the number of well-classified, misclassified and very badly classified pixels, allowing a better understanding of the difference between MLP results and RNN results (see Table 2).

Table 2 clearly indicates that the RNN better identified the pixel classes for both the convective case and the stratiform case. Misclassified and highly misclassified pixels are more present for MLP. Between the misclassified and very misclassified pixels, in the case of the convective class, the RNN misidentified 177 pixels against 256 for the MLP. For the stratiform class, the RNN missed 191 pixels against 282 for the MLP.

The good classification rates are estimated at 95.9% and 93.7% for the RNN and the MLP, respectively.

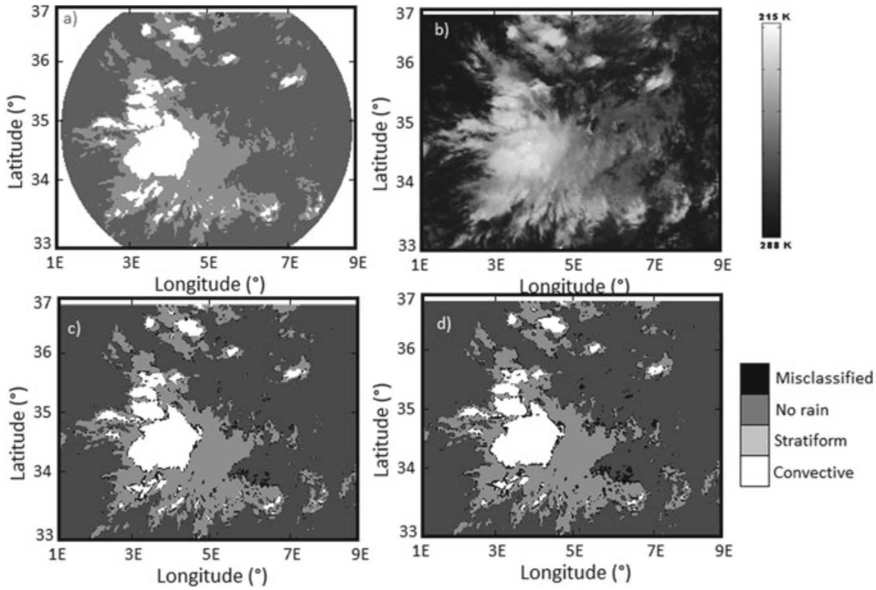


Fig. 3 Classification results, **a** preprocessed radar image for comparison, **b** MSG IR10.8 image, **c** classification using RNN, **d** classification using MLP

Table 2 Number of well-classified, badly classified and very badly classified pixels

Class types	Pixels classified by radar	Classified by RNN			Classified by MLP		
		Convective	Stratiform	No rain	Convective	Stratiform	No rain
Convective	1645	1468	135	42	1389	194	62
Stratiform	3421	83	3230	108	131	3139	151
No rain	5523	11	50	5462	45	79	5399

2.2 Application to All Scenes

The RNN and MLP models were tested on all the scenes of the rainy season. To better highlight the difference and the contribution of the RNN model, we calculated evaluation parameters using the comparisons between the classification results and the correspondences with the radar data. For the two models, Table 3 gives the correspondence coefficients between the classifications and the radar data.

These evaluation parameters are POD, POFD, FAR, Bias, CSI and PC calculated using (Eqs. 4–9). The statistical results are given in Table 4

$$\mathrm{POD} = \frac{a}{a + c} \tag{4}$$

Table 3 Overview of the combinations in a contingency table

Identified by satellite method		Observed by radar		
		Class (i)	No class (i)	Total
	Class (i)	a	b	$a+b$
	No class (i)	c	d	$c+d$
	Total	$a+c$	$b+d$	$a+b+c+d=n$

$$\text{POFD} = \frac{b}{b+d} \tag{5}$$

$$\text{FAR} = \frac{b}{a+b} \tag{6}$$

$$\text{Bias} = \frac{a+b}{a+c} \tag{7}$$

$$\text{CSI} = \frac{a}{a+b+c} \tag{8}$$

$$\text{PC} = \frac{a+d}{n} \tag{9}$$

Table 4 gives the different evaluation parameters obtained when comparing the classifications and the reference radar data. As with the instantaneous scene, the RNN indicates the best classification results.

The two models show the same tendency in terms of bias. An overestimation for the convective class and an underestimation for the stratiform class is observed but with the best values for RNN.

The values obtained POD, POFD and FAR confirm the performance of the RNN model even if with slight improvements for both the convective class and the stratiform class. In the same optic, we also noted superior performances for RNN in terms of CSI and PC for the two classes, convective and stratiform.

All these results clearly show that the classification of precipitation intensities can be optimized by incorporating temporal information concerning the evolution of precipitation systems.

3 Conclusion

The objective of this article is to show the contribution of recurrent neural networks in the classification of precipitation intensities. Despite its randomness, precipitation often persists in one state before changing states, and this information is incorporated through the use of RNN.

Table 4 Values obtained for the evaluation parameters for MLP and RNN

	Convective precipitation						Stratiform precipitation					
	POD (%)	POFD (%)	FAR (%)	Bias	CSI (%)	PC (%)	POD (%)	POFD (%)	FAR (%)	Bias	CSI (%)	PC (%)
MLP	91	09	27	1.1	82	92	84	16	30	0.86	78	90
RNN	93	08	22	1.08	87	95	91	12	28	0.89	84	92
Optimal values	100	0	0	1	100	100	100	0	0	1	100	100

The RNN used is developed by modifying the architecture of MLP consisting of an input layer, a hidden layer and an output layer. For the RNN, the three outputs are connected to a few neurons in the hidden layer. To see the impact of this MLP modification (RNN), we implemented both models.

First, we applied the two models to an instantaneous scene consisting of stratiform precipitation with embedded convective precipitation cells. The results indicate that the RNN model made it possible to better identify the different classes. Indeed, the rate of good classification for the RNN is 95.9% against 93.7% for MLP.

Secondly, the scenes during the rainy season between October 2010 to November 2011 were classified. As for the instantaneous scene, the performances of RNN were highlighted. Compared to radar data, different evaluation parameters are calculated. All indicate that the best values were obtained for the case of RNN.

After showing the importance of incorporating information on the temporal behavior of precipitation systems, it would now be interesting to find a better architecture of a recurrent neural network for classification.

References

1. Huffman GJ, Adler RF, Morrissey MM, Bolvin DT, Curtis S, Joyce R, Mcgavock B, Susskind J (2001) Global precipitation at one-degree daily resolution from multisatellite observations. J Hydrometeorol 2:36–50
2. Levizzani V (2003) Satellite rainfall estimations: new perspectives for meteorology and climate from the EURAINSAT project. Ann Geophys 46:363–372
3. Oukali S, Lazri M, Labadi K, Brucker JM, Ameur S (2019) Development of a hybrid classification technique based on deep learning applied to MSG/SEVIRI multispectral data. J Atmos Solar Terr Phys 193:105062
4. Thies B, Nauss T, Bendix J (2008) Precipitation process and rainfall intensity differentiation using Meteosat second generation spinning enhanced visible and infrared imager data. J Geophys Res 113
5. Kuhnlein M, Appelhans T, Thies B, Nauss T (2014) Improving the accuracy of rainfall rates from optical satellite sensors with machine learning—a random forests-based approach applied to MSG SEVIRI. Remote Sens Environ 141:129–143
6. Lazri M, Labadi K, Brucker JM, Ameur S (2020) Improving satellite rainfall estimation from MSG data in Northern Algeria by using a multi-classifier model based on machine learning. J Hydrol 584:124705
7. Lazri M, Ameur S (2018) Combination of support vector machine, artificial neural network and random forest for improving the classification of convective and stratiformrain using spectral features of SEVIRI data. Atmos Res 203:118–129
8. Eumetsat (2004) Applications of meteosat second generation—conversion from counts to radiances and from radiances to brightness temperatures and reflectance, http://oiswww.eumetsat.org/WEBOPS/msg_interpretation/index.html
9. Elman JL (1990) Finding structure in time. Cognitive Sci 14;16(2):179–211
10. Werbos PJ (1990) Backpropagation through time: what it does and how to do it. Proc IEEE 78;19(10):1550–1560

Tomato Plant Leaf Disease Detection Using Inception V3

Harsh Baheti, Anuradha Thakare, Yash Bhople, Sudarshan Darekar, and Om Dodmani

Abstract Plant disease detection is beneficial because it detects disease symptoms early on, such as when diseases appear on plant leaves. First of all, the system takes photos of plant leaves, which are then processed and matched to photographs of diseased plant leaves in the database. And then, it will show result, if any disease is their then it will notify us. After that, farmer can take action to solve problem very easily. In this paper, we take tomato as our sample plant. Tomato plant leaves are taken to identify its diseases. The disease is detected using the Inception V3 model (Convolutional Neural Network). In terms of testing and validation accuracy, the intermediate results are presented. It is observed that the best training accuracy, 88.98% is obtained at the 10th epoch and the best validation accuracy, 85.80% is obtained at the 8th epoch. Lastly, we conclude our result.

Keywords Tomato disease · Bacterial spot · TensorFlow · Keras · Inception V3

1 Introduction

The agriculture sector provides the most amount of food for people in a country. Their productivity is increased by using various methods such as fertilizers. But due to the excess use of these methods now we can see that productivity has declined. Pests and climate change have also drastically affected crop yield and which resulted

H. Baheti (✉) · A. Thakare · Y. Bhople · S. Darekar · O. Dodmani
Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, India
e-mail: harsh.baheti19@pccoepune.org

A. Thakare
e-mail: anuradha.thakare@pccoepune.org

Y. Bhople
e-mail: yash.bhople19@pccoepune.org

S. Darekar
e-mail: sudarshan.darekar19@pccoepune.org

O. Dodmani
e-mail: om.dodmani19@pccoepune.org

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_5

in a huge effect on the economy. As a result, it is necessary to monitor and properly care for our agricultural commodities to boost our annual product yield. To ensure the crop is healthy manual inspection is done by most of the farmers about the disease on the crop but this is a hectic process. By using enhanced computer technology, we can minimize the effects of diseases in plants and also maximize agricultural productivity. The goal of this project is to use machine learning to design and implement real-time decision support hardware for identifying healthy and unhealthy leaves.

Symptoms, spots, color, and other characteristics of various diseases that affect different regions of the plant can be identified. To increase crop production rates, the agriculture sector needs a less time-consuming, efficient, and autonomous diagnosing methodology. Development in computer technology has helped in minimizing the diseases in crops.

Machine learning techniques combined with image processing are assisting in the early detection of diseases to reduce the impact. The texture and color of the leaf are collected using external hardware, and processing is done applying image processing techniques such as segmentation, grayscale conversion, and then applying algorithms to the datasets available for a plant, in this case, tomato. We can get the result about the type of disease the plant is having. This report shows the detection of ten types of diseases in tomato crops with a variety of about forty thousand images of the diseased leaf available in the dataset. The more the images in the dataset, the more the accuracy of the program. Thus, we can help in increasing crop yield by using advanced computer technology.

Plant disease is the leading cause of crop volume and quality reduction in agriculture; plant disease detection is critical for preventing agricultural losses. Agricultural productivity is extremely important to the economy. If we do not focus on this area, it can have major consequences for plants, resulting in worse product quality, quantity, or productivity.

This research is divided into four sections: Sect. 2 discusses the existing research; Sect. 3 represents the workflow for tomato plant disease detection with detailed discussion on dataset selection and intermediate results. Section 4 presents the conclusion of the model.

2 Related Research

The paper includes Tomato Disease Classification using AlexNet. [1]. Two types of experimental setups are available: 10-class classifier = healthy (1 class) + diseased (9 classes) and 2-class classifiers = healthy (1 class) + sick (1 class). Three training scenarios are used to test each instance. Finally, we highlight a specific feature within the PlantVillage dataset. Using a Deep Neural Network, Improved Vision-Based Detection of Strawberry Diseases [2]. A two-stage cascade disease detection model was deployed using the backbone feature extractor PlantNet, which was pre-trained on plant data such as the PlantCLEF dataset for the LifeCLEF 2017 competition. The proposed system has been carried out for plant disease detection as well as vitamin deficiency by various sensors [3]. On the basis of the image gathered, disease diagnosis for various areas of plants is done in five steps: image processing, image acquisition, feature extraction, segmentation, and classification with a classifier.

The majority of plant illnesses are visible on the leaves or stems [4]. Because of the diverse nature of visual examples, the precise evaluation of these outwardly observed illnesses, bugs, and characteristics has not been concentrated at this time. As a result, there is a growing demand for increasingly explicit and modern picture design knowledge. It will be extremely beneficial to a variety of farmers. An illness detection and classification system are based on Android [5]. Young farmers using cellphones will be able to take photos of sick plant leaves using the planned technique. The image will be sent to the application, which will do the categorization.

A clustering-based image threshold is performed using Otsu's technique [6]. This initiative employs an innovative idea to identify impacted crops and give agriculture industry remediation strategies. The diseased region of the leaf is segmented and examined using the k-means clustering technique. The photos are put into the application, which uses them to identify ailments. It is an excellent option for the agricultural population, especially in rural villages. It is a good approach for lowering clustering time and the size of the infected area. It can be used to create a plant disease detection system for helping farmers to detect plant disease infection early [7]. It also dedicates future studies on automatically estimating the security of the disease.

SVM [8] have analyzed the data after that it classifies that data and then the regression is done. Detection time and accuracy are two parameters for disease detection. Various techniques of segmentation [9], feature extraction, and classification are reviewed and discussed in terms of various parameters.

The VGGnet16 architecture is used to create a convolutional neural network model [10] for the recognition of sick and healthy leaves, several optimizers are tested to examine the accuracy. The summarized comments on related research are tabulated in Tables 1 and 2 depicts a list of methods and accuracy values for crop disease detection from existing research.

3 Workflow for Tomato Plant Disease Detection

3.1 Inception V3

The algorithm is implemented in Keras using the Inception V3 classifier. Inception V3 provided higher accuracy with fewer image data sets than other machine learning techniques such as SVM, Gaussian Nave Bayes, logistic regression, and linear discriminant analysis. The flow chart for Training and Testing of Inception V3 and the model for plant disease detection are represented in Figs. 1 and 2, respectively. TensorFlow and Keras API are used to implement the project. TensorFlow is an open-source package for dataflow programming that may be used to accomplish a variety of machine learning tasks. Keras API is used in conjunction with TensorFlow

Table 1 Comparison of related research in terms of technology, strength, and limitations

S. No.	Paper Title	Technology	Strength	Limitations
1	Tomato disease classification using AlexNet	AlexNet architecture	The AlexNet architecture for detecting tomato illness. The findings were quite accurate, indicating that the dataset is simple	It is possible that certain classes were recorded under different settings than others
2	Improved vision-based detection of strawberry diseases using a deep neural network	Deep neural network (DNN)	The single-step technique to plant disease detection reduces the network's computing complexity, which improves detection speed, which is its key benefit over the multi-step approach	The lower row of the two-stage cascaded detector substantially reduces the false-positive flaws in the upper row of the single-stage detector. Comparing the performance of their suggested approach in relative terms is tricky
3	Image processing-based detection of diseases and nutrient deficiencies in plants	Image processing, color moment method, color co-occurrence method	Fisher discriminant analysis was used to build the diagnosis model which showed an identification accuracy of 83.08–90.77%	Accurate nutrition analysis models are not possible because they vary with the growth stages
4	Disease detection in plant leaf using image processing technique	Image enhancement, k-means clustering, features extraction, image acquisition	As k-means clustering, this paper outlines the key image processing utilized for the diagnosis of leaf diseases. This procedure can help with a more precise diagnosis of leaf disease	It is only used for leaves and not for other parts of the plant
5	A novel approach for plant disease detection	A novel approach for plant disease detection	The illness information will be stored on an ORACLE database server in the system. This server will be accessed based on the classification result to offer management information that corresponds to the detected ailment as well as techniques for executing the inference	This works only on Android devices/Mobile

(continued)

Table 1 (continued)

S. No.	Paper Title	Technology	Strength	Limitations
6	Plant disease detection and its solution using image classification	Disease detection, production rate, k-means clustering, voice navigation, infection region	This project offers a method for dealing with leaf illnesses, as well as an analysis of the total percentage of the diseased leaf and its surroundings	It exclusively uses image processing and data mining to research crop diseases and characteristics
7	Plant disease detection using image classification	Plant disease, image processing, feature extraction, manual monitoring, and segmentation	Increased the rate of accuracy up to 95%	It required smart or mobile phones, and images will be uploaded to detect the disease
8	Wheat leaf disease detection using machine learning method- a review	Wheat disease detection, machine learning, SVM, segmentation, clustering, classification	Effective in high dimensional spaces It is also memory efficient	(1) This method is only for Wheat (2) SVMs do not offer probability estimations directly
9	Plant disease detection techniques	Segmentation techniques: partition clustering, edge detection, artificial neural networks	The study tells us about different techniques used for plant disease detection. Neural networks provide better accuracy which is higher than 90%	We need large datasets for training to get better accuracy
10	Deep learning application for plant diseases detection	Deep convolutional neural network (DCNN), SGD (stochastic gradient descent), Adagrad, root mean square propagation (RMSprop)	The SGD model reached an accuracy of 90% validation test. The AdaDelta model has proved 90% on the validation set	The SGD model reached an accuracy of 90% validation test. The AdaDelta model has proved 90% on the validation set

Table 2 List of methods and accuracy values for crop disease detection from existing research

Methods	Accuracy (%)
AlexNet architecture	85
Deep neural network (DNN)	81
Image processing, color moment method, color co-occurrence method	83.08–90.77
K-means clustering, features extraction, image acquisition, image enhancement	88
Image processing, segmentation, classification	72
K-means clustering, voice navigation, infection region	78
image processing, feature extraction, manual monitoring, and segmentation	95
machine learning, SVM (support vector machine), segmentation, clustering, classification	82
Partition clustering, edge detection: BPA (back propagation analysis), principal component analysis, singular value decomposition (SVD)	90
DCNN, SGD, AdaGrad, RMSprop	90

to support convolutional networks and to make debugging and exploration easier. For typical use situations, it also provides a straightforward and consistent interface.

Inception V3 is a Keras image classification model that may be loaded with ImageNet weights that have been pre-trained. It has a total of 22,314,794 parameters out of that 512,010 are trainable and 21,802,784 are non-trainable with an accuracy of 93.7%.

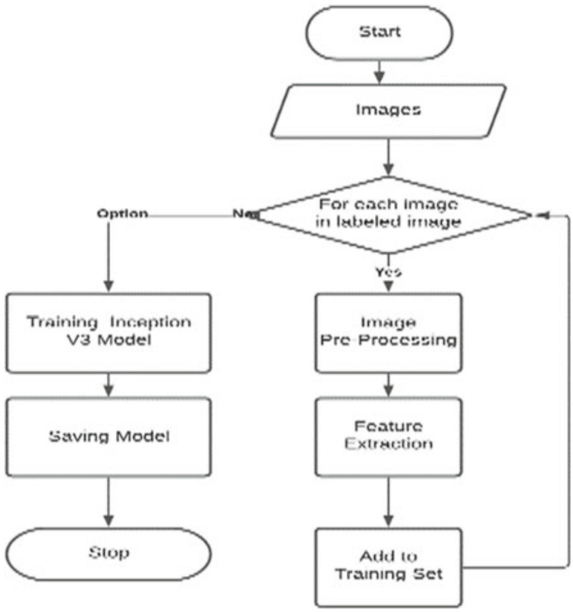
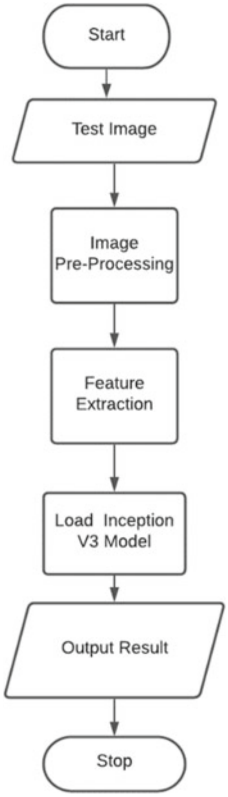


Fig. 1 Flow chart for training model of plant disease detection

Fig. 2 Flow chart for testing model of plant disease detection



The labeled datasets are divided into two types which are training data and testing data. The feature vector is generated for the training dataset using ImageDataGenerator from TensorFlow. The generated feature vector is trained under an Inception V3 Model classifier. The trained model is also given the feature vector for the testing data generated by ImageDataGenerator for prediction. These feature vectors are maintained as part of the training datasets. ImageDataGenerator is used to extract the feature vectors for the test image. The saved and trained classifier is given these created feature vectors to forecast the results. In the given flowchart, we train and test model using our given dataset. On the basis of leaves parameters, we decide the healthiness of it.

3.2 Dataset Description

The dataset selected for the proposed work is described in terms of parameters: disease name, number of original images, number of images for validation, total images, and remarks in Table 3.

Dataset Name	Tomato
Usage Information	License Data files © Original Authors
Visibility	Public
Maintainers	Nouaman Lamrahi Mohammed HMIMOU
Dataset URL	https://www.kaggle.com/noulam/tomato

For tomato, early blight leaf shows small brown target rings and dry dead plant tissue in center. If leaf has small water soaked or wet looking circular areas, it is called tomato bacterial spot. In tomato Early Blight, we find large dark brown blotches with a green gray edge on tomato leaf, and if there are small spot pale green or yellowish with indefinite margins, then it is tomato leaf mold disease.

For tomato Septoria Leaf Spot, we can observe small watered soaked circular spots under the leaf. White to yellow speckles to form on the upper surface of the leaves are symptoms for regular and Two-Spotted Spider Mite. In tomato leaf, it becomes yellow between veins in case of tomato yellow leaf curl virus. Lastly for tomato target spot, we can find spots with concentric target like circles and a velvety black, fungal lesion in the center.

3.3 Intermediate Results of Model Training

All input images are loaded into the Inception model's input layer at the start. The photos are then loaded into the Inception V3 architecture for feature extraction. Deep neural nets classify our photographs based on pre-trained knowledge using these feature-extracted images. The images are finally transmitted to the output layer.

The experimentation of the proposed system is carried out on tomato leaves from the selected dataset for 10 epochs with help Inception V3. Table 4 depicts the training loss, training accuracy, validation loss, and validation accuracy per epochs. It is observed that the best training accuracy, 88.98% is obtained at the 10th epoch and the best validation accuracy, 85.80% is obtained at the 8th epoch.

Figure 3 represents the training and validation percentage, whereas Fig. 4 represents the training and validation loss values.

4 Conclusion

This research study is on the design and development of CNN system for detecting tomato plant diseases. Tomato plant leaves are taken to identify its diseases. We use Inception V3 model for system, and the intermediate results are presented in terms of testing and validation accuracy.

Table 3 Dataset details in terms of parameters: disease name, number of original images, number of images for validation, total images, and remark

Directory	Disease name	Number of original images	Number of images from data set used for validation	Total images for specific diseases	Remarks
1	Tomato bacterial spot	1702	425	2127	It can impact the entire tomato plant's ground components, including the leaves, stems, and fruit
2	Tomato early blight	1920	480	2400	Alternaria solani is too responded. It may cause problems with a tomato's entire structure
3	Tomato late blight	1851	463	2314	It is caused by the Oomycete pathogen Phytophthora investments (P.infestans)
4	Tomato leaf mold	1882	470	2352	It is caused by P.fulva
5	Tomato septoria leaf spot	1745	436	2181	It is caused by Septoria lycopersici
6	Tomato spider mites two-spotted spider mite	1741	435	2176	It causes the area's mesophyll cells to collapse, resulting in extremely minute white chlorotic patches on the leaves
7	Tomato target spot	1827	457	2284	They contain small dark lesions on their leaves

(continued)

Table 3 (continued)

Directory	Disease name	Number of original images	Number of images from data set used for validation	Total images for specific diseases	Remarks
8	Tomato yellow leaf curl virus	1961	490	2451	It is a virus from the genus Begomovirus and the family Geminiviridae
9	Tomato mosaic Virus	1790	448	2238	It contains the mosaic virus
10	Tomato healthy	1926	481	2407	Healthy leaves

Table 4 List of training loss, training accuracy, validation loss, and validation accuracy per epochs

Epoch No.	Training loss	Training accuracy (%)	Validation loss	Validation accuracy (%)
1	2.94	68.01	1.91	77.27
2	2.00	79.62	2.67	78.02
3	1.98	82.29	2.49	79.69
4	2.07	83.49	2.59	80.83
5	1.87	85.12	2.63	81.42
6	1.76	87.15	2.67	82.36
7	1.85	87.17	2.84	82.79
8	1.74	87.89	2.28	85.80
9	1.64	88.53	3.38	81.57
10	1.74	88.98	2.59	84.84

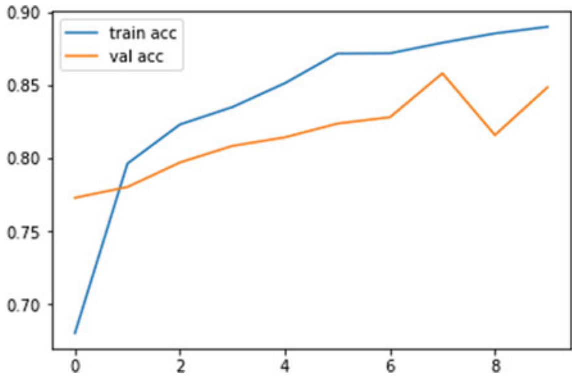
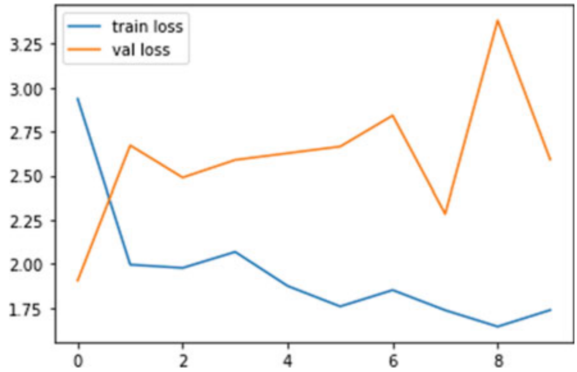


Fig. 3 Training accuracy and validation accuracy

Fig. 4 Training loss and validation loss



For image segmentation and image detection approaches, we employed a variety of algorithms that may be utilized to identify and classify various plant diseases automatically. For the experiment purpose, we have taken the tomato leaf data from the Kaggle website. In the dataset, there are 10 diseases and also contain disease-infected parts of images. Around 15% of image samples are used for validation of the designed classifier model.

It is observed that the best training accuracy, 88.98% is obtained at the 10th epoch and the best validation accuracy, 85.80% is obtained at the 8th epoch. The average training accuracy is 83.85%, and the average validation accuracy is 81.45%. In future, this study is useful for the next research on this domain. In past, there have been numerous healthy ways to combat diseases in recent years, but if the disease is detected early on, it does far less damage and is easier to cure. On the other hand, if it is discovered at a later time, it might result in a very low yield, as well as additional pesticide and manure costs. The farmer will not be able to avoid all types of losses simply by detecting disease. That's why, we studied on it and worked it.

References

1. Georgantopoulos P, Constantinopoulos S, Kosmopoulos D (2019) Tomato disease classification using AlexNet
2. Kim B, Han YK, Park JH, Lee J (2021) Improved vision-based detection of strawberry diseases using a deep neural network. Front Plant Sci 11. https://doi.org/10.3389/fpls.2020.559172
3. Ghorai A Mukhopadhyay S, Kundu S, Mandal S, Roy Barman A, Roy M, Jash S, Dutta S (2021) Image processing based detection of diseases and nutrient deficiencies in plants. 25(24)
4. Padmavathi K, Thangadurai K (2016) Disease detection in plant leaf using image processing technique. Indian J Sci Tech 9:1–6
5. Sarda S, Sormare S, Rahinj U (2019) A novel approach for plant disease detection. 5(3)
6. Saradhambal G, Dhivya R, Latha S, Rajesh R (2014) Plant disease detection and its solution using image classification
7. Saranya, Deepthi M, Bhuvaneshwari M, Gomathy R (2019) Plant disease detection using image processing. 8(3)

8. Dixit A, Nema S (2018) Wheat leaf disease detection using machine learning method—a review. IJCSMC 7(5)
9. Khan U, Oberoi A (2019) Plant disease detection techniques: a review. https://doi.org/10.13140/RG.2.2.10724.30081
10. Fatimazahra J Anas H, Bouaaddi A (2019) Deep learning application for plant disease detection. 1–6. https://doi.org/10.1145/3372938.3372983

Astute Farm Monitoring Using WSN and AI—A Solution for Optimally Monitoring Environmental Conditions

Sanjivani Kulkarni, Shilpa Budhavale, and Vaishali Langote

Abstract Agriculture is very significant economic sectors in every country. Farm Monitoring aims to accomplish exact management of irrigation, fertilizer, disease, and insect prevention in crop farming. In agricultural area, wireless sensor networks (WSNs) are used to gather data and communicate it to servers over a wireless link. Eighteen input characteristics were utilized to create the model, and crop yield was found and organized into three key components. When creating the multiclass model, the relative significance of the components is taken into account. For categorization of three crops: rice, groundnut, and sugarcane, an objective function is defined. A multiclass model based on a hybrid deep learning classifier approach (CNN + LSTM) is used. Furthermore, data visualization analysis is utilized to identify essential approaches in progress of smart agriculture that efficiently increase efficacy of production and assure agricultural product quality. Use of smart agriculture is progressively being incorporated into agricultural production, and advent of Internet of Things (IoT) is giving it a technological boost. Agricultural tasks precisely accomplished using the IoT' detecting, transmission, observing, and input capacities, which saves farmers' time and enhances crop yields and advantages them in long run. We installed smart agriculture IoT equipment in farm for monitoring reasons and used the algorithm in our research to do an actual-scenario analysis; the findings show that this suggested scheme is actually practical. The categorization findings are compared to the results acquired from on-the-ground agricultural specialists.

Keywords Data visualization analysis · Deep learning classifier · Hybrid classification · Smart farming

S. Kulkarni (✉) · S. Budhavale · V. Langote
Computer Science and Engineering, DVK MIT World Peace University, Pune, India
e-mail: sanjivani.kulkarni@mitwpu.edu.in

S. Budhavale
e-mail: shilpa.budhavale@mitwpu.edu.in

V. Langote
e-mail: vaishali.langote@mitwpu.edu.in

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_6

1 Introduction

Agriculture is most fundamental sources of income for people in India, and it is critical to the growth of the rural economy [1]. The use of information technology, which is extensively employed in environmental monitoring, industry, and other parts of life, is lacking in conventional agriculture production [2]. WSNs are a widespread use of information technology in agriculture, capable of monitoring soil data in specific areas and helping use of traditional irrigation systems. By gathering data during the agricultural production process, it has the ability to improve irrigation methods and make agriculture smarter [3].

WSNs are vital in smart agriculture because they monitor and collect data from agricultural areas for different uses. Precision irrigation, fertilizers, and pesticides are used in agriculture to improve soil fertility, decrease water waste, fertilizer misuse, and sicknesses. WSN sensor nodes in fields may wirelessly communicate collected soil data to the sink node [4]. In [5], the authors use WSNs to collect data on temperature, soil moisture content, humidity, and wind speed.

Due of mediators' lack of information regarding crop output and yield, full-season farmers are paid less (agents for bar-gaining). Farmers' efforts would be worthwhile if they were well trained in crop production, yield pricing, and crop marketing. If not, agricultural yields will be reduced, impacting the economy. As a consequence, if the government supports these talents in addition to traditional farming methods, the country's economy will likely improve [6].

Expert-driven FS and extraction are required in the case of classical ML approaches. A professional must create a feature extraction method that can generate the most relevant characteristics and input them into a traditional ML classifier. After that, the classifier is trained to learn from data and apply what it has learned to fresh data in order to make a classification judgement.

However, recently [7, 8], amazing results were produced using deep learning approaches, changing the area of text and object categorization. Rather of depending on handmade features, these methods may be utilized as end-to-end approaches since they function by automatically learning the essential characteristics from the raw data supplied as input, without the requirement for expertise.

The current study aimed to offer the CNN–LSTM hybrid deep learning technology as a viable and high-potential strategy for addressing numerous computer vision difficulties in agriculture. The technology supports mixed crop farming and assists farmers who have access to the system at any time and from any location.

That's what this study recommends. Among the features of our intelligent agriculture platform are a Python-based Internet application that assists in the collecting of data from IoT devices [9]. The recommended intelligent agriculture system uses affordable sensors, making it more appealing. This research analyses data from the intelligent agriculture system and publicly available datasets. The research looks at a farm's environmental conditions and crops to grow. In addition to farm environmental aspects, the study studies how weather conditions affect environmental changes and which crops are ideal for the farm depending on time sequence. Data

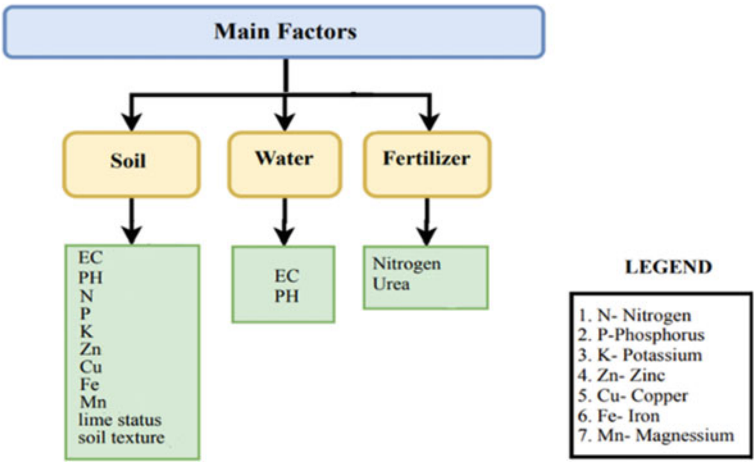


Fig. 1 Environmental factors

on soil, water, and fertilizer were gathered for numerous crops from three distinct locations of Maharashtra, India, to create the multiclass model.

The following principles are included in our suggested data analysis approach: (1) IoT sensors are used to collect data from farm fields, (2) using WSN, data sent from a farm field to a device, (3) data collection and standardization, (4) data analysis to examine relationship among various environmental factors, and as a result, farmers' rules of thumb; (5) determining if a selected crop has been planted in the proper soil, and (6) the hybrid multiclass classifier system is compared to the outcomes of other traditional classifiers including Naive Bayes, Random Forest, and SVM and proposed CNN + LSTM model (Fig. 1).

The proposed data analysis technique has been put to the test, and it has helped to determine if locally grown crop is a good fit. The suggested method is feasible and helps farmers comprehend their farm's environmental indices, according to results of the experiments. IoT devices can measure atmospheric pressure, air temperature, humidity, soil moisture content, illumination, and soil electrical conductivity [4]. Sensors will be used to assess soil macronutrients such as nitrogen, phosphorus, and potassium, as well as other soil parameters such as moisture, pH, and temperature.

2 Literature Review

Foreign developed nations began collecting agricultural data earlier, and their study and implementation of intelligent agriculture is more advanced. The agricultural environment monitoring service system was conceived and constructed by Hou et al. [10]. The system collected photos of soil and crops using wireless sensor technology and GPS positioning technology, allowing for remote crop monitoring and crop management, as well as the utilization of solar power to assure the equipment's proper operation. The Climate Corporation used climate, geographic location, and other factors to give natural catastrophe insurance to farmers, ensuring that agricultural production was protected to some extent [11]. Various sensors are used to measure soil factors including as temperature, pH, light, humidity, and moisture [12]. Utilizing an Analog to Digital Converter, the values are converted to digital and serially transferred to cloud using a Raspberry Pi. Outcome is shown on a laptop or through a mobile app. With the use of IoT, the system monitors the entire soil parameters. Soil factors such as pH, soil moisture, humidity, and temperature are continually monitored utilizing sensors in order to ensure effective crop yield. The creation of an optical transducer [13] is used to construct a system in which soil fertility is improved, and soil quality is improved. Low, medium, and high levels of NPK are achieved. The data is collected using an Arduino microcontroller, and the analog output is transformed to digital.

People are presented with a greater data system as information advances that has drew the attention of domestic and foreign academics that specialize in agricultural data analysis. Lamehari et al. fostered the rural construction by examining farming ecological information [14] that help makers and middle person organizations in settling on better choices, smoothing out the dynamic interaction, and accomplishing the objective of expanded horticultural usefulness and logical normal asset the executives. Gabriel and colleagues created and implemented a system for monitoring soil and analyzing soil fertility, as well as giving farmers with actionable advice for soil improvement. Li Xiufeng et al. proposed a visual intuitive framework that could convey network information administrations to clients and make information examination more straightforward [15]. Kursa et al. [13] used Boruta, an all-relevant feature selection approach that aggregates all features that are crucial to the result in certain situations. Most classic feature selection algorithms, on the other hand, use a minimally optimum strategy in which they rely on a limited group of characteristics that produce the least amount of error on a chosen classifier. Marcano Cedeno et al. [16] suggested a component choice procedure dependent on consecutive forward choice utilizing a feed forward neural organization to decide forecast mistake as choice models.

Sensor technology, Quick Response (QR) Code technology, RFID technology, and embedded system technology have all been created for practical execution of IoTs. The use of sensors is one of them. It is a type of sensing device that can detect information from objects and converts it other forms for outcome based on particular criteria, in order to meet information transmission, recording, processing, and control needs [17].

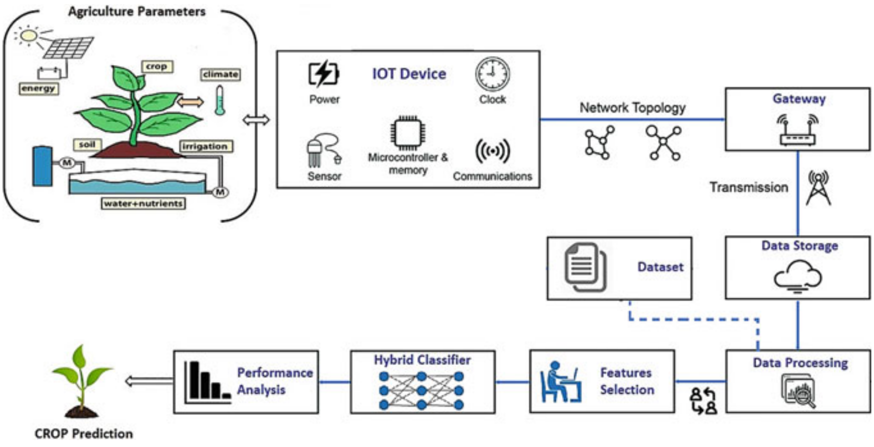


Fig. 2 Proposed system architecture

3 Proposed Work

3.1 *Proposed System*

In smart agriculture, the use of IoT technology leads to better agricultural production management, environmental monitoring of production and aquaculture, and quality and safety control of agricultural products. It has the ability to identify problems and offer a management platform for agricultural production, animal husbandry, and aquaculture at crucial locations, ensuring the quality of agricultural products. Figure 2 depicts the suggested design of a smart agricultural IoT system.

3.2 *Components Used in Proposed System*

The device monitors the farm and gives the farmer several forms of information about the present condition based on the readings of various sensors like humidity, temperature, soil moisture, UV, IR, and soil nutrients [4]. Farmers' can take quick action which will help them in increasing the farming production and make the most use of natural resources, making their product (crop yield) ecologically friendly. By appropriately monitoring the many current conditions, our proposed system will boost the amount and quality of the crops. It is an IoT device that uses the "Plug and Sense" principle. Laptops and smart phones can display real-time data for several factors.

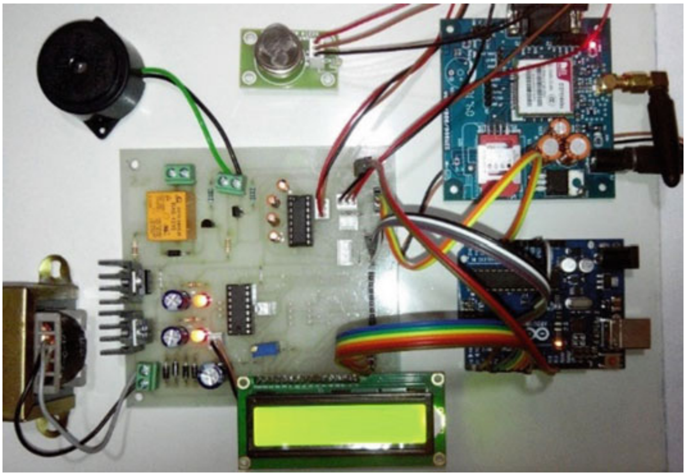


Fig. 3 Working IoT model used farm

Following are several important sensors details which are used to get live farm reading. Figure 3 shows the organization of various sensors to Arduino device and laptop/computer to get live farm reading. 1. Soil moisture sensor 2. Soil temperature sensor 3. Soil pH sensor.

3.3 System Modules

Following are several important modules which are used during the execution of proposed work for crop prediction.

1. Data Cleaning and Normalization: Data cleaning and normalization are the initial steps in data analysis. For IoT data transfer, our system leverages 4G networks. The data is cleaned first, as farm environments often change in a linear rise or decline pattern, it seldom fluctuates dramatically.
2. Features Selection: After performing the preprocessing important features are selected from the dataset. Given the input model, the sample data is divided by picking the best features from all of the characteristics. The feature selection procedure based on the tree model may draw specific higher importance features [18], allowing important characteristics to be selected from a large number of options. Figure 4 shows the features with their features importance value in descending order.
3. ML Classification: To extract and identify essential information from massive data, data mining and machine learning methods are utilized. In our work, we have used Naive Bayes [19], SVM [20], and RF [12] algorithm for classification.

Fig. 4 Feature importance calculation

Cu	0.164475
P	0.154644
Mn	0.147950
EC	0.134915
S	0.128977
N	0.091529
B	0.049403
Zn	0.047928
Fe	0.037074
K	0.022646
OC	0.014967
pH	0.005491

4. Hybrid Deep Learning Classifier (CNN + LSTM): CNN [7] is often used in feature engineering because of its ability to concentrate on the most obvious aspects in the line of sight. Because it has the virtue of expanding according to the sequence of time, LSTM [8] is often used in time series. Based on the features of CNN and LSTM, a stock forecasting model based on CNN–LSTM is constructed. The model's basic structure comprises CNN and LSTM, as well as an input layer, a 1D convolution layer, and a pooling layer.

Algorithm

(1) Input: Farm dataset with multiple features is given as input to the CNN–LSTM model.
(2) Data Normalization: As there is a wide gap in the input dataset, the data normalization technique is used to normalize the data in order to improve the train the model output.
(3) Initialize the weights and biases of each layer of the CNN–LSTM during network setup.
(4) CNN layer calculation: The input data is sequentially processed through the CNN layer's convolution layer and pooling layer, the input data is feature extracted, and the output value is obtained.
(5) Calculation of the LSTM layer: The LSTM layer calculates the CNN layer's output data, and the output value is acquired.
(6) Output layer: The final result of the LSTM layer is fed into the complete connection layer to acquire the output value.
(7) The associated error is computed by comparing the output value generated by the output layer to the true value of this piece of data.
(8) To determine if the end condition has been reached, a predetermined number of cycles must be performed, weight must be below a specific threshold, and forecasting error rate must be below a specified threshold. The CNN–LSTM network will be updated, and if one of the end conditions is satisfied, step 10 will be conducted; otherwise, step 9 will be performed.

(9) Backpropagation of estimated errors: Continue training the network by propagating the estimated error in the reverse way, updating the weight and bias of each layer, and returning to step 4.
(10) Save the model: Save the learned model for crop prediction with fresh data.
(11) Output result: To finish the forecasting procedure, output the repaired results.

Figure 5 shows the CNN + LSTM model which is used for training and testing purpose for crop prediction. It includes several layers namely convolution, max pooling, flatten, dense, and dropout. Additionally, it used two LSTM layers after max pooling layer. Model uses ReLu as activation function. And dropout probability is set to 0.01. The complete system runs for 10 epoch, and accuracy and loss are calculated to get the performance of proposed hybrid model.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 15, 64)            1664
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 14, 64)            0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 14, 32)            6176
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 13, 32)            0
_________________________________________________________________
lstm (LSTM)                  (None, 13, 128)           82432
_________________________________________________________________
lstm_1 (LSTM)                (None, 13, 128)           131584
_________________________________________________________________
flatten (Flatten)            (None, 1664)              0
_________________________________________________________________
dense (Dense)                (None, 64)                106560
_________________________________________________________________
dropout (Dropout)            (None, 64)                0
_________________________________________________________________
dense_1 (Dense)              (None, 16)                1040
_________________________________________________________________
dropout_1 (Dropout)          (None, 16)                0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 17
=================================================================
```

Fig. 5 CNN + LSTM model

4 Result and Discussion

4.1 Dataset Description and Experimental Setup

All the experiments are performed on environment of Intel i5, 3.2 GHz, 8 GBs of RAM, 1 TB of hard disk, 512 GB of SSD, and Windows 10 operating system. Anaconda (Jupiter Notebook) IDE is used and Python 3.7 technology is used. Soil dataset to predict the crop for Maharashtra district (Pune/Raigad/Nagpur) is downloaded from https://soilhealth.dac.gov.in/PublicReports/NSVW_total. It contains pH, OC, EC, N, K, P, S, Zn, Fe, Mn, Cu, B, etc., features.

4.2 Evaluation Metrics

The accuracy of chosen classifiers is assessed using a variety of performance indicators. The classification accuracy, precision, and recall are assessed to confirm the results of the proposed hybrid deep learning classifier and other classifiers. The formula for calculating these measures is as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \qquad \text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad F_1\ \text{Score} = \frac{2 * \text{Precisioion} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

True positive, true negative, false positive, and false negative values are represented as TP, TN, FP, and FN, respectively. Figures 7 shows the accuracy, precision, and recall scores of all the classifiers. NB had the lowest performance when compared to the others making it inappropriate for learning complicated structures for the subject data. All other classification models were surpassed by the proposed system (CNN + LSTM), which had the best accuracy, precision, recall, and F_1 Score.

4.3 Result Analysis

With increase in number of epoch, the accuracy remains nearly same as shown in Fig. 6a. Figure 6b shows the training and validation loss comparison of CNN + LSTM model for 10 epochs. The validation loss after 10 epochs is 0.08. With increase in number of epoch, the loss is reduced which can be seen in Fig. 6b. Figure 7 shows the training and validation accuracy comparison of CNN + LSTM model for 10 epochs. The validation accuracy after 10 epochs is 95.20%.

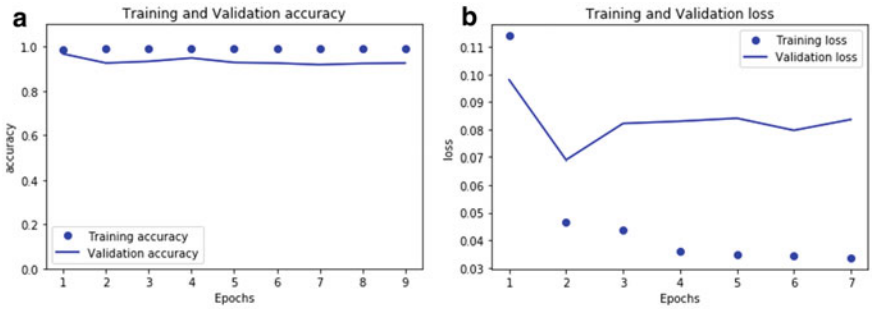


Fig. 6 **a** Training and validation accuracy and **b** training and validation loss comparison of CNN + LSTM graph

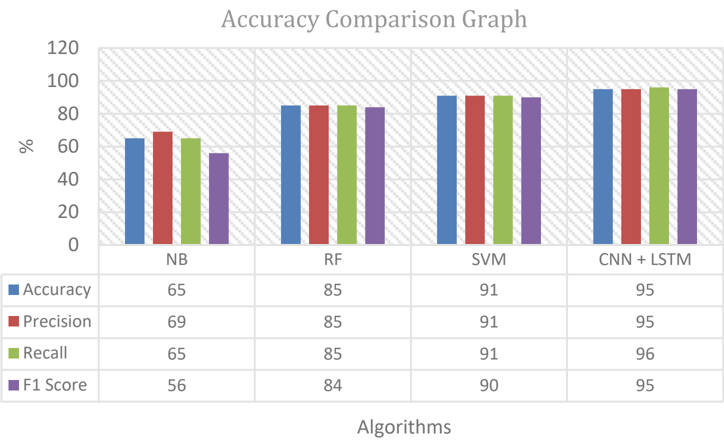


Fig. 7 Performance parameters comparison graph

5 Conclusion

Machine learning is a scientific field that has applications in crop prediction research. Farmers are keen to find out how much they can expect to produce; therefore, crop cultivation forecasting is critical in agriculture. Previously crop yield prediction is used to be done by taking into consideration farmers' fundamental knowledge of specific areas of land and the crops that will be cultivated there. Several machine learning and deep learning technologies are utilized and analyzed in agriculture to anticipate future crop production or which crop to be take in that soil. Our study focused on use of IoT sensors to collect real-time data from the farm, as well as employing feature selection methods to extract relevant features for prediction models. Propose system uses traditional machine learning algorithms and proposed hybrid classification model (CNN + LSTM) to forecast the crop perdition. Our findings reveal that Hybrid CNN + LSTM classifier beats other approaches with tenfold and 80–20% data splitting range and gives maximum accuracy of 95.20%.

References

1. Cervantes-Godoy D, Dewbre J (2010) Economic importance of agriculture for poverty reduction. Agriculture and Fisheries Working Papers, No. 23, OECD Publishing
2. Elijah O, Rahman TA, Orikumhi I, Leow CY, Hindia MN (2018) An overview of internet of things (IoT) and data analytics in agriculture: benefits and challenges. IEEE 5(5):3758–3773
3. Roopaei M, Rad P, Choo KR (2017) Cloud of things in smart agriculture: intelligent irrigation monitoring by thermal imaging. IEEE 4(1):10–15
4. Muangprathub J, Boonnam N, Kajornkasirat S, Lekbangpong N, Wanichsombat A, Nillaor P (2019) IoT and agriculture data analysis for smart farm. Comput Electron Agric 156:467–474
5. Shouyi Y, Leibo L, Renyan Z, Zhongfu S, Shaojun W (2013) Design of wireless multi-media sensor network for precision agriculture. China 10(2):71–88
6. Tseng F-H, Cho H-H, Wu H-T (2019) Applying big data for intelligent agriculture-based crop selection analysis. IEEE Access 7:116965–116974
7. Barburiceanu S, Meza S, Orza B, Malutan R, Terebes R (2021) Convolutional neural networks for texture feature extraction. Applications to leaf disease classification in precision agriculture. IEEE Access 9:160085–160103
8. Chen S, Han X, Shen Y, Ye C (2021) Application of improved LSTM algorithm in macroeconomic forecasting. Comput Intell Neurosci
9. Chen J, Yang A (2019) Intelligent agriculture and its key technologies based on Internet of Things architecture. IEEE Access
10. Hou L, Zhao S, Xiong X et al (2016) Internet of things cloud: architecture and implementation. IEEE Commun Mag 54(12):32–39
11. Manogaran G, Varatharajan R, Lopez D et al (2018) A new architecture of Internet of Things and big data ecosystem for secured smart healthcare monitoring and alerting system. Futur Gener Comput Syst 82:375–387
12. Chai Z, Zhao C (2020) Enhanced random forest with concurrent analysis of static and dynamic nodes for industrial fault classification. IEEE Trans Industr Inf 16(1):54–66
13. Kursa MB, Jankowski A, Rudnicki WR (2010) Boruta–a system for feature selection. Fund Inform 101:271–285
14. Long TB, Blok V, Coninx I (2016) Barriers to the adoption and diffusion of technological innovations for climate- smart agriculture in Europe: evidence from the Netherlands, France, Switzerland and Italy. J Clean Prod 112:9–21
15. Thierfelder C, Rusinamhodzi L, Setimela P et al (2016) Conservation agriculture and drought-tolerant germplasm: Reaping the benefits of climate- smart agriculture technologies in central Mozambique. Renewable Agric Food Syst 31(5):414–428
16. Marcano-Cedeno A, Quintanilla-Domínguez J, Cortina-Januchs MG, Andina D (2010) Feature selection using sequential forward selection and classification applying artificial metaplasticity neural network. In Industrial electronic conference. In: 36th Annual conference on IEEE industrial electronics society, Glendale-USA
17. Jiang S, Lian M, Lu C et al (2018) Ensemble prediction algorithm of anomaly monitoring based on big data analysis platform of open-pit mine slope. Complexity

18. Reif DM, Mckinney A, Jason H (2006) Feature selection using a random forests classifier for the integrated analysis of multiple data types. 1–8
19. Manek AS, Shenoy PD, Mohan MC et al (2017) Aspect term extraction for sentiment analysis in large movie reviews using Gini Index feature selection method and SVM classifier. WWW 20:135–154
20. Huang K, Jiang H, Zhang X (2017) Field support vector machines. IEEE Trans Emerging Top Comput Intell 1(6):454–463

Intelligent Systems in Internet of Things and Cloud Computing

A Comprehensive Literature Review of Artificial Intelligent Practices in the Field of Penetration Testing

Dipali N. Railkar and Shubhalaxmi Joshi

Abstract Penetration testing (PT) is a commonly available approach to dynamically assess the defenses of a computer network via preparation and execution of every probable attack to identify and utilize available existing vulnerabilities. Traditionally, the PT takes place in a manual way where the pentesters are initially required to investigate the target system, then utilize the explored vulnerabilities in various methods for penetrating the system and compromising network resources. It is a lengthier, difficult, and laborious process which could not be simply formulated and is also vulnerable to errors. So, recently, automated PT (APT) models have been designed using artificial intelligence (AI) techniques based on reinforcement learning (RL). Since numerous APT models are presented in the literature, this paper focuses on the compressive review of the recently presented APT tools for network and web-based applications including cloud and IoT. The review is made to explore the significance of AI techniques in improving the PT and vulnerability detection of the system. The existing techniques are elaborated, identified the importance, and other characteristics. Besides, a brief result analysis of the reviewed models takes place to highlight the experimental results. Finally, a number of future directions in the field of PT using RL techniques with respect to scalability, accuracy, and requirement of real-time detection of vulnerabilities related to network efficiency are discussed.

Keywords Penetration testing · Network efficiency · Vulnerability detection · Artificial intelligence · Reinforcement learning

D. N. Railkar (✉) · S. Joshi
Department of Master of Computer Application, MIT-WP University, Pune, Maharashtra, India
e-mail: railkar.dipali@gmail.com

S. Joshi
e-mail: shubhalaxmi.joshi@mitwpu.edu.in

D. N. Railkar
Department of Master of Computer Application, PCCoE, SPPU University, Pune, Maharashtra, India

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_7

1 Introduction

With the widespread utilization of computer networks and the frequent existence of security incidents, cyber-security becoming more important problem. One of the powerful method to resolve this problem is to measure the system security feature through penetration testing (PT) [1, 2]. The PT is an authorized proactive attack method applied to assess the security feature of any organization starting from digital assets to network, especially to detect its existing vulnerabilities by performing ethical attacks on it. The technique is well-grounded, and many commercial tools are accessible to support the pentester in implementing the difficult process [3, 4]. In the field of cyber-security, ethical hacker needs defense training to implement attacks in the training environments, so that beginners could get practical knowledge of handling security incidents. The PT technique is well suited for performing the attacker's role in this type of scenario. In general, to perform PT, pentester first needs to investigate the targeted system, next exploits the detected susceptibilities in various forms for penetrating the systems, and compromises network resources in proof-of-concept attack [5]. This procedure is a complex task, laborious, and inefficient which needs a large amount of tacit knowledge that could not be easily formalized, as well as prone to human error. Thus, increasing number of persons attempted to utilize method-based attack planning to create the attack by on the targeted system.

In realistic terms, PT as demonstrated in Fig. 1 is a multistage method that frequently needs a higher amount of expertise and competence because of the complex digital assets like large and medium networks. Basically, the study has examined the feasibility of automatic tools for the distinct PT phases (identification, exploitation, and reconnaissance) to relieve the human experts from the burden of repeated tasks [6]. But, automation by themselves does not attain considerable advantages based on outputs, time, and resources. Since PT is a very interactive and dynamic procedure of making decisions and exploring that needs critical and advanced cognitive skills which are hard to duplicate using automation. A common question arises regarding the ability of AI to offer a possible solution which exceeds simple automation to accomplish experts-like output [7].

In other areas of study, AI has proved to be very useful for offloading work from humans as well as dealing with detail and depth that humans cannot address accurately or fast enough [8]. Fast development in AI and especially the machine learning (ML) sub-region made us believe that AI-related PT schemes use well-established algorithms and methodologies to make consecutive decisions in uncertain environments which could connect the gap among expertise and automation that PT community experiences [9]. In this perception, the current PT framework and systems begin to move from performing experts' task to become more optimized, autonomous, and intelligent targeting that each present threats are efficiently and systematically checked with or without human interference [10]. Moreover, this system must improve resource usage by removing irrelevant directions and time-consuming and ensure that no threat is over-looked.

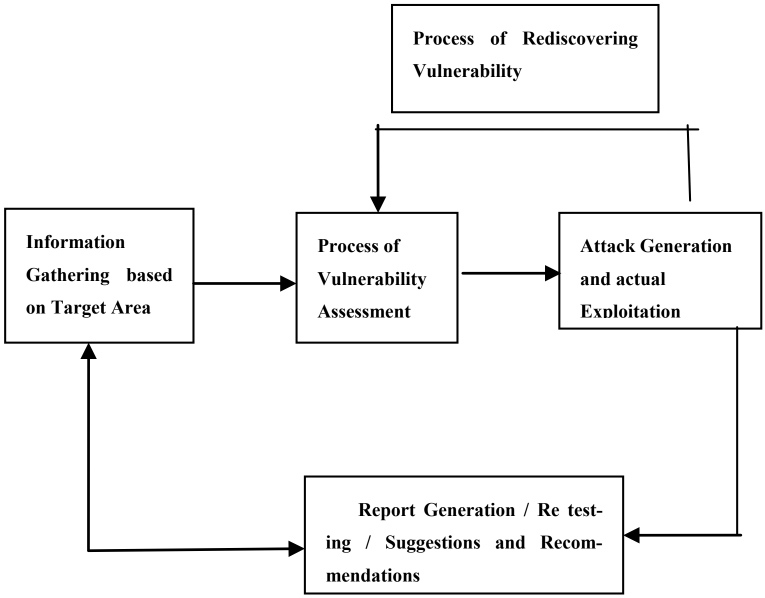


Fig. 1 Process involved in APT model

This paper aims to provide a complete survey of the recently presented APT tools for various applications including network. The survey is carried out for identifying the importance of AI techniques in boosting the PT and vulnerability detection of the system with respect to digital environment. The existing techniques are explained, acknowledged the importance, and other characteristics. Furthermore, a brief result analysis of the reviewed models takes place to emphasize the experimental results. Lastly, a number of future directions like scalability and requirement of real-time detection related to different parameters targeting accuracy, time, and human resources of vulnerabilities are discussed.

2 Existing Automated Penetration Testing Models

This section has discussed the recently developed APT models that exist in the literature as, Hu et al. [11] proposed an automated PT structure that utilizes deep reinforcement learning (DRL) for automating the PT process. The primary phase, can utilize the Shodan search engine for collecting appropriate server information for creating a realistic network topology, and utilize multi-host multistage vulnerability analysis (MulVAL) for generating an attack tree to that topology; typical search techniques were utilized for finding every feasible attack path from that tree and for building a matrix representation as required by DRL techniques. Secondary phase, it can utilize the deep Q-learning network (DQN) technique for discovering exploiting attack paths in feasible system.

Ghanem and Chen [12] presented and estimate an AI-based pentesting model that creates ML approaches such as RL for learning as well as reproducing average and difficult pentesting activity. The presented method was called as intelligent automated penetration testing system (IAPTS) comprising elements which combine with industrial PT structures for enabling them for capturing data, learning in experiences, and reproducing tests under the future same as testing cases. Casola et al. [13] introduced a technique which permits effortlessly executing a coarse-grained security estimation of cloud applications with automating the setup and implementation of PT. This technique relies on the skill of application structure and on the accessibility of catalog containing security compared with data gathered in several sources and appropriately connected.

Bhardwaj et al. [14] identified that blockchain technologies comprise security considerations such as non-competent, insufficient access, and irreversible transaction approaches. The attack vectors as these are not initiated on web portals and other applications. This investigation presents a novel PT structure to smart contracts and decentralized apps. Hassan et al. [15] presented a detection technique, SAISAN where the solution detects local file inclusion (LFI) defects of web applications. This study is also applied SAISAN with tools where the outcome of tools are related to manual PT technique to understand the accuracy of the technique. Koroniotis et al. [16] introduced a DL-based PT structure such as long short-term memory recurrent neural network-enabled vulnerability identification (LSTM-EVI). It can employ this structure with a novel cyber-security-oriented test bed that is smart testbed that contained both physical as well as virtual elements. The structure is estimated utilizing this testbed and on real-time data source.

Zhou et al. [17] presented an improved DQN named NDSPI-DQN for addressing the sparse reward issue and huge action space issue from large-scale conditions. Primary, it can be reasonably integrating five delays to DQN, containing noisy nets, soft Q-learning, dueling structures, arranged experience replay, and intrinsic curiosity method for improving the exploration performance. Secondary, it can decouple the performance and separate the estimators of neural network (NN) for calculating two elements of performance individually, for decreasing the performance spaces. Stetsenko [18], a novel technique of penetration time evaluation to data system was presented. It can be grounded on utilizing data on vulnerability and presenting an initial and comprehensive analysis of feasible options to data attacks execution utilizing a model for simulating. The stochastic Petri net with informational arcs is utilized to create a model.

The PT technique offered security protection by intrusion detection system/intrusion prevention system (IDS/IPS) tools Suricata from the minimum efficiency embedding IoT devices are projected in [19]. The PT tool was projected with emphasis on software tools like NMAP and Metasploit. It can be utilized testing technique was explained in deep. An IDS/IPS tools, Suricata was executed from embedding platform Raspberry Pi 3. Zhou et al. [20] presented a network information gain-based automated attack planning (NIG-AP) technique for achieving autonomous attack path detection. This technique formalizes PT as a Markov decision procedure and utilizes network data for obtaining the rewards that lead an agent for choosing optimum response performances for discovering hidden attack paths in the intruder perspectives.

Bojjagani et al. [21] examined mobile banking applications (MBA) of different banks running on two leading platforms of iOS and android using dynamic and static analysis. They presented threat models, to identify different susceptibilities severely. Also performed systematic research of many unknown susceptibilities, especially in mobile banking applications and exhibited how MBA is susceptible to man-in-the-middle attack (MitM) attack. They observed that some MBAs are utilizing simple HTTP protocols for transferring client information without concerning security requirements. Huizinga [22] covered the entire parts of the ML method, from generating traffic samples to pre-processing the information to testing and training models. The study result shows that classification and pre-processing could be done quickly during a pentest. For the constrained amount of classes, we determined, the resultant models are very precise.

Shi et al. [23] proposed the process and principle of conventional PT technique for large-scale networks and analyses its shortcoming in real-time application. Next, cyberspace search engine and network fingerprinting technologies combined together a PT architecture for large-scale network-based network fingerprint are realized. Speicher et al. [24] introduced the first methodology to conduct wide-ranging what-if analyses to consider mitigation in a conceptually well-established method. To compare and evaluate mitigation approaches, we apply simulated PT, that is, automatic detection of attacks, based on network system to which a subset of a provided set of mitigation actions. For example, variations to the system updates, configuration, network topology, and so on are employed.

Yadav et al. [25] presented first-of-its-kind, IoT-PEN, and a PT architecture for IoTs. The architecture enables a client–server framework where each IoT node acts as a client and "a system with resource" as a server. IoT-PEN is a flexible, end-to-end automated and scalable PT architecture for determining each possible way attackers could break the targeted system using target graphs. Sandhya et al. [26] discussed the requirement to exploit PT, the advantage of employing Wireshark for the similar and continue to illustrate one approach of employing the tools to execute PT. Mostly, the network regions are vulnerable to security attacks by attackers. Lidanta et al. [27] aimed to estimate the security of WLAN network. It is performed by the PT approaches and vulnerability assessment. Penetration testing execution standard (PTES) has been utilized as a benchmark for the research. PT is performed by imitating various attacks, such as DoS, packet sniffing, and unauthorized access attacks.

Zhao et al. [28] used the rule tree model for achieving the automation procedure of PT, and all the chains of the rule tree save a whole attack procedure. By utilizing the results of PT, we present the security evaluation technique for meeting the National Institute of Standards and Technology (NIST) guideline, and it could create uncertain parameters of security evaluation strong. With the continuous increase of rule trees, the presented technique could enhance the effectiveness and accuracy of security evaluation. Qiu et al. [29] presented an automated generation method of penetration graph which enhances the network protocol beforehand creating the penetration graph that could decrease the unwanted data efficiently. The integrate penetration graph generation technique to the Common Vulnerability Scoring Scheme (CVSS) data, which increases the consistency of all the attack paths.

Zheng et al. [30] developed PTV which is a visual method for the PT method summarization-based auxiliary decision and visual narrative. It comprises two essential mechanisms: (i) an element that efficiently shows the outcomes of PT (ii) a visual interface that shows personalized PT path. To develop PTVis, PT path integrates PT tools and methods that are constructed by a customized exploration and cooperative multi-view, facilitating the exploration of PT. Arnaldy and Perdana [31] performed a security assessment website with PT terms. This PT can be implemented by the man-in-the-middle attack model. Still, this technique is extensively utilized by the attackers who are not accountable for the performance of Sniffing that employed to tap from a targeted system that focuses on searching for confidential data. We use some PT systems, such as brute force, SQL injection, and XSS (cross-site scripting) attacks.

Chaudhary et al. [32] employed ML method in the post-exploitation stage of PT for assessing the susceptibility of the network and henceforth, contributed to the automation procedure of PT. This study trained the agent by utilizing reinforcement learning by offering a suitable environment to detect sensitive documents and examine compromised networks. With various network platforms in training, we believe to simplify that the agent allows for extensive applications. Goutam and Tiwari [33] aimed at web application security. In the presented study, an architecture was constructed to analyze the susceptibilities. Afterward PT, according to the further vulnerabilities, architecture is developed that offer additional privacy to this website.

Ye et al. [34] developed a SQL blind bug related to second-order fragmentation redeployment. Through modeling, the SQL injection attack behavior is studied. Our technique could efficiently minimize the SQL injection and improve the performance. By the pervasiveness of the universal strength of IoT, the electric power data security defense system work has to be seriously considered. Jain et al. [35] designed a PT related Java Code Testing Tool (PJCT) for checking privacy attributes of provided Java code. The presented method could identify the absence or presence of seven characteristics presented originally for testing the privacy of the applications efficiently. In the study, the efficiency analysis of PJCT was displayed and compared to other methods like PIC was illustrated.

Singh et al. [36] explained and found specific scenarios that could demonstrate the difference in manual and automated methods for PT. There are a few conditions where manual testing operate more effectively when compared to vulnerability scanners or automatic scripts to find privacy problems in web application. The concept of web application susceptibilities was employed in the testing process, including Open Web Application Security Projects (OWASP), an online community devoted to web security. Bechtsoudis and Sklavos [37] introduced a PT framework and methodology to depict susceptibilities in all the network layers. In addition, performed a wide-ranging investigation of a network PT against a simulation lab setup, exposing network misconfiguration and security implications to its user and entire network.

Overstreet et al. [38] focus on testing an Amazon Echo vulnerability against a DoS attack. By utilizing wireless system, and assuming that attackers had accessed our personal system, they watched the network traffics of the Amazon Echo when it is under a DoS attack. Next, utilized one sample of Kali Linux to execute the attacks on the system, while other samples of Kali were utilized for monitoring the networks at the time of attacks.

3 Results and Discussion

This section inspects the recently developed APT techniques in the literature. Table 1 offers the comparative analysis of the APT models in terms of different measures.

Figure 2 investigates the acc_y, $prec_n$, and $reca_l$ analysis of the APT approaches. The figure demonstrated various results obtained by the different models such that the KNN model has obtained least acc_y, $prec_n$, and $reca_l$ values of 99.83, 99.67, and 100, correspondingly. Along with that the SVM model has obtained acc_y, $prec_n$, and $reca_l$ of 99.83, 99.67, and 100. In addition, the multilayer perceptron (MLP) model has tried to accomplish reasonably acc_y, $prec_n$, and $reca_l$ of 99.83, 99.67, and 100, respectively. However, the log short-term memory (LSTM) system has resulted in higher acc_y, $prec_n$, and $reca_l$ of 99.91, 99.83, and 100, respectively.

Figure 3 examines the $spec_y$, F_{score}, and AUC analysis of the APT methods. The figure stated that the KNN approach has reached toward $spec_y$, F_{score}, and AUC values of 99.66, 99.83, and 99.83, correspondingly. Also the SVM approach has reached to $spec_y$, F_{score}, and AUC values of 99.66, 99.83, and 99.83, whereas the MLP method has tried to accomplish reasonably $spec_y$, F_{score}, and AUC values of 99.66, 99.83, and 99.83. But, the LSTM methodology has resulted in superior $spec_y$, F_{score}, and AUC values of 99.83, 99.91, and 99.91, correspondingly.

Table 1 Comparative analysis of APT technique in terms of different measures

Methods	Accuracy	Precision	Recall	Specificity	*F*-score	AUC
KNN	99.83	99.67	100.00	99.66	99.83	99.83
SVM	99.83	99.67	100.00	99.66	99.83	99.83
MLP	99.83	99.67	100.00	99.66	99.83	99.83
LSTM	99.91	99.83	100.00	99.83	99.91	99.91

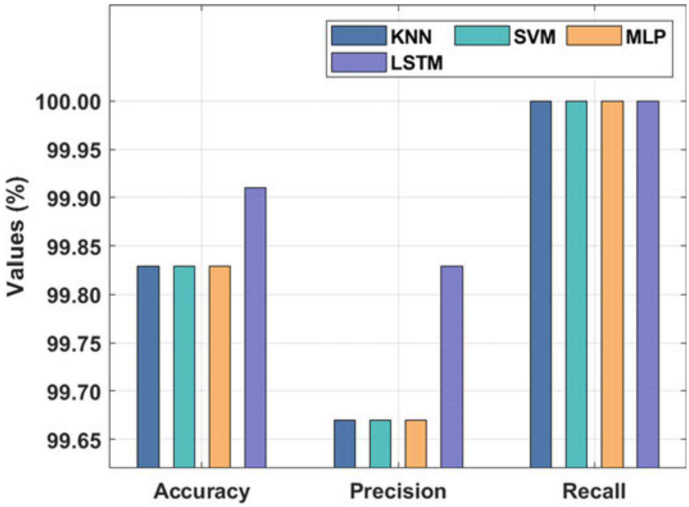


Fig. 2 acc_y, $prec_n$, and $reca_l$ analysis of APT techniques

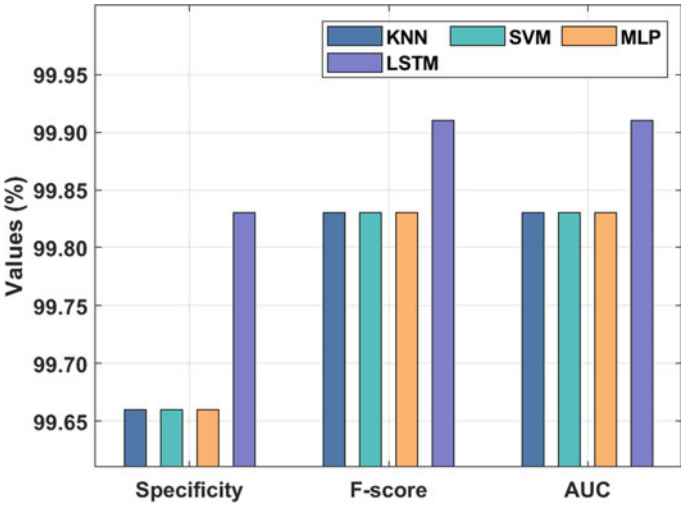


Fig. 3 $spec_y$, F_{score}, and AUC analysis of APT techniques

4 Future Directions

Several AI methods employ detection time as a quantifying metric, e.g., in terms of minutes or seconds. For the automated PT process, researchers have used existing vulnerability scanning tools and datasets. One possible study focus is to develop an AI-based method for identifying vulnerabilities in real time. To achieve this one can develop a system for continue scanning mechanism. To attain optimum accuracy, we proposed that future solutions must be developed for the specific system or application domain including a small network, instead of too generic or abstract. Abstract applications, like those developed to generate or search for common exploits over several hosts, might only identify on the surface types or simple vulnerabilities. As well, future AI methods need to test different phases of their model during the multiple PT phases (e.g., first compromise compared to post-exploitation), i.e., possible to facilitate more realistic learning and simulation. Also, we observed that many surveys employ simulations or test beds for assessment, instead of real-time systems. Therefore, one study possibility is to develop a suite of standardized networks or systems which is quite complex, realistic, and have features, i.e., typical of distinct practical applications. This would offer a general framework for evaluating and benchmarking their methods (similar to utilizing a common dataset to benchmark the several performance metrics of security methods). The study also implies that while implementing APT researchers used existing vulnerability scanning tool and based on that PT process is attempted. Also, assessment criteria must move away from time toward more qualitative measures, for instance, "impact or consequence" to determine the efficiency of AI application in vulnerability assessment or penetration testing methods.

5 Conclusions

In order to investigate the behavior of the existing APT models, this paper has performed a complete survey of the recently presented APT tools for various applications including the network architecture. The survey is carried out for identifying the importance of AI techniques in boosting the PT and vulnerability detection of the system. The existing techniques are explained, acknowledged the importance, and other characteristics. More over the PT models that are implemented with AI techniques are using Shodhan, Metasploit, MulVal, etc., for performing vulnerability assessment and penetration testing which are highly effective in terms of test coverage and accuracy. These PT models are implemented in the controlled environment, and we observed that these simulations or test beds need to incorporate the execution of post-exploitation process, as this will add value to the existing work.

The paper highlights the main impact of PT with support of RL techniques to real issues of vulnerabilities can be optimized and one can improve the performance of the security with respect to time consumption and accuracy. This will lead to organizations to have human error free PT framework at place. Furthermore, a brief result analysis of the reviewed models takes place to highlight the experimental results. Lastly, a number of future directions like scalability and requirement of real-time detection of vulnerabilities are discussed.

References

1. Creasey J, Glover I (2017) A guide for running an effective penetration testing program. CREST Publication, Slough, UK
2. Applebaum A, Miller D, Strom B, Korban C, Wol R (2016) Intelligent, automated red team emulation. In: Proceedings of the 32nd annual conference on computer security applications (ACSAC '16), Los Angeles, CA, USA, 5–8 December 2016, pp 363–373
3. Obes J, Richarte G, Sarraute C (2013) Attack planning in the real world. arXiv:1306.4044
4. Hoffmann J (2015) Simulated penetration testing: from Dijkstra to Turing Test++. In: Proceedings of the 25th international conference on automated planning and scheduling, Israel, 7–11 June 2015
5. Sarraute C. Automated attack planning. Available online: https://arxiv.org/abs/1307.7808
6. Sarraute C, Buffet O, Hoffmann J. POMDPs make better hackers: accounting for uncertainty in penetration testing. Available online: https://arxiv.org/abs/1307.8182
7. Jajodia S, Noel S (2008) Topological vulnerability analysis: a powerful new approach for network attack prevention, detection, and response. In: Algorithms, architectures and information systems security. Indian Statistical Institute Platinum Jubilee series, pp 285–305
8. Durkota K, Lisy V, Bosansk B, Kiekintveld C (2015) Optimal network security hardening using attack graph games. In: Proceedings of the 24th international joint conference on artificial intelligence (IJCAI-2015), Buenos Aires, Argentina, 25–31 July 2015
9. Maeda R, Mimura M (2021) Automating post-exploitation with deep reinforcement learning. Comput Secur 100:102108
10. Simon K, Moucha C, Keller J (2017) Contactless vulnerability analysis using Google and Shodan. J Univ Comput Sci 23:404–430
11. Hu Z, Beuran R, Tan Y (2020) Automated penetration testing using deep reinforcement learning. In: 2020 IEEE European symposium on security and privacy workshops (EuroS&PW). IEEE, pp 2–10
12. Ghanem MC, Chen TM (2020) Reinforcement learning for efficient network penetration testing. Information 11(1):6
13. Casola V, De Benedictis A, Rak M, Villano U (2018) Towards automated penetration testing for cloud applications. In: 2018 IEEE 27th international conference on enabling technologies: infrastructure for collaborative enterprises (WETICE). IEEE, pp 24–29
14. Bhardwaj A, Shah SBH, Shankar A, Alazab M, Kumar M, Gadekallu TR (2021) Penetration testing framework for smart contract blockchain. Peer-to-Peer Netw Appl 14(5):2635–2650
15. Hassan MM, Bhuyian T, Sohel MK, Sharif MH, Biswas S (2018) SAISAN: an automated local file inclusion vulnerability detection model. Int J Eng Technol 7(2–3):4
16. Koroniotis N, Moustafa N, Turnbull B, Schiliro F, Gauravaram P, Janicke H (2021) A deep learning-based penetration testing framework for vulnerability identification in internet of things environments. arXiv preprint arXiv:2109.09259
17. Zhou S, Liu J, Hou D, Zhong X, Zhang Y (2021) Autonomous penetration testing based on improved deep Q-network. Appl Sci 11(19):8823
18. Stetsenko IV, Savchuk V (2020) Information system penetration testing using web attack automated simulation. In: International conference on computer science, engineering and education applications. Springer, Cham, pp 396–406
19. Zitta T, Neruda M, Vojtech L, Matejkova M, Jehlicka M, Hach L, Moravec J (2018) Penetration testing of intrusion detection and prevention system in low-performance embedded IoT device. In: 2018 18th international conference on mechatronics-mechatronika (ME). IEEE, pp 1–5
20. Zhou TY, Zang YC, Zhu JH, Wang QX (2019) NIG-AP: a new method for automated penetration testing. Front Inf Technol Electron Eng 20(9):1277–1288
21. Bojjagani S, Sastry VN (2017) VAPTAi: a threat model for vulnerability assessment and penetration testing of android and iOS mobile banking apps. In: 2017 IEEE 3rd international conference on collaboration and internet computing (CIC). IEEE, pp 77–86
22. Huizinga T (2019) Using machine learning in network traffic analysis for penetration testing auditability

23. Shi P, Qin F, Cheng R, Zhu K (2019) The penetration testing framework for large-scale network based on network fingerprint. In: 2019 international conference on communications, information system and computer engineering (CISCE). IEEE, pp 378–381
24. Speicher P, Steinmetz M, Hoffmann J, Backes M, Künnemann R (2019) Towards automated network mitigation analysis. In: Proceedings of the 34th ACM/SIGAPP symposium on applied computing, pp 1971–1978
25. Yadav G, Paul K, Allakany A, Okamura K (2020) IoT-PEN: an E2E penetration testing framework for IoT. J Inf Process 28:633–642
26. Sandhya S, Purkayastha S, Joshua E, Deep A (2017) Assessment of website security by penetration testing using Wireshark. In: 2017 4th international conference on advanced computing and communication systems (ICACCS). IEEE, pp 1–4
27. Lidanta FZ, Almaarif A, Budiyono A (2021) Vulnerability analysis of wireless LAN networks using penetration testing execution standard: a case study of cafes in Palembang. In: 2021 international conference on ICT for smart society (ICISS). IEEE, pp 1–5
28. Zhao J, Shang W, Wan M, Zeng P (2015) Penetration testing automation assessment method based on rule tree. In: 2015 IEEE international conference on cyber technology in automation, control, and intelligent systems (CYBER). IEEE, pp 1829–1833
29. Qiu X, Jia Q, Wang S, Xia C, Lv L (2014) Automatic generation algorithm of penetration graph in penetration testing. In: 2014 ninth international conference on P2P, parallel, grid, cloud and internet computing. IEEE, pp 531–537
30. Zheng S, Wu Y, Wang S, Wei Y, Mu D, He H, Han D, Liao J, Chen H (2020) PTVis: visual narrative and auxiliary decision to assist in comprehending the penetration testing process. IEEE Access 8:194523–194540
31. Arnaldy D, Perdana AR (2019) Implementation and analysis of penetration techniques using the man-in-the-middle attack. In: 2019 2nd international conference of computer and informatics engineering (IC2IE). IEEE, pp 188–192
32. Chaudhary S, O'Brien A, Xu S (2020) Automated post-breach penetration testing through reinforcement learning. In: 2020 IEEE conference on communications and network security (CNS). IEEE, pp 1–2
33. Goutam A, Tiwari V (2019) Vulnerability assessment and penetration testing to enhance the security of web application. In: 2019 4th international conference on information systems and computer networks (ISCON). IEEE, pp 601–605
34. Ye Y, Guo J, Xu X, Li Q, Liu H, Di Y (2019) High-risk problem of penetration testing of power grid rainstorm disaster artificial intelligence prediction system and its countermeasures. In: 2019 IEEE 3rd conference on energy internet and energy system integration (EI2). IEEE, pp 2675–2680
35. Jain S, Johari R, Kaur A (2015) PJCT: penetration testing based JAVA code testing tool. In: International conference on computing, communication & automation. IEEE, pp 800–805
36. Singh N, Meherhomji V, Chandavarkar BR (2020) Automated versus manual approach of web application penetration testing. In: 2020 11th international conference on computing, communication and networking technologies (ICCCNT). IEEE, pp 1–6
37. Bechtsoudis A, Sklavos N (2012) Aiming at higher network security through extensive penetration tests. IEEE Latinamerica Trans 10(3):1752–1756
38. Overstreet D, Wimmer H, Haddad RJ (2019) Penetration testing of the Amazon echo digital voice assistant using a denial-of-service attack. In: 2019 SoutheastCon. IEEE, pp 1–6

IoT-Cloud-Enabled Smart Framework for Real-World Applications

Manzoor Ansari and Mansaf Alam

Abstract The tsunami of the Internet of Things (IoT), prevalent deployment, and widespread adoption of smart devices directly or indirectly impact virtually every sector due to their extensive use. To succeed in the current and future IoT and connected device environment, service providers must have scalability, ubiquity, reliability, and performance. With the advent of IoT, a massive amount of data has been generated, growing exponentially every millisecond. This seemingly unlimited amount of data has necessitated a reconsideration of numerous technologies for data collection and processing. This massive amount of data serves as the fuel for various analytics and intelligent systems, such as cloud computing technology. Cloud-IoT, a new paradigm for gathering, storing, and processing data generated by IoT devices, has grown to achieve this attribution. As a crucial component of the IoT, the cloud provides vital exclusive application services across a broad spectrum of the application domain, making it an essential service. This article introduces a smart framework for developing IoT-cloud-enabled real-world applications that use intelligent sensors. The proposed framework enables rapid application development and service delivery using IoT-cloud Integrations. Finally, the article presents a use case scenario for IoT-cloud-enabled air quality monitoring framework, and some challenges hampering IoT-cloud-based adoption are also discussed.

Keywords Internet of things · Cloud computing · Smart framework · IoT-cloud applications · Parameters · Air pollution monitoring

M. Ansari (✉) · M. Alam
Department of Computer Science, Jamia Millia Islamia, New Delhi, India
e-mail: manzoor188469@st.jmi.ac.in

M. Alam
e-mail: malam2@jmi.ac.in

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_8

1 Introduction

The Internet of Things (IoT) is concerned with the configuration, networking, surveillance, and handle of "Internet of Devices" or "Things" which including air quality sensors, thermostats, electrical equipment, transportation, domestic appliances, webcams, and medical equipment which have been previously unconnected to the Internet [1]. The IoT represents a paradigm shift in the possibilities of Internet-connected objects. The IoT is employed in a variety of industries and sectors, including the environment, energy, residential, tourism, logistics, manufacturing, agribusiness, and health care.

According to experts, the IoT will have a latent effect of $11 trillion per year, or 11% of the global economy, by 2025 [2]. By 2035, users are anticipated to adopt one trillion IoT devices [3]. Although technological advancements have made the IoT a reality, this paradigm still has several unresolved issues that need significant study efforts, as highlighted in several recent research [4–7]. However, cloud computing has been hailed as a viable option for the IoT. Cloud computing is a service delivery model for distributing and configuring computer resources (e.g., networking, processing, storage, and applications) [8]. These resources can be promptly supplied and delivered on-demand with minimal administration effort or communication with service providers, with users only paying for the services they use. Additionally, cloud computing promises inexpensive initial investment, high availability, fault tolerance, and virtually unlimited scalability, among other advantages that have enthralled academia and industry [9]. Cloud services are intriguing to IoT because they may be accessed anonymously and ubiquitously over the Internet.

The convergence of IoT with cloud computing can help overcome IoT technological limitations in terms of processing, storage, and energy and improve IoT infrastructures and applications. Due to their restricted energetic capacities, most IoT devices have low processing power and storage, comparable to the several sophisticated processing activities required. In order to circumvent this limitation, such devices may be able to transfer data to be processed and stored effectively on the cloud. Connecting things to the cloud means exploiting IoT-based capabilities and providing them as pay-per-use cloud services, leading to the Cloud of Things (CoT) [10]. As a result of the many advancements in the field of IoT and cloud environment, the IoT-cloud-enabled application has been on the rise in the recent decade. Therefore, development of various IoT-cloud-enabled framework is an emerging field of applications domain.

1.1 Motivation for the Proposed Framework

In recent years, a number of conceptual IoT-enabled cloud frameworks have been designed to address the significance and usability of IoT and cloud convergence.

Many IoT-enabled cloud applications are now moving to the cloud to take advantage of its scalability and flexibility, as well as the ability to pay for what you use.

The aims of this study to develop a comprehensive IoT-cloud-based paradigm that enables real-time monitoring, statistical analysis, and decision-making strategy for IoT data. In this paper, we try to analyze the Internet of Things (IoT) and cloud computing (Cloud) roles in the IoT-cloud paradigm. In Sect. 2, we conduct a comprehensive comparative study of the integration of IoT with the cloud paradigm and also present a new paradigm for cloud computing on IoT data generated by a variety of heterogeneous sources. Section 3 proposes a novel framework for the processing and analysis of IoT data using several cloud computing applications. Furthermore, Sect. 4 demonstrates a case study that defines the role of the IoT-cloud-enabled model in air pollution monitoring. In Sect. 5, we present various IoT-cloud-enabled applications on the cloud that have been validated in real-world applications. Section 6 discusses numerous challenges associated with cloud computing applications to IoT framework, and Finally, Sect. 7 concludes the study.

2 Related Work

Several researchers have adopted various approaches to realize the benefits of current breakthroughs in cloud computing and IoT (Internet of Things) in the current era. Numerous efforts have been made to enable remote monitoring of end-users and continuous collection of rich information to their physical environment using networked sensors embedded in their living environments. We have shown related work in the domains of several IoT-cloud intelligent applications. Additionally, this section attempts to describe the benefits, challenges, and quality parameters of IoT-cloud enabled applications.

With the advent of the IoT-cloud applications in the literature [11], researchers introduce a data gathering approach based on a literature analysis to determine how much cloud computing has impacted the IoT. They have also used e-health as a case study to build and implement an e-health intelligent network infrastructure to prevent data transmission delays and presents a framework for applications that leverage wireless sensors.

In [12], the authors propose a new approach for upgrading intelligent cities that have reduced their energy consumption at every level to secure the IoT's deployment toward a green environment. Additionally, they discuss key developments, resources, risks, and prospective opportunities in these domains.

A significant focus of the literature [13] is on a common strategy to integrate the IoT and cloud computing, which is referred to as the CloudThings architecture. The authors also examine an IoT-enabled smart home scenario to understand the requirements for IoT applications.

In the literature [14], the authors propose a data-centric Internet of Things framework that leverages the Azure public cloud to implement a centralized Internet of Things management service.

This framework was based on their previous work [15], in which they presented a cloud-based remote monitoring and management solution for a plant wall system. To improve human health, the authors of [16] propose an IoT-enabled (ETAPM-AIT) Environmental Toxicology for Air Pollution Monitoring. The ETAPM-AIT model was presented to detect eight pollutants using an IoT sensor array: NH_3, CO, NO_2, CH_4, CO_2, PM2.5, temperature, and humidity. Numerous simulations are run to assess the proposed ETAPM-AIT model's ability to monitor air quality over time. The findings indicate that the ETAPM-AIT model proposed here outperforms other existing techniques.

A framework (Héctor) in [17], the authors have provided automated testing of IoT applications that enables the execution of user-defined studies on agnostic IoT testbeds. The proposed framework has generated virtual testbeds with adaptive networks for testing IoT applications regardless of device availability. They have shown the framework's functional feasibility for IoT applications, but still, the performance effect of emulation indicates the importance of physical isolation between edge device simulation and application data sources in sensor network IoT applications.

Table 1 summarizes the smart IoT-cloud Framework to other related work based on several classifications such as cloud computing (CC), Internet of Things (IoT), and big data analytics (BDA). Additionally, the study evaluated the techniques using more criteria such as real-time aspect (RTA), energy efficiency (EE), privacy and security (PS), efficient resource provision (ERP), context-aware sensing capability (CASC), cost-effectiveness (CE), AI-based technique (AI), scalability (SC), Data Quality (DQ), availability (AV), adaptability (AD), interoperability (IO), and decision-making model (DMM). The study analyzes numerous works based on their values: *Y* indicates includes that parameter, whereas *N* excludes that parameter and—indicates not considered.

3 Proposed IoT-Cloud-Enabled Smart Framework for Real-World Applications

In Fig. 1, the proposed framework is designed on the cloud and IoT, and it may be used in a variety of real-world applications. The framework makes use of physical sensors embedded in real-world environments, as well as Web services, to allow end-users to monitor IoT-cloud applications. The cloud platform governs the installed devices depending on the responses received from real-time Web services. The decisions are made following certain imposed constraints, which will vary depending on the application being deployed. Service providers deploy the end-user application interface in a cloud platform to deliver and monitor the end-user application. The proposed framework consists of four phases which are discussed as:

Table 1 Comparative analysis for IoT-cloud-enabled framework

Parameters																	
S. No.	Refs.	Key contributions	IoT	CC	EE	PS	ERP	CASC	CE	RTA	AI	DQ	AV	AD	BDA	IO	DMM
1	Majhi et al. [18]	Forecasting earthquake magnitude with a neural network and optimization technique	N	N	–	–	–	–	–	Y	Y	–	–	–	–	–	Y
2	Pirmagomedov et al. [19]	Predicting earthquakes with animal behavior	Y	N	–	–	–	–	–	Y	Y	–	–	–	–	–	–
3	Sood et al. [20]	A cloud-IoT architecture based on social cooperation is presented to monitor flood-related Big Data and forecast flood conditions for a given location	Y	Y	–	–	–	–	–	Y	–	–	–	–	–	–	–
4	Kaur et al. [21]	Flood forecasting using IoT-cloud energy-efficient cloud framework	Y	Y	Y	–	–	–	–	Y	Y	Y	–	–	Y	–	–
5	P’erez et al. [22]	The IoT-acquired mobility data is processed at a different computational level: fog and cloud to predict traffic flow	Y	Y	–	–	–	–	–	Y	Y	–	–	–	–	–	–

(continued)

Table 1 (continued)

Parameters																	
S. No.	Refs.	Key contributions	IoT	CC	EE	PS	ERP	CASC	CE	RTA	AI	DQ	AV	AD	BDA	IO	DMM
6	Wan et al. [23]	A computationally load balancing solution for edge nodes is suggested to allow efficient edge computing in the Internet of Vehicles with 5G networks without significantly degrading	Y	Y	–	–	–	–	–	Y	–	–	–	–	–	–	–
7	Zhang et al. [24]	The authors have described a deep neural network-based learning model for determining the velocity and method of transit of moving objects	Y	Y	–	–	–	–	–	Y	–	–	–	–	–	–	–
8	Liu et al. [25]	Suggested to balance the compute burden on the edge nodes in a vehicular network based on the Internet of Things	Y	Y	–	–	–	–	–	Y	–	–	–	–	–	–	–
9	Chauhan et al. [26]	Used big data analytics for HIV/TB patients to detect patterns in socioeconomic aspects	N	N	–	Y	–	–	–	N	–	–	–	–	Y	–	Y

(continued)

Table 1 (continued)

Parameters																	
S. No.	Refs.	Key contributions	IoT	CC	EE	PS	ERP	CASC	CE	RTA	AI	DQ	AV	AD	BDA	IO	DMM
10	Junior et al. [27]	Standardizes the publish-subscribe topic and payload with context-aware messages at both ends (device and cloud platform)	Y	Y	N	Y	–	Y	–	Y	N	–	–	–	–	–	Y
11	Kaginalkar et al. [28]	IoT, sensor, cloud computing, and AI/ML modal proposed to improve air quality	Y	Y	–	N	–	–	–	Y	Y	–	–	–	Y	Y	Y
12	Abdelouahid [29]	Suggested an IoT interoperability architecture with smart building applications	Y	Y	–	–	–	–	–	–	–	–	–	–	–	Y	–

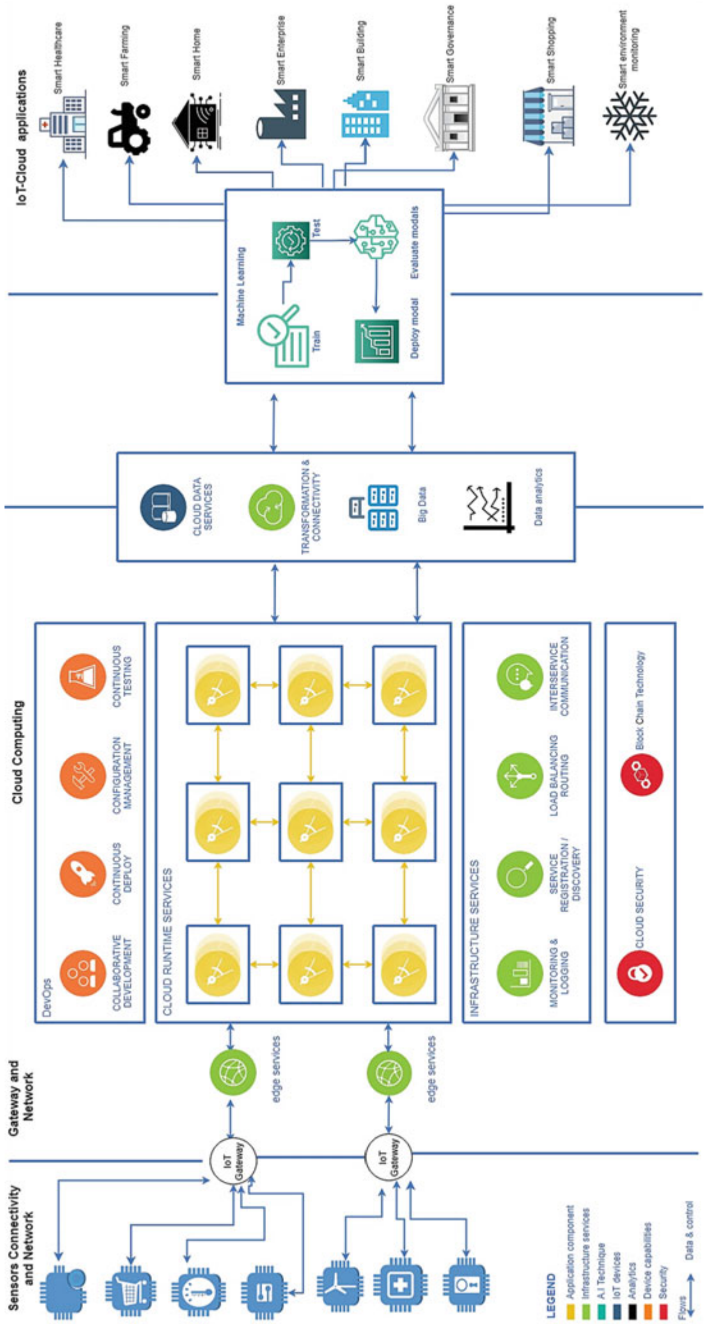


Fig. 1 IoT-cloud-enabled smart framework for real-world applications

3.1 Sensor Connectivity and Network Phase

Various Sensors and actuators are deployed in real-time IoT-cloud applications, which requires the development of the framework. This phase comprises heterogeneous IoT devices equipped with sensors and actuators to collect and control real-time IoT data from their physical environment. Sensors with unique identification can be remotely operated and located at the physical layer. There are many different aspects to these technologies. They include humidity sensors, smartwatches, intelligent cameras, smart air quality sensors, embedded systems, and other intelligent gadgets.

With IoT sensors, the connection can be established between the physical and digital worlds, with a single IoT system containing many sensors and actuators. Based on the information provided by the sensors, the actuators carry out real-world operations such as monitoring the weather and measuring moisture fluctuations. The main goal of this phase is to detect and gather IoT data before passing it on to the following phase, which is the gateway and network phase.

3.2 Gateway and Network Phase

The second phase, the Gateway and Network, aims to provide data routing and transmission functionalities. Thus, the primary purpose of this phase is to transport data across diverse networks without losing information effectively. This phase is used by Internet gateways, switches, and routers. Computing-intensive activities are strenuous in IoT because of resource constraints. Thus, IoT-generated data are sent to cloud data centers for processing.

The network layer connects network infrastructure, such as gateways, switches, routers, and other wired and wireless devices. This phase manages end-to-end communication and transmits messages between objects and systems. The gateway translates between several network communication protocols, including IoT, to gateway communication via ZigBee, Z-wave, and Bluetooth low energy.

3.3 Cloud Computing Phase

Cloud computing manages and collects a large volume of IoT data for training and testing purposes in this phase. The data obtained from various sensors have various data frequencies and formats. Additionally, the data collection frequency and methodology used by IoT sensors vary. As a result, it is critical to integrate data into a suitable format and pre-process it for machine learning. The cloud platform leverages the processed data to build several learning models that consider the capabilities of the different edges and the peculiarities of the individual IoT data. The cloud platform then selects the optimal model for the current state of the individual task and transmits it to the Edge platform together with the model's application statistics.

3.4 IoT-Cloud Applications Phase

The next phase is the IoT-cloud application, which receives IoT data from the preliminary phase to develop various services and applications. This phase often comprises the user interface, methods for IoT data models, decision-making modal, business rules, and other components required for the individual IoT service or application. It provides end-user with a customized interface to their own needs when interacting with IoT-cloud applications.

It facilitates the detection, management, and delivery of IoT devices. It collects data from all levels and presents it to the requesting applications in an improved version. The primary advantage of Restful Web services is that it enables seamless interaction between applications written in various technologies. Because the IoT can simulate many applications, there is a significant need for REST APIs. The REST-based design generates APIs, source code, and reusable modules used in IoT systems.

4 Use Case Scenario (Air Quality Monitoring)

The proposed framework is discussed in a more detailed manner to refine the application framework (air quality monitoring) presented in Fig. 4. An air quality monitoring system providing high spatial resolution real-time information is one of the essential mechanisms for developing management strategies for air quality improvement [30].

In the last few decades, air quality monitoring has been well-established and well-known concept. Various sensing devices have been investigated to monitor the impact of our living environment. They included a CO sensor, a CO_2 sensor, SDS011 sensors (which can detect PM2.5 and PM10), a temperature sensor, a VOC sensor, a humidity sensor, and other sensors essential for collecting data for air quality analysis. The data collected from these sensors are then analyzed and integrated to provide advanced analytics about the air quality in our environment at a high spatio-temporal resolution. This method has critical components of the IoT devices that gather data, cloud computing infrastructure that supports data analysis, and end-user applications (IoT-cloud-enabled air quality monitoring). The data flow diagrams for the Wireless network and IoT Sensor nodes and the cloud portal processes are depicted in Figs. 2 and 3 [31].

The device is integrated with various sensors that enable it to monitor a series of environmental parameters, including location and time. The device has a Navigation system and Universal Mobile Telecommunications Service (UMTS) modules for

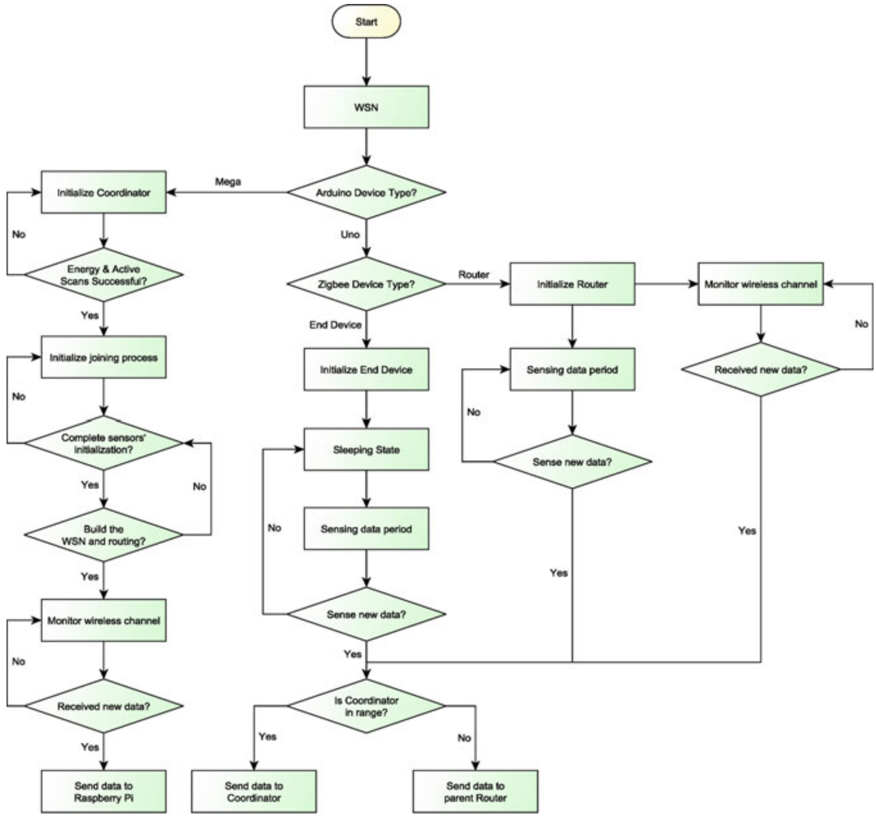


Fig. 2 Process flow on IoT sensors-cloud-wireless networks

tracking position and real-time communication with a cloud-based server, respectively. Such devices transmit georeferenced data on the air quality to a central server. Due to the proposed framework modular design, including more sensors as needed is possible. According to the IoT-cloud-based air quality monitoring framework depicted in Fig. 4, a central server is configured to efficiently analyze the data collected by the various air quality sensors and IoT devices, which is then transmitted to cloud platforms via network gateways and Wi-Fi, where it is stored and further analyzed at the data decision mechanism. To generate meaningful results, the massive volume of unstructured data must be painstakingly evaluated. However, the raw data from the sensors frequently contain errors such as missing values, outliers, and other types of erroneous measurements. Thus, cleaning and processing IoT data are essential before initiating any analysis. After analysis, the obtained results from the data decision modal are transmitted wirelessly to the IoT-cloud application phase. An IoT-cloud-enabled air quality monitoring framework is suggested to efficiently

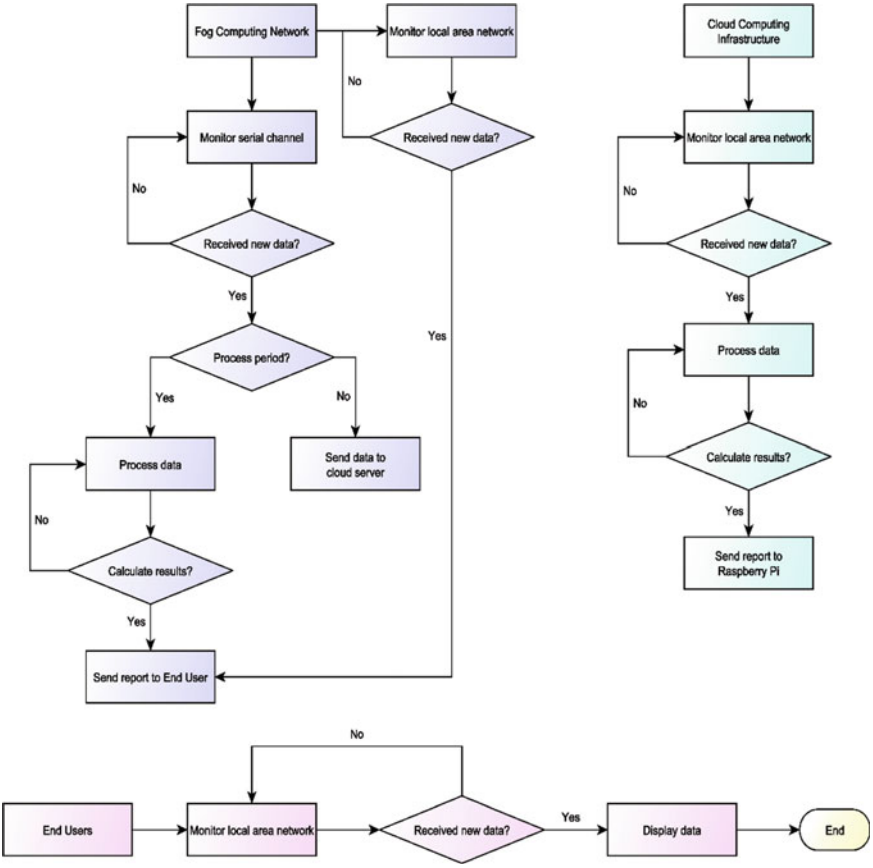


Fig. 3 Flowchart of processes on IoT Cloud

monitor air quality data and real-time access. As a result, air quality has been monitored via the webserver and through the application, connected to associated intelligent devices. Air quality monitoring is now robust and efficient mainly due to the smart framework, which allows access from anywhere using smart devices. Thus, the smart framework has been designed to be very similar to deploying the webserver using various existing IoT-cloud platforms (such as google cloud, amazon, azure).

5 Various Real-World IoT-Cloud Applications

In recent years, we have witnessed numerous IoT-cloud-enabled applications which have been proposed in Fig. 5, the application scenario facilitated by the convergence of cloud and IoT.

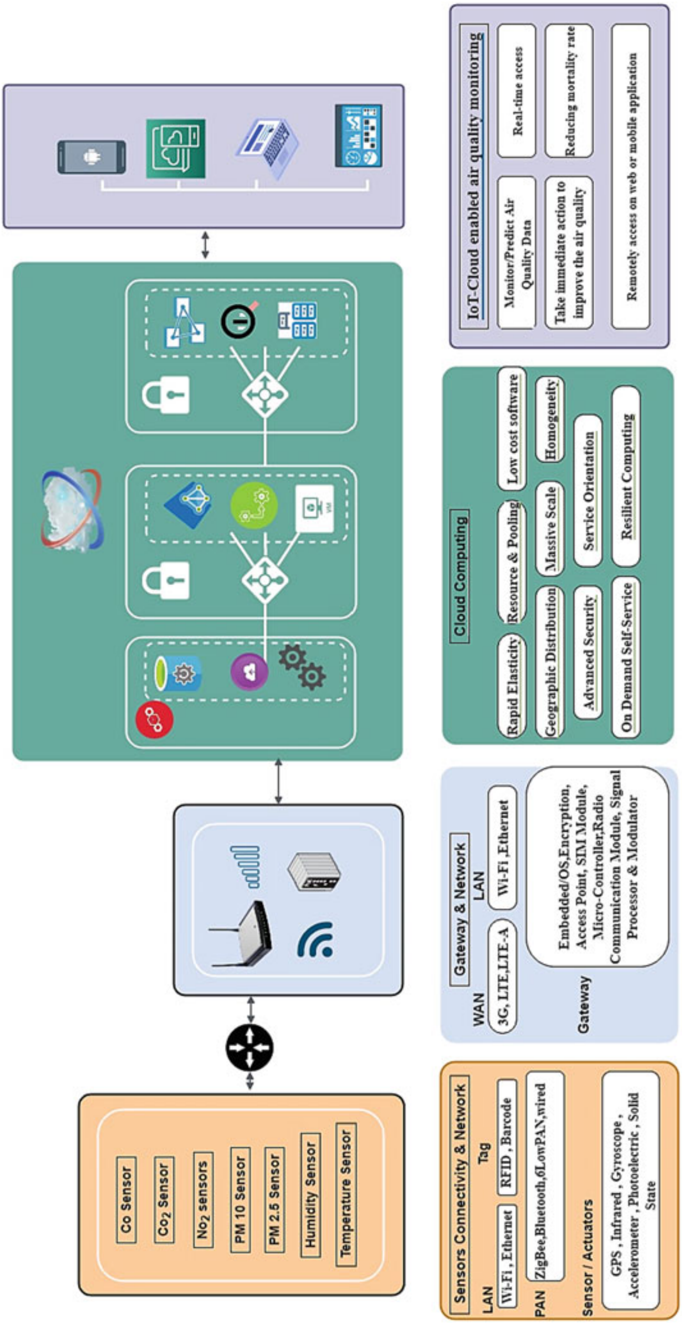


Fig. 4 IoT-cloud-enabled air quality monitoring framework (a use case scenario)

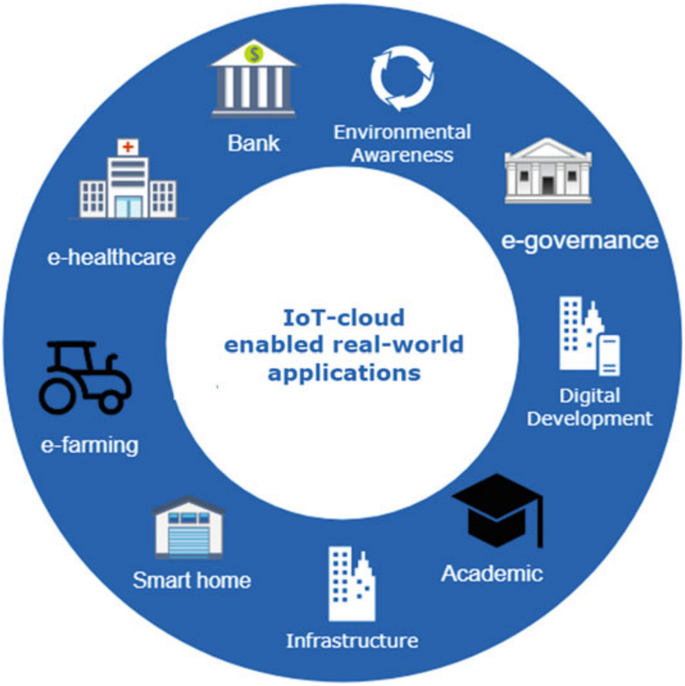


Fig. 5 IoT-cloud-enabled real-world applications

5.1 Smart Health Care

In the last few years, technological advances such as the IoT with cloud computing have solved many real-world problems. Innovative health care is one of them. Numerous cloud services, intelligent gadgets, and mobile Internet continuously enable cost-effective, well-organized, efficient, and excellent ubiquitous medical services. Cloud computing in health care assists in the globalization of technical aspects, minimizing the need for expertise, and governing technology [32]. To develop an extensive healthcare framework, electronic and medical fields require demonstrative and autonomous gadgets and sensors capable of associating with the cloud to deliver personalized administration and contentment.

5.2 Smart Agriculture

Climate-Smart Agriculture (CSA) is a strategy for reshaping agricultural development in the context of climate change realities. Over the years, there have been numerous changes in how people perform agricultural work. The new-age farmer has evolved beyond considering agriculture to thinking creatively about farming. Researchers in this domain have developed theoretical frameworks that employ smart devices to investigate the aspects that influenced plant growth and then carry out agricultural activities based on their findings [33].

5.3 Smart Home

With a diverse variety of embedded systems, the Internet of Things (IoT) has a variety of applications in the home environment, including the automation of mutually multi occurrences. Numerous devices such as lights, domestic appliances, smartphones, and security cameras are linked to the Internet in an IoT-cloud-enabled Smart Home environment, allowing users to monitor and control equipment regardless of their time or geographical constraints. The Web-enabled, standardized, and uniform identity of residential equipment is also essential [34].

5.4 Smart Environment Monitoring

Smart Environmental monitoring is a technique of collecting data on meteorological conditions, endangered species protection, air quality monitoring, and various other characteristics related to our environment. Numerous sensors and other observational devices are integrated into applications to monitor environmental parameter effects periodically.

Among the most significant contributions to this domain are the following: Low-cost sensor technologies and wireless communication systems such as the Internet of Things (IoT) enable the use of arrayed sensing devices to monitor air pollution in real time at several locations. It measures NO_2 and CO concentrations using low-cost electrochemical sensors and particulate matter using an infrared sensor (PM). Solar energy or the main power supply may be used to power the node. It supports long-range low-power communication via public or private Lo-RaWAN IoT networks, as well as short-range high-data-rate communication over Wi-Fi [35].

6 Related Challenges of IoT-Cloud-Enabled Applications

IoT-cloud-enabled applications have traditionally been explored in terms of how integrating IoT with the cloud offers many benefits and facilitates the development of such applications. However, these applications present several challenges for sophisticated cloud environments, including reliability, heterogeneity, scalability, security and privacy concerns, and resource allocation.

6.1 Privacy and Security

Nowadays, ensuring the security and privacy of IoT-cloud-enabled applications has become a key concern in the digital era. The reasons for it include occasional compromise of consumer-generated IoT data by cybercriminals or third-party users and the mutation of critical sensor-based data due to the advent of malware threats. The answer to such issues is to limit access to vital and sensitive data while enforcing rigorous privacy protection measures [36].

6.2 Reliability

IoT-cloud-enabled systems provide vital decision-making in medical, home automation and societies, and smart traffic control by using devices and sensor data. This system must communicate in a seamless, completely adaptable, and persistent manner. However, specific IoT devices have limited resources, ensuring dependability becomes critical. To develop and implement a reliable IoT-cloud framework, it is required to identify and resolve variances in the application or device, which increases the system's robustness [37].

6.3 Scalability

IoT-cloud applications and various services are often insufficiently trained to operate optimally in big and small contexts concurrently, resulting in insufficient scalability between the user and the devices. To facilitate scaling, the IoT-cloud framework must be flexible in both small and large contexts, capable of managing a diverse set of data storage services, and self-contained [38].

6.4 Performance

Any Internet of Things (IoT) applications robustness, fault-tolerance, time-criticality, interoperability, scalability, and self-organization are all dependent on the system's ability to self-organize, which is a feature of self-organizing systems. The greater the number of these factors, the better the performance will be by default, resulting in cost optimization in the long term [39].

6.5 Heterogeneity

Multiple heterogeneous devices should be integrated into sophisticated IoT-based cloud applications to enhance the system's performance. Computational expenses and energy consumption also affect the performance of large-scale environments, and high-performance computing equipment is necessary to operate successfully in large-scale environments [40].

6.6 Resource Allocation

With the growth of the Internet of Things (IoT) devices, cloud data centers have played a critical role in enabling IoT data analysis and data-driven IoT applications. Cloud resources in virtual computers are required to provide data-driven services. Simultaneously, IoT data are often sent between service components. Thus, to optimize the response time of IoT services, it is necessary to address network issues and eliminate network congestion during resource allocation [41, 42].

7 Conclusions and Future Directions

In recent years, academic institutions and industry organizations have expressed an interest in Internet of Things (IoT) technologies. It has now become an essential part of our daily life. It can connect almost everything in our world to everything else. In addition to being difficult to design, IoT systems have limited storage and retrieval capacities. Integrating cloud computing and IoT can bring significant benefits to many IoT applications. In this paper, we design an intelligent framework that incorporates various real-world applications based on IoT and cloud computing technologies. This study proposes a novel framework for integrating cloud-level IoT services on a smart device using the concept of Intention. According to our proposed framework, the purpose of a user or device is to take a proactive role between connected devices in cloud-scale IoT networks. Additionally, we present an IoT-cloud-enabled air quality monitoring smart framework as a case study that simultaneously incorporates real-time assessments and controls of environmental air quality by analyzing IoT data. Future research directions will focus on implementing and validating the proposed system interoperability.

References

1. Raj P, Raman AC (2017) The internet of things: enabling technologies, platforms, and use cases. Auerbach Publications
2. Manyika J, Chui M, Bisson P, Woetzel J, Dobbs R, Bughin J, Aharon D (2015) Unlocking the potential of the internet of things. McKinsey Global Institute, p 1
3. Sparks P (2017) The route to a trillion devices. White Paper, ARM
4. Atzori L, Iera A, Morabito G (2010) The internet of things: a survey. Computer networks 54(15):2787–2805
5. Shi Y, Ding G, Wang H, Roman HE, Lu S (2015) The fog computing service for healthcare. In: 2015 2nd international symposium on future information and communication technologies for ubiquitous healthcare (Ubi-HealthTech). IEEE, pp 1–5
6. Cisco (2018) Fog computing and the internet of things: extend the cloud to where the things are [White paper]. Accessed on 21 Feb 2018
7. Dastjerdi AV, Gupta H, Calheiros RN, Ghosh SK, Buyya R (2016) Fog computing: principles, architectures, and applications. In: Internet of things. Morgan Kaufmann, pp 61–75
8. Mell P, Grance T (2011) The NIST definition of cloud computing
9. Zhou M, Zhang R, Zeng D, Qian W (2010) Services in the cloud computing era: a survey. In: 2010 4th international universal communication symposium. IEEE, pp 40–46
10. Distefano S, Merlino G, Puliafito A (2012) Enabling the cloud of things. In: 2012 sixth international conference on innovative mobile and internet services in ubiquitous computing. IEEE, pp 858–863
11. Hussein WN, Hussain HN, Humod IM (2022) A proposed framework for healthcare based on cloud computing and IoT applications. Mater Today Proc 60:1835–1839
12. Abiodun MK, Adeniyi EA, Awotunde JB, Bhoi AK, AbdulRaheem M, Oladipo ID (2022) A framework for the actualization of green cloud-based design for smart cities. In: IoT and IoE driven smart cities. Springer, Cham, pp 163–182
13. Zhou J, Leppanen T, Harjula E, Ylianttila M, Ojala T, Yu C et al (2013) Cloudthings: a common architecture for integrating the internet of things with cloud computing. In: Proceedings of the 2013 IEEE 17th international conference on computer supported cooperative work in design (CSCWD). IEEE, pp 651–657
14. Liu Y, Hassan KA, Karlsson M, Pang Z, Gong S (2019) A data-centric internet of things framework based on azure cloud. IEEE Access 7:53839–53858
15. Liu Y, Hassan KA, Karlsson M, Weister O, Gong S (2018) Active plant wall for green indoor climate based on cloud and internet of things. IEEE Access 6:33631–33644
16. Asha P, Natrayan L, Geetha BT, Beulah JR, Sumathi R, Varalakshmi G, Neelakandan S (2021) IoT enabled environmental toxicology for air pollution monitoring using AI techniques. Environ Res 112574
17. Behnke I, Thamsen L, Kao O (2019) HéCtor: a framework for testing IoT applications across heterogeneous edge and cloud testbeds. In: Proceedings of the 12th IEEE/ACM international conference on utility and cloud computing companion, pp 15–20
18. Majhi SK, Hossain SS, Padhi T (2020) MFOFLANN: moth flame optimized functional link artificial neural network for prediction of earthquake magnitude. Evol Syst 11(1):45–63
19. Pirmagomedov R, Blinnikov M, Amelyanovich A, Glushakov R, Loskutov S, Koucheryavy A et al (2018) IoT based earthquake prediction technology. In: Internet of things, smart spaces, and next generation networks and systems. Springer, Cham, pp 535–546
20. Sood SK, Sandhu R, Singla K, Chang V (2018) IoT, big data and HPC based smart flood management framework. Sustain Comput Inf Syst 20:102–117
21. Kaur M, Kaur PD, Sood SK (2021) Energy efficient IoT based cloud framework for early flood prediction. Nat Hazards 109(3):2053–2076
22. Pérez JL, Gutierrez-Torre A, Berral JL, Carrera D (2018) A resilient and distributed near real-time traffic forecasting application for fog computing environments. Futur Gener Comput Syst 87:198–212

23. Wan S, Li X, Xue Y, Lin W, Xu X (2020) Efficient computation offloading for internet of vehicles in edge computing-assisted 5G networks. J Supercomput 76(4):2518–2547
24. Zhang R, Xie P, Wang C, Liu G, Wan S (2019) Classifying transportation mode and speed from trajectory data via deep multi-scale learning. Comput Netw 162:106861
25. Liu J, Wang W, Li D, Wan S, Liu H (2019) Role of gifts in decision making: an endowment effect incentive mechanism for offloading in the IoV. IEEE Internet Things J 6(4):6933–6951
26. Chauhan R, Kaur H, Chang V (2021) An optimized integrated framework of big data analytics managing security and privacy in healthcare data. Wirel Pers Commun 117(1)
27. Junior NF, Silva AA, Guelfi AE, Kofuji ST (2021) Privacy-preserving cloud-connected IoT data using context-aware and end-to-end secure messages. Procedia Comput Sci 191:25–32
28. Kaginalkar A, Kumar S, Gargava P, Niyogi D (2021) Review of urban computing in air quality management as smart city service: an integrated IoT, AI, and cloud technology perspective. Urban Clim 39:100972
29. Abdelouahid RA, Debauche O, Marzak A (2021) Internet of things: a new Interoperable IoT platform. Application to a smart building. Procedia Comput Sci 191:511–517
30. Ansari M, Alam M (2022) IoT-cloud enabled statistical analysis and visualization of air pollution data in India. In: Proceedings of data analytics and management. Springer, Singapore, pp 125–139
31. Tsipis A, Papamichail A, Angelis I, Koufoudakis G, Tsoumanis G, Oikonomou K (2020) An alertness-adjustable cloud/fog IoT solution for timely environmental monitoring based on wildfire risk forecasting. Energies 13(14):3693
32. Anuradha M, Jayasankar T, Prakash NB, Sikkandar MY, Hemalakshmi GR, Bharatiraja C, Britto ASF (2021) IoT enabled cancer prediction system to enhance the authentication and security using cloud computing. Microprocess Microsyst 80:103301
33. Sinha BB, Dhanalakshmi R (2022) Recent advancements and challenges of internet of things in smart agriculture: a survey. Future Gener Comput Syst 126:169–184
34. Chen SY, Lai CF, Huang YM, Jeng YL (2013) Intelligent home-appliance recognition over IoT cloud network. In: 2013 9th international wireless communications and mobile computing conference (IWCMC). IEEE, pp 639–643
35. Ali S, Glass T, Parr B, Potgieter J, Alam F (2020) Low-cost sensor with IoT LoRaWAN connectivity and machine learning-based calibration for air pollution monitoring. IEEE Trans Instrum Meas 70:1–11
36. Al-Turjman F, Zahmatkesh H, Shahroze R (2019) An overview of security and privacy in smart cities' IoT communications. Trans Emerg Telecommun Technol e3677
37. Moore SJ, Nugent CD, Zhang S, Cleland I (2020) IoT reliability: a review leading to 5 key research directions. CCF Trans Pervasive Comput Interact 1–17
38. Gupta A, Christie R, Manjula PR (2017) Scalability in internet of things: features, techniques and research challenges. Int J Comput Intell Res 13(7):1617–1627
39. Mutlag AA, Khanapi Abd Ghani M, Mohammed MA, Maashi MS, Mohd O, Mostafa SA et al (2020) MAFC: multi-agent fog computing model for healthcare critical tasks management. Sensors 20(7):1853
40. Mostafa SA, Gunasekaran SS, Mustapha A, Mohammed MA, Abduallah WM (2019) Modelling an adjustable autonomous multi-agent internet of things system for elderly smart home. In: International conference on applied human factors and ergonomics. Springer, Cham, pp 301–311
41. Ali SA, Ansari M, Alam M (2020) Resource management techniques for cloud-based IoT environment. In: Internet of things (IoT). Springer, Cham, pp 63–87
42. Ali SA, Alam M (2018) Resource-Aware Min-Min (RAMM) algorithm for resource allocation in cloud computing environment. arXiv preprint arXiv:1803.00045

Empirical Analysis of the Impact of Homomorphic Encryption on Cloud Computing

Chinmayee Rout, Srinivas Sethi, Ramesh Kumar Sahoo, and J. Chandrakanta Badajena

Abstract Cloud computing has been of greater convenience in recent years owing to its flexibility and on-demand service availability. Cloud computing services allow multiple users to access the available resources concurrently. So, trust is a key parameter from every client's point of view. While accessing the resources in the cloud security becomes the primary concern for every participant and cryptography provides a secure room for every client. The cryptographic solution can be achieved in two different ways such as the conventional mechanism and the homomorphic encryption (HE) mechanism. The main objective of these two solutions is to maintain the confidentiality, integrity, and availability (CIA) of the information and resources stored in the cloud storage. The conventional mechanism provides a secure means to encrypt and decrypt the information at the sender and receiver sides respectively. But for any kind of operation, the information needs to be decrypted which increases the computational overhead which shows the path to the HE. In HE, the user can directly perform any kind of operation on the encrypted data which can help to reduce the computational time and cost to a much more significant value. In this current research work, the main focus is on the HE. Several HE mechanisms such as the RSA, Elgamal, Paillier, and DGHV have been implemented and the performance has been analyzed in terms of computational time. The empirical analysis demonstrates that the RSA homomorphic algorithm shows an average encryption and decryption time of 138.5 and 206.25 s which is lower as compared to others. Considering average CPU utilization, RSA shows around 27% with an average of 30 MB memory utilization which is found lower as compared to other algorithms.

Keywords Scalability · Availability · Homomorphic encryption · Cloud computing

C. Rout (✉)
Utkal University, Bhubaneswar, Odisha, India
e-mail: rchinmayee1985@gmail.com

S. Sethi · R. K. Sahoo
Indira Gandhi Institute of Technology, Sarang, Dhenkanal, India

J. Chandrakanta Badajena
Odisha University of Technology and Research, Bhubaneswar, India

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_9

1 Introduction

Cloud computing is a trending technique that has rapidly growing processing patterns in recent years, as seen by the several widely used cloud administrations, such as AWS and EC2 [1]. In cloud computing system service provider allows different virtual assets such as storage, processor, and platform to the client for performing numerous tasks by using those cloud resources. Clients can access these assets on a pay-as-you-go basis via a web-based interface. Cloud computing is attracting the attention of a variety of users these days since it provides extensible and on-demand access to resources [2].

Like a utility over an electric power network, cloud computing relies on asset sharing to achieve intelligibility and economies of scale. The storage gives processing administrations dependent on three central cloud computing models: Saas, PaaS, and IaaS [2]. The five highlights of cloud computing are:

- Scalable administration
- Flexibility
- Location independence
- Self service
- On-demand administration.

As cloud computing becomes more extensively accepted, users, are increasingly dependent on cloud systems for storing sensitive and private information. With the increase in the number of users, ensuring overall data integrity and confidentiality poses a challenge that needs to be addressed properly. Encryption has somehow been able to provide a quite essential solution in this scenario. On the other hand, encryption at a large scale demands added computational cost and complexity. In a cloud deployment, encryption techniques fall short of implementation goals as the deployments are done over third-party services that are being run on third-party servers [3]. It is difficult to privatize the stored data from the service provider while allowing them to execute a variety of operations on it. Homomorphic encryption technology enables the management of ciphertext data while maintaining anonymity. It can instantly manipulate the ciphertext stored in the cloud and send it to clients as ciphertext [4].

Cloud computing security in contrast to cryptography, has a key goal is to keep the CIA property of information security intact. Confidentiality, integrity, and availability are all hallmarks of the CIA. Confidentiality focuses on the encryption and decryption processes to ensure that the information stored or exchanged by the client is only accessible to the intended receiver. The integrity focuses on keeping the data's uniqueness and aims to avoid MITM attacks by preventing any form of alteration between the communication channels [5]. The confidentiality and integrity processes are critical to the availability process. Because this method deals with the availability of proper information to the client, if both are maintained, the availability will be present automatically. There are two alternative techniques for achieving the aforesaid cloud security aspect: classical cryptography and homomorphic encryption.

- Classical Cryptography: The encryption and decryption processes are two different approaches provided by this system. To prevent unwanted access, the information to be saved in the cloud storage must be encrypted. Any operation on the stored data necessitates the decryption process because the operation is not possible in the encrypted format, which becomes the fundamental flaw in the traditional conventional approach. When the size of the data saved is large, then the computational overhead of decrypting the data grows.
- Homomorphic Encryption: The lacuna present behind the conventional mechanism provides room for homomorphic encryption (HE) where the decryption process is strictly avoided for doing any kind of operation on the encrypted data resulting in a decrease in computational overhead.

Our research aims at identifying the potential limitations of conventional cryptography methods in cloud computing and the benefits of using the HE techniques. Our work proceeds in two phases. The first phase is to implement the four HE algorithms such as RSA, Elgamal, Paillier, and DGHV and the second phase is to measure the performance of the above-said methods in terms of computational time, CPU utilization, and Memory utilization.

The following is the outline of the paper. Section 2 is linked to the related work. The methods employed during the research activity will be discussed in Sect. 3. Sections 4 and 5 hold the result with statistical analysis and conclusion, respectively.

2 Prioritise Risks Insights to Uncover Hidden Threats

In this section, several research works have been analyzed based on the statistical analysis along with the key concept considered during the work.

Zhang et al. in [3] had proposed an identity-based homomorphic system to avoid malicious activity during the storage and different kind of operation. The key concept present behind this development is securing the key pair while sharing data between the client and server since the data security depends upon the key pairs needed or encryption and decryption process. Elhabob et al. in [4] had proposed a BGV based homomorphic encryption system to secure the public key during the data-sharing session and the proposed system was intended for the internet of the vehicle system. Deng et al. [5] focused on flexible data sharing over a public cloud to avoid the MITM attack. The key concept present in the proposed work is to strengthen the authorization part.

Yang et al. [6] had focused on developing an efficient cloud storage system based on Ring Learning with Error (RLWE) method to avoid the DoS and DDoS attack during data sharing. Qui et al. in [7] had proposed a framework based on homomorphic encryption methods for preserving data during the mining process in a public cloud to avoid various cloud-based attacks. Liao et al. in [8] had developed a cost-effective decryption and encryption process for green cloud computing to provide high-speed data storage and access. The main objective was to reduce the computational overhead. Ahuja and Mohanty in [9] had developed an attribute-based encryption method based on homomorphic encryption BGV and DGHV methods to achieve flexible access rights and fine-grained access control. The main focus is on the key pair and certificate exchange mechanism between the client and server.

De and Ruj [10] had proposed an encryption mechanism based on the KP-ABE mechanism to decrease the computational time. The key focus was on reducing the time required to encrypt and decrypt the data over a public cloud. Min Zhao and Geng [11] had implemented various homomorphic encryption algorithms such as LWE, RLWE, RSA, BGV, and BFV to identify the effect of the homomorphic encryption process on cloud computing. Tran et al. [12] had developed a system based on homomorphic encryption for handling the user feedback and the feedback will be shared by using the key pair. Liu et al. [13] had proposed a framework based on RSA and decision tree training (DTT) to bridge the gap present in between the information privacy and efficiency of cloud storage while dealing with a public cloud. Brakerski [14] had implemented various full homomorphic and partially homomorphic encryption methods to analyze the homomorphic features used for cloud security.

In cloud computing storage while storing or accessing the stored data from the server the data loss can be there and Zheng et al. [15] had proposed a system based on homomorphic encryption to avoid the data loss and the impact of the distortion factor on the stored data had been severely decreased. Joshi [16] had proposed a secure framework based on Paillier HE to secure the data on the cloud. Asuncion et al. in [17] and Yan et al. in [18] had implemented the RSA and Paillier HE for the facial image and video data encryption, manipulation over a public cloud, respectively. The analysis shows that the homomorphic mechanism is also effective on image and video data along with the numeric data.

The author in [19] proposed a privacy-preserving platform to deal with the big data in the cloud. Fully homomorphic encryption (FHE) is a new and sophisticated cryptosystem that allows you to analyze encrypted data. The proposed system includes BGV and Fuzzy C-means clustering (FCMC) technique and the system is scalable in terms of both data and computation.

In [20] Rahman et al. suggested an AI-based edge service composition model for edge networks. Additionally, utilizing the BGV fully homomorphic encryption (FHE) technique, a privacy-preserving AI service composition framework has been provided for performing composition on encrypted QoS data. Using a synthetic QoS dataset, many experiments were conducted to assess the performance of the proposed framework.

With the goal of designing and developing security mechanisms for cloud computing paradigm utilizing Paillier HE, Bijeta et al. introduced a unique framework in [21] to offer a representation of a particular architectural pattern for delivering security for different cloud providers. In contrast to current methods such as RSA and Elgamal, the proposed system is compared.

In [22], Alexandru et al. created a cloud-based protocol for a restricted quadratic optimization problem comprising multiple participants, each of them holds private data. The system uses DGK partially homomorphic encryption to secure the communication which is based on projected gradient ascent on the Lagrange dual issue. The approach is proved to provide computational privacy using formal cryptographic notions of indistinguishability.

Mert et al. proposed two hardware designs with high-performance polynomial multipliers for expediting the encryption and decryption processes of the Brakerski/Fan-Vercauteren (BFV) homomorphic encryption system in [23]. The presented model was applied using 1024-degree polynomials with 8-bit and 32-bit coefficients for plaintext and ciphertext, respectively. As a proof of concept, when compared to pure software solutions, the proposed framework delivers about $12\times$ and $7\times$ latency speedups for offloaded encryption and decryption processes, respectively, including I/O activities.

For the cloud computing platform, Park et al. [24] introduced a privacy-preserving reinforcement learning (PPRL) system. For RSA fully homomorphic encryption, the proposed framework uses a cryptosystem based on learning with errors (LWE). The suggested PPRL framework's performance is analyzed and evaluated in terms of communication and computational overhead.

Private Fast Text (PrivFT) was proposed by Al Badawi et al. in [25] as an efficient and non-interactive approach for text classification while maintaining content privacy utilizing Cheon-Kim-Kim-Song (CKKS) Fully Homomorphic Encryption (FHE). Four distinct types of cloud datasets, including Yelp, AG, IMDB, and DBPedia, were used to test the suggested architecture. The results were compared to those of other models such as XLNet, BERT ITPT, and ULMFit.

Wu et al. presented a novel secure and efficient outsourced k-means clustering (SEOKC) approach in [26], which uses completely homomorphic encryption with ciphertext packing to perform parallel processing without incurring additional costs. The suggested system protects privacy in three different ways such as database security, data access patterns, the privacy of the clustering results.

In [27], the author seeks to provide a noise-free, efficient, and symmetrically verifiable FHE based on a novel mathematic framework using Elgamal. The noise in an encryption method is continuous and independent of the homomorphic evaluation of ciphertexts. Simple matrix operations like multiplication and addition are used to get the homomorphy of the proposed scheme. In a cloud setting, the multiplication process of our encryption system takes just a few moments to complete.

In [28], the author focuses on storing data in an encrypted manner on the cloud using fully homomorphic encryption such as RSA, Paillier, and BGV. The data is kept in Amazon Web Service's (AWS) public cloud's DynamoDB. In the public cloud, the user's computation is performed on encrypted data. When the results are needed, they can be downloaded to the client's computer. Table 1 represents the evaluation parameter used by different authors.

Table 1 Evaluation parameter used in different cryptography technique

References	Cryptography technique	Evaluation parameter			
		CPU/GPU utilization (%)	Memory utilization (MB)	Execution time (ms)	
				Encryption (ms)	Decryption (ms)
[19]	BGV	–	–	35,000 (For 4000 data) 58,000 (For 6000 data) 69,000 (For 8000 data) 95,000 (For 10,000 data)	
[20]	BGV	–	–	1200	
[21]	RSA, Paillier and Elgamal	–	–	RSA-10830 Paillier-21223 Elgamal-2264 MPaillier-331	RSA-13054 Paillier-26627 Elgamal-3361 MPaillier-138
[22]	DGK			2500 (Input size = 15) 8500 (Input size = 55) 14,000 (Input size = 95)	
[23]	BFV		140	91.6	19.5
[24]	RSA	–	15	–	–
[25]	CKKS	40	–	170 (Total execution time)	
[26]	RSA	–	148	1700 (Total execution time)	
[27]	Elgamal	–	–	10 (For 256 bits) 26 (For 512 bits) 111 (For 1024 bits) 849 (For 2048 bits) 10,075 (For 4096 bits)	1 (For 256 bits) 3 (For 512 bits) 9 (For 1024 bits) 28 (For 2048 bits) 84 (For 4096 bits)

3 Working Principle

In this section, the working principle adopted during the current research work is elaborated along with various HE techniques that have been discussed which are being considered during the work.

Initially, the EC2 instance of AWS is created with a 4v CPU and 32 GiB memory with an Intel i7 processor with 3.2 GHz clock speed. Figure 1 shows the

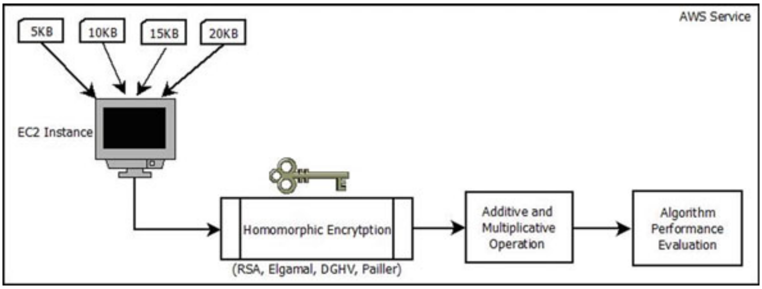


Fig. 1 Working principle of the adopted methodology

overall working of the adopted methodology during the research work. The detailed procedure adopted during the research work is as follows

Step1: Create the Amazon EC2 instance with proper configuration.
Step2: Consider various text files having sizes 5 KB, 10 KB, 15 KB, and 20 KB as the input to the created EC2 instance.
Step3: To the input text file different homomorphic encryption algorithms such as RSA, Paillier, DGHV, and Elgamal encryption algorithms are implemented.
Step4: Additive and Multiplication Operation are performed on the ciphertext.
Step5: All of the above-said algorithms are evaluated based on some influencing parameters such as memory and CPU utilization.

3.1 RSA

RSA [21, 24, 26, 28] is one of the examples of asymmetric cryptography in which the public key and private key are used for encrypting and decrypting the data, respectively. There are three distinct steps present in this methodology named as key generation, encryption, and decryption phase.

Key Generation

1. Choose A and B as the prime number
2. $M = A \times B$
3. $\phi(M) = (A - 1) \times (B - 1)$
4. Choose public key e such that gcd $(\phi(M), e) = 1$
5. Private Key $d = e^{-1} \text{mod } \phi(M)$.

Encryption

The plaintext is converted into ciphertext as given in Eq. 1.

$$\text{Cipher} = (\text{Message})^e \text{ mod } N \tag{1}$$

Decryption

In this the plain text can be obtained as in Eq. 2.

$$\text{Message} = (\text{Cipher})^{d} \bmod N \tag{2}$$

Multiplicative Homomorphic Property

Let $E()$ is used as the encryption which takes the Message1 and Message 2 plaintext as the input and produces $C1$ and $C2$ as two ciphertext. This function can be defined in Eqs. 3 and 4.

$$E(\text{Message1.Message2}) = E(\text{Message1}).E(\text{Message2}) \tag{3}$$

$$C1.C2 = (\text{Message1.Message2})^{e} \bmod N \tag{4}$$

3.2 Elgamal

Elgamal [27] HE is asymmetric key cryptography that is based on a public key. There are three different steps present in this methodology named as key generation, encryption, and decryption phase. The steps of this method is elaborated as below.

Key Generation

1. Choose any two large numbers m and n such that $n < m$ and gcd $(m, n) = 1$
2. Generate a cyclic group G_m
3. Randomly select $x \in \{1, 2, \ldots, m - 1\}$
4. Choose p from G_m as the public key
5. q is the private key which can be calculated as p^x.

Encryption

1. Select c from G_m
2. $D = m^c$ and $e = q^k = p^{xk}$
3. Ciphertext = Message * e.

Decryption

1. $e' = d^x$
2. Message = Ciphertext * e'.

Multiplicative Property

Let $E()$ is used as the encryption which takes Message1 and Message 2 plaintext as the input. $C1$ and $C2$ are two ciphertexts then the homomorphic property can be found by Eqs. 5 and 6.

$$E(\text{Message1.Message2}) = E(\text{Message1}).E(\text{Message2}) \tag{5}$$

$$C1.C2 = (\text{Message1.Message2})^e \bmod N \tag{6}$$

3.3 Paillier

Paillier [28] homomorphic encryption algorithms have the additive property of the homomorphic process. Key generation, encryption, and decryption are three distinct phases present in this method.

Key Generation Phase

1. Choose any two prime number A and B
2. Find $N = A \times B$
3. $\mu = \text{LCM } (A - 1, B - 1)$
4. $L(z)$ as $(z - 1)/n$ and X is the plain text where $0 < X < n$
5. Choose any integer a such that $1 < a < n^2$
6. $\rho = L(a^{\mu} \% \, n^2)^{-1}$
7. Choose (n, a) as the public key
8. (μ, ρ) = Private key pair.

Encryption

Let Message X be the plain text that lies between 0 and n and C is the ciphertext. So the ciphertext can be defined as the following

1. Choose a random integer y in the range 0 and n
2. $C = (a^X y^n) \ \% \ n^2$.

Decryption

1. $X = [L\ (C^{\mu} \ \% \ n^2)].[\rho \ \% \ n]$.

Additive Property

Let $E()$ is used as the encryption which takes Message1 and Message 2 plaintext as the input. $C1$ and $C2$ are two ciphertexts then the additive property can be defined as Eqs. 7 and 8.

$$E(\text{Message1} + \text{Message2}) = E(\text{Mesggae1}) + E(\text{Message2}) \tag{7}$$

$$C1.C2 = a^{(\text{Message1+Message2})}.(y1.y2)^n \ \% \, n^2 \tag{8}$$

3.4 DGHV

Dijk, Gentry, and Halevi [9] in 2010 presented an FHE technique based on the modular operations over integers The method is dependent on a method, which entails creating a symmetric HE from an integer, converting it to an asymmetric SWHE by merging an Approximate Greatest Common Divisor (AGCD), then compressing the SWHE to speed up decryption (fully homomorphic encryption). This technique has the benefit of requiring only integer-based operations and not operations based on ideal lattices of a polynomial ring.

Key Generation

1. Choose x as the odd number.
2. Choose p as the random number such that x mod p is even.
3. $P = \{x0, x1, x2, \ldots, xn\}$
4. $r \in (-2^p, 2^p)$.

Encryption

$$\text{Ciphertext} = \left(M + 2r + 2\sum xi\right) \bmod x_0$$

Decryption

$$M = (C * r) \bmod x_0$$

4 Result with Statistical Analysis

The following is a list of the hardware and software used in this experiment: The processor is an Intel i5 11th generation, memory is 8 GB (2 * 4 GB), the hard drive is a 1 TB SATA II with 2.4 GHz clock speed and Linux is the operating system. The RSA, ElGamal, and DGHV algorithms were used to encrypt and decrypt 20 MB files in this research. The system is connected with the AWS cloud service with a target environment as Amazon Linux AMI 2018.03 with SSD as the volume. Then r6g.xlarge EC2 having 4v CPU and 32 GiB memory with Intel i7 processor with 3.2 GHz clock speed has been adopted on which the entire work has been executed. On EC2 the JDK environment is installed and the above said encryption algorithm has been implemented with different input size. For encryption and decryption process over the provided text files having sizes 5, 10, 15, and 20 KB. Table 2 shows the overall computational time for the encryption and decryption process for the above-said methodology. The encryption and decryption time of all of the described methods are measured by using the Java currentTimeMillis method. Two variables are named as start–end which are initiated by the time generated by system.currentTimeMillis(). Finally, the lapse time

Table 2 Encryption and decryption time for different algorithms

	Encryption (in s)				Decryption (in s)			
	5 KB	10 KB	15 KB	20 KB	5 KB	10 KB	15 KB	20 KB
RSA	91	110	136	151	176	191	232	256
ElGamal	112	134	176	191	191	212	231	264
Paillier	182	234	302	368	245	357	402	472
DGHV	169	198	247	276	213	268	327	342
Avg time (in s)	138.5	169	215.25	246.5	206.25	257	298	333.5

is calculated by subtracting these variables. Figure 2 shows the performance comparison different algorithm in contrast to average encryption and decryption time. Table 3 shows the mathematical problem comparison upon which the above-mentioned algorithms work. Figure 4 shows the memory utilization of the program which is being calculate by differentiating two Java system call totalmemory() and freememory() of runtime type. Figure 3 shows the CPU utilization of the above discussed homomorphic encryption algorithms upon different input size which can be calculated by diving SystemLoadAverage() and SystemAvailableProcessors() of system class OperatingSystemMXBean.

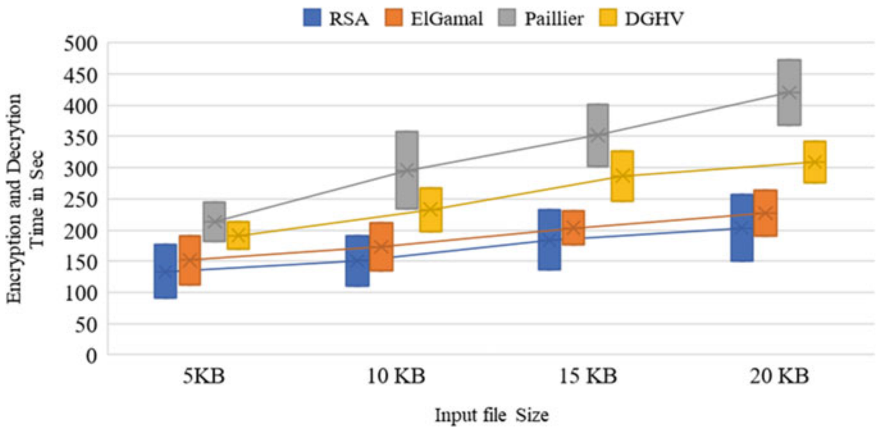


Fig. 2 Performance comparison of different mechanism with encryption and decryption time

Table 3 Mathematical problem comparison

Algorithms	RSA	ElGamal	DGHV	Paillier
Mathematical problems	Linear	Discrete	AGCD	Discrete

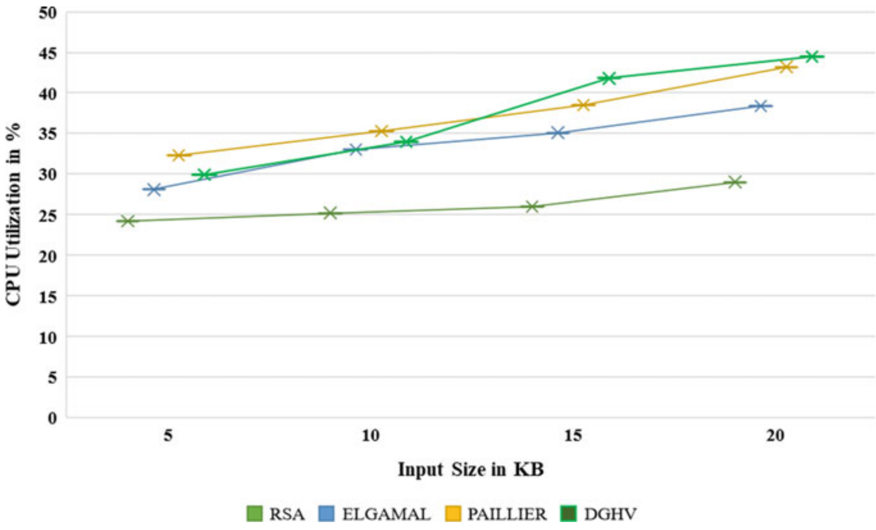


Fig. 3 CPU Utilization of different algorithms with different input size

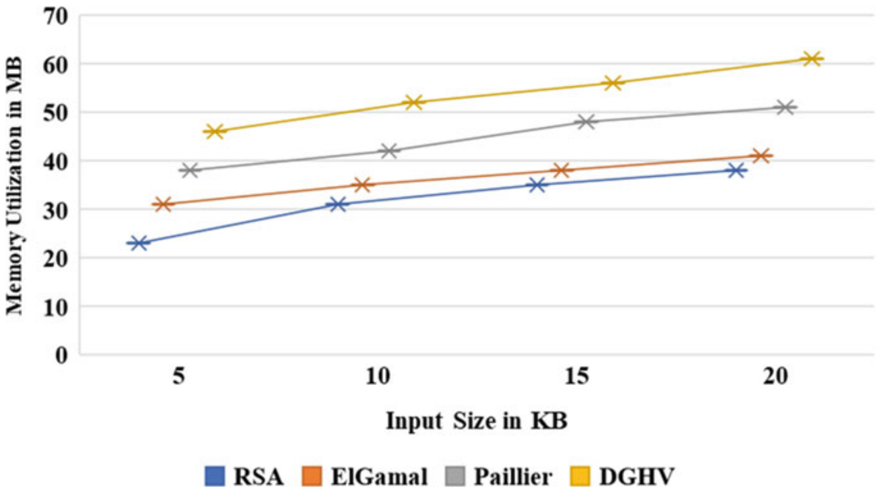


Fig. 4 Camparison of memory utilization

5 Conclusion

With the widespread adoption of cloud computing, cloud platform security has emerged as a serious challenge, which can, in turn, prove as a hindrance towards the development of cloud infrastructure. To provide security four possible approaches have been focused on in the current research work. The computational overhead caused due to the conventional cryptography mechanism shows the path for homomorphic cryptography. In this current research work, four HE techniques have been implemented such as the RSA, Elgamal, Paillier, and the DGHV. Finally, these four algorithms have been evaluated for encrypting and decrypting different files of sizes 5, 10, 15, and 20 KB in the AWS cloud system. The performance of these above-said mechanisms has been compared in terms of encryption, decryption time, memory, and CPU utilization. The experimental analysis shows that the performance of the RSA homomorphic algorithm performs better with an average of 138.5 and 206.25 for encryption and decryption time in second, respectively. The average memory utilization of RSA is found 30 MB and the average CPU utilization of 27%.

References

1. Alaya B, Laouamer L, Msilini N (2020) Homomorphic encryption systems statement: trends and challenges. Comput Sci Rev 36:100235
2. Shen W, Su Y, Hao R (2020) Lightweight cloud storage auditing with deduplication supporting strong privacy protection. IEEE Access 8:44359–44372
3. Zhang Y, Yu J, Hao R, Wang C, Ren K (2018) Enabling efficient user revocation in identity-based cloud storage auditing for shared big data. IEEE Trans Dependable Secure Comput 17(3): 608–619. https://doi.org/10.1109/TDSC.2018.2829880
4. Elhabob R, Zhao Y, Sella I, Xiong H (2019) Efficient certificateless public key cryptography with equality test for internet of vehicles. IEEE Access 7:68957–68969
5. Deng H, Qin Z, Wu Q, Guan Z, Deng RH, Wang Y, Zhou Y (2020) Identity-based encryption transformation for flexible sharing of encrypted data in public cloud. IEEE Trans Inf Forensics Secur 15:3168–3180
6. Yang Y, Huang Q, Chen F (2018) Secure cloud storage based on RLWE problem. IEEE Access 7:27604–27614
7. Qiu S, Wang B, Li M, Liu J, Shi Y (2020) Toward practical privacy-preserving frequent itemset mining on encrypted cloud data. IEEE Trans Cloud Comput 8(1):312–323
8. Liao Y, Zhang G, Chen H (2020) Cost-efficient outsourced decryption of attribute-based encryption schemes for both users and cloud server in green cloud computing. IEEE Access 8:20862–20869
9. Ahuja R, Mohanty SK (2017) A scalable attribute-based access control scheme with flexible delegation cum sharing of access privileges for cloud storage. IEEE Trans Cloud Comput 8(1):32–44
10. De SJ, Ruj S (2020) Efficient decentralized attribute based access control for mobile clouds. IEEE Trans Cloud Compu 8(1):124–137
11. Min Zhao E, Geng Y (2019) Homomorphic encryption technology for cloud computing. Procedia Comput Sci 154:73–83
12. Tran J, Farokhi F, Cantoni M, Shames I (2019) Implementing homomorphic encryption based secure feedback control for physical systems. arXiv preprint arXiv:1902.06899
13. Liu L, Chen R, Liu X, Su J, Qiao L (2020) Towards practical privacy-preserving decision tree training and evaluation in the cloud. IEEE Trans Inf Forensics Secur 15:2914–2929
14. Brakerski Z (2019) Fundamentals of fully homomorphic encryption. In: Providing sound foundations for cryptography: on the work of Shafi Goldwasser and Silvio Micali, pp 543–563
15. Zheng S, Wang Y, Hu D (2019) Lossless data hiding based on homomorphic cryptosystem. IEEE Trans Dependable Secure Comput 18(2):692–705
16. Joshi S (2020) An efficient Paillier cryptographic technique for secure data storage on the cloud. In: 2020 4th international conference on intelligent computing and control systems (ICICCS). IEEE, pp 145–149

17. Asuncion AEC, Guadalupe BCT, Yu WES (2019) Implementation and analysis of homomorphic facial image encryption and manipulation. In: Proceedings of the 2019 4th international conference on multimedia systems and signal processing, pp 158–166
18. Yan X, Lu Y, Liu L, Wan S, Ding W, Liu H (2020) Exploiting the homomorphic property of visual cryptography. In: Cryptography: breakthroughs in research and practice. IGI Global, pp 416–427
19. Alabdulatif A, Khalil I, Yi X (2020) Towards secure big data analytic for cloud-enabled applications with fully homomorphic encryption. J Parallel Distrib Comput 137:192–204
20. Rahman MS, Khalil I, Atiquzzaman M, Yi X (2020) Towards privacy preserving AI based composition framework in edge networks using fully homomorphic encryption. Eng Appl Artif Intell 94:103737
21. Seth B, Dalal S, Kuma R. Hybrid homomorphic encryption scheme for secure cloud data storage. In: Recent advances in computational intelligence. Studies in computational intelligence, vol 823. http://doi.org/10.1007/978-3-030-12500-4_5
22. Alexandru AB, Gatsis K, Shoukry Y, Seshia SA, Tabuada P, Pappas GJ (2021) Cloud-based quadratic optimization with partially homomorphic encryption. IEEE Trans Autom Control 66(5)
23. Mert AC, Öztürk E, Savaş E (2020) Design and implementation of encryption/decryption architectures for BFV homomorphic encryption scheme. IEEE Trans Very Large Scale Integr (VLSI) Syst 28(2)
24. Park J, Kim DS, Lim H. Privacy-preserving reinforcement learning using homomorphic encryption in cloud computing infrastructures. http://doi.org/10.1109/ACCESS.2020.3036899
25. Al Badawi A, Hoang L, Mun CF, Laine K, Aung KMM. PrivFT: private and fast text classification with homomorphic encryption. http://doi.org/10.1109/ACCESS.2020.3045465
26. Wu W, Liu J, Wang H, Hao J, Xian M (2021) Secure and efficient outsourced k-means clustering using fully homomorphic encryption with ciphertext packing technique. IEEE Trans Knowl Data Eng 33(10)
27. El-yahyaoui A, Ech-Cherif El Kettani MD (2019) A verifiable fully homomorphic encryption scheme for cloud computing security. Technologies 7:21. https://www.mdpi.com/journal/technologies. http://doi.org/10.3390/technologies7010021
28. Poteya MM, Dhote CA, Sharma DH (2016) Homomorphic encryption for security of cloud data. In: International conference on communication, computing and virtualization. http://doi.org/10.1016/j.procs.2016.03.0237th

Comparative Analysis of Full Training Set and Cross-Validation for Machine Learning Approach to Run Smart Wheelchair

Jenamani Chandrakanta Badajena, Srinivas Sethi, and Ramesh Kumar Sahoo

Abstract A smart wheelchair can be perceived as a wheelchair that can provide autonomous mobility and features to persons who are incapable of using their own body for self-mobility. There are broadly two categories of smart wheelchairs. In the first category, there are provisions for the usage of wheelchairs using in-built functions built into the wheelchair. In the second category, the smart wheelchair can use the signal obtained from human brain using *Brain–Computer Interface* (*BCI*), for the movement of human being who are incapable of using their own body for the various functions of a wheelchair. It has been substantial research on providing accuracy for the movement of smart wheelchairs. In this regard, there can be significant differences between the accuracy received from the user-supplied dataset and from the operational aspect of the wheelchair when put to the real operational environment. We have hence tried to improve the accuracy of our model by incorporating a cross-validation approach to our model. Cross-validation is a statistical method for evaluation of machine learning methods by segregating two different sections where the first section is used to develop and learn a model using training dataset. Further, it has been validated using testing dataset. Finally, test the stability of our machine learning model with real-time dataset.

Keywords BCI · Smart wheelchair · Accuracy · Cross-validation · Stability · Machine learning

J. C. Badajena (✉)
Utkal University, Bhubaneswar, Odisha, India
e-mail: chand.cet@gmail.com

S. Sethi · R. K. Sahoo
Indira Gandhi Institute of Technology, Sarang, Dhenkanal, India

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_10

1 Introduction

BCI is a thought translation process that can enable the communication between machine and human brain activity. The activity of the human brain is typically recorded by the Electroencephalography (EEG) signals. EEG is an electrophysiological activity monitoring method that is used to capture brain signals. In our experimental setup, we captured the brain signal from the participant at the time of attention and meditation. The general purpose of this study is fourfold. The brain activity is examined in the firstfold. In the secondfold, a stable environment with various scenarios is developed. The thirdfold is to evaluate the concept using a machine learning method, and the last fold is to implement the concept in a smart wheelchair to complete the task. Brain waves control an autonomous smart wheelchair. The human makes decisions to move the wheelchair (forward, backward, right, left, and stop) using smart controls [1], and the motion is automated accordingly. The developments in a smart wheelchair outfitted with environmental sensors, an HD camera, a 3D scanner, a Bluetooth device, and a computer-based system as the main processing unit, with a special emphasis on computer cluster architecture. In addition to the available functionalities, an intelligence technique for user control is incorporated into the wheelchair platform. The aim of such functionality is to provide a BCI framework that can be expanded to provide more natural and engaging control. This includes developing, testing, evaluating, and deploying a novel data analysis approach for the study of human cognitive function and the psychological and behavioral consequences of a BCI system on a human user's cognitive ability. In this paper, we have addressed the model's architecture and working process, as well as the technique of data collecting, analysis, and implementation in machine learning, and how to transform it into a wheelchair motor instruction.

Furthermore, we have tried to refine the productivity of a smart wheelchair using a cross-validation approach. In other words, we have tried to test the stability of our model when the wheelchair is put to perform in the actual scenario. The stability of the model can be referred to as the efficiency of the smart wheelchair. In our study, we have used the *k*-folds approach for cross-validation [2]. The improved stability in the cross-validation approach will guarantee a high accuracy of our model when put into performing in a real or unseen scenario. This will, in turn, improve the usability and user experience for an immobile person using a smart wheelchair.

2 Background

The author in [3] addresses that due to the lesser amount of data for each participant in the EEG Movement/Imagery collection, none of the classifiers can be deemed the best. As a result, most classifiers require a train-validate-test split of the data. The authors selected models without any loss of significant information. In [4], the authors represent the use of a 3D convolutional neural network (CNN) model to uphold spatial and temporal information for multi-channel EEG sensor data on an attention detection activities. It has been proposed to evaluate a public dataset, and the proposed model outperformed the baseline method on intra, and intra subject, and subject-adaptive classification circumstances. An attention optimization method aims to increase the performance of the attention value output while reducing the real-time data transmission latency, and the approach has some practicality and promotion value [5]. To utilize brain-controlled mobile robots useable in actual circumstances, initially, BCI system performance must be improved [6]. Second, given all existing BCI systems' restricted and inconsistent performance, it is critical to identify strategies to enhance and assure the robotic system's overall driving performance. Finally, a consistent performance assessment technique should be devised for performance analysis of such methods and systems. An experiment imitating the everyday application was conducted to validate and assess available technology [7]. The activities included having a user driving the structure closer to a person and talking with him, traveling to another space via a door and performing some real-world tasks, and some healthy participants in the test and completing all of the objectives. Chunmei [7] proposes a coefficient-based strategy based on machine learning (ML) utilizing EEG sensor signals to address the problem of poor alertness of emotion stimulation mechanisms in most previous investigations. This solution proved to surpass the benchmark approaches in terms of correctness or accuracy and outperformed the benchmark algorithms in terms of speed. KNN [8] algorithm and multilayer perception neural network (MLPNN) [8, 9] were used to categorize EEG data from negative and positive emotions. The channel selection was built up by the classification algorithm. Individual participants used the EEG channel to accomplish various classifications. The generated feature vector is utilized to categorize related negative and positive emotions using a corresponding classifier. Euclidean distance, Minkowski distance, and hamming code, the KNN classifier was combined with a new object were used for feature extraction and distance computation.

The several forms of EEG signals, along with their frequency ranges and amplitudes. These signals were utilized to design further and simulate the proposed BCI framework, which captures perceptual and sensitive information from the temporal, frontal, parietal, and occipital lobes [8].

3 Methodology

In our approach, we have used test validation to collect data from some real participants. From the test validation, we have tried to predict the stability of our proposed model using a cross-validation approach. We have not used the supplied test and percentage split approach as the scope of our dataset is not very large in number.

In the case of machine learning approaches, getting a higher value of accuracy may not always yield the same accuracy when the model is put for usage in the real world. In other words, the high value of efficiency may not always be a suitable condition for the model to be selected as a suitable one. Hence, the model has to go through resampling in terms of cross-validation approaches. The cross-validation approach has different ways of implementation. We have used k-fold cross-validation method to predict the steadiness of our model. In k-fold cross-validation, the entire dataset needs to be divided into k number of subsets or folds [2]. In the first step, the model is trained using the whole group, and in the second step, the data items need to be divided into k subsets, and each subset is used to test the model one after the other. This model is considered to be an unbiased one, and it also uses involves training as well as testing using every subset of the dataset.

The data was obtained from a hundred individuals using an EEG machine (NeuroMax32) and processed using machine learning techniques such as SVM, Decision Tree (J48), and Naive Bayes classifier. Then the command is sent to the wheelchair's motor, and the wheelchair begins to work based on the command data, and the process is repeated symmetrically. The data was collected from the human brain by the EEG machine. It was analyzed by machine learning and given to the microcontroller through Bluetooth attached to the wheelchair. Further, the command will be sent to the motor of the wheelchair. The wheelchair began working as per the data given by the Bluetooth, and the process will continue in this manner, and the data was also collected from the person.

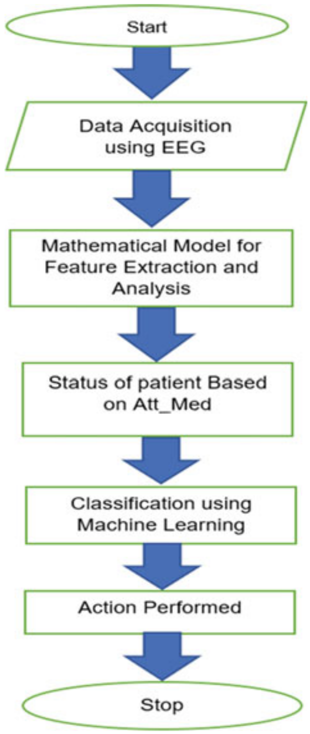
The proposed paradigm for categorizing attention and meditation in various situations is described in this part.

In the first step, features are collected from the NeuroMax 32 device by using the feature extraction principle in the form of EEG signals from several lobe points of the brain. The features were selected using the feature selection principle, and then they were segmented using an equal distribution method. After that, the data was preprocessed in anticipation of extraction in the future. Then, by the selection of the proper EEG channel, the feature vector was generated. The data was then examined and grouped into positive and negative. Data from different lobes will be paired and analyzed. T6-O2, F8-T4, T4-T6, FP2-F8, and so on for right sides, P3-O1, FP1-F7, T3-T5, F7-T3, and so on for left sides. P3-O1 and T4-T6 are two pairings that produce EEG response data that spans from 0 to 100 and represent a person's meditation and concentration level. The collected data and EEG signal were converted using the Digital Wavelet Transform (DWT). The model's last task is classification and prediction. It was employed throughout the analytical stage. The classifier was then used to obtain accuracy. The flowchart of our proposed work is shown in Fig. 1.

4 Data Analysis and Modeling

Let us consider $W1$ and $W2$ as the weight factors assigned to the attention label Al and the meditation label Ml, respectively. We have provided 70% of our attention to the attention level, so $W1 = 0.7$, and 30% of our attention to the meditation level, so $W2 = 0.3$, and we have used Eqs. (1) and (2) to describe the person's status using attention and meditation [10].

Fig. 1 Flowchart for proposed work



Al = data captured from *T4–T6* ("set of lobes points are considered for attention").
Ml = data captured from *P3-O1* ("set of lobes points are considered for meditation")

$$\text{Att\_Med} = Al * W1 + Ml * W2 \tag{1}$$

$$\text{Att\_med}_{\text{status}} = \begin{cases} \text{low} & \text{for Att\_med} < 35 \\ \text{medium} & \text{for } 35 < \text{Att\_med} < 55 \\ \text{high} & \text{for Att\_med} > 55 \end{cases} \tag{2}$$

In Eq. (1), Att_med reflects a person's mental status based on his attention and mediation level, which has been further classified as low, medium, and high as shown in Eq. (2). The wheelchair's speed will be regulated based on the person's mental state. It can also control the wheelchair's stop and move. In this work, the wheelchair will halt when Att_med status is low, move slowly in medium status, and move steadily in high status. To do so, we examined the data using several machine learning methodologies in order to assess the model's accuracy [10].

The final step to manage the wheelchair on the choice of the suitable movement of a person is classification, and this can be done with the implementation of ML algorithms, which is a part of AI. Machine learning models and applications can be categorized into supervised machine learning, unsupervised machine learning, and reinforcement machine learning. Classification of the proposed model has been done using various machine learning algorithms in full training set and cross-validation modes in WEKA Simulator. The full training set approach used only when users have all the data and want to create a descriptive model rather than a predictive model. We don't need to make new predictions because we have all of the data. We want to build a model to better understand the problem, prepare the proposed model using the entire training dataset, and then test it using the dataset that we have taken in this work.

There is not a single algorithm that can be effective even after being trained with all patterns from a dataset. To overcome this situation, we used the cross-validation technique in our work. Cross-validation is a statistical technique for evaluating a learned model and testing the performance of the proposed model. Cross-validation is generally used in machine learning. The technique also helps to compare and check the stability of our purposed model. The technique is easy to implement, understand, and it provides an accurate performance. The cross-validation technique is a powerful technique for selecting the best model for our proposed model.

The dataset is first split into equal k-folds. For each k-fold in our dataset, the model was put to test with $(k - 1)$ folds of the dataset, and in each iteration, errors were observed and recorded. This step was repeated until all folds of the k-fold were taken into iteration. If the received efficiency is lower than the results obtained previously, then it can be regarded as underfitting and a higher value will be regarded as overfitting. In our setup, we segregated our data into ten-folds.

- **Implementation of the full training set in the proposed model**

  1. Define the set of parameters and load the full training set (T_{ds})
  2. Data is separated into two groups: training and testing.
  3. Prepare preprocessing with the help of the training set.
  4. Before we start training, we need to build and optimize the model for training set execution

     – Regularization of parameters.
     – Parameter tuning.

  5. Test and set the best model for test set execution

     – For classification of mathematical model.

  6. To get the final result for the test and set, the best model for test set execution: for accuracy.

- **Implementation of k-fold Cross-Validation in the proposed model**

  1. Define the set of parameters combination W, for the current smart wheelchair model. İf the current smart wheelchair model has no parameters, W is the empty set.

2. Segment the dataset into k-folds with approximately equitable distribution of cases and controls.
3. For fold k_t in the k-fold.
   1. set the fold k_t as test set and the rest $(k - 1)$ folds consider as training set.
   2. Train the model on the dataset mentioned as training set. For each iteration, a new working model is trained independent of the model in the previous iteration.
   3. Validate the test set and evaluate the model performance.
   4. To get the final result to calculate the average performance over k-folds.

5 Performance Evaluation Parameter

The various parameters used for evaluating machine learning algorithms used for classification in the proposed work are {*False Positive* (*FP*), *True Positive* (*TP*)} *rate*, *Recall*, *Precision*, *F-Measure*, *and* {*Mean Absolute Error* (*MAE*), *Relative Absolute Error* (*RAE*), *Root Mean Squared Error* (*RMSE*), *and Root Relative Squared Error* (*RRSE*)}.

Certain data will be accurately identified as positive (*TP*) for each conceivable parameter value chosen to discriminate between two classes or separate cases, while others may be not in correctly classified as negative (*FN*).

The *FP rate* is computed as $\frac{\mathrm{FP}}{\mathrm{FP+TN}}$, where *FP* stands for *false positives* and *TN* represents for *true negatives* is the total no. of negatives. It's the possibility of a false alarm being generated in which a positive result is returned when the real value is negative.

$$\text{Recall} = \frac{\text{TPR}}{\text{FN} + \text{TPR}} \tag{3}$$

$$\text{Precision} = \frac{\text{TPR}}{\text{FPR} + \text{TPR}} \tag{4}$$

$$F\text{-Measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

Precision, Recall, and F-Measure has been calculated using Eqs. (3), (4), and (5). The test accuracy in the statistical binary classification method can be measured using F-score or F-Measure. The value of the F-score lies between 0 and 1, where value 1 indicates perfect accuracy of Precision and Recall, and value 0 shows both the Precision and Recall are zero, and their accuracy cannot be considered further. Instances that have been correctly categorized are utilized to compile true positive and true negative occurrences (TP + TN).

The same erroneously categorized examples were utilized for both false negative and false positive classification (FP + FN). The receiver operating characteristic is a graph for performance evaluation that can be useful in classifying performance from the use of many classifiers (ROC). This aids in the presentation of the link between true positive and false positive rates. A diagonal method is represented by ROC, a machine learning prediction concept. The performance evaluation includes three parameters: Mean Absolute Error (MAE), Root Mean Square Error (RSME), and Relative Absolute Error (RAE) (RAE). The MAE measures the average measurement of erroneous items in a given collection of values, where values represent the probabilistic data item distribution. The standard deviation, also known as the root mean square error (RSME), supplied information on the departure of individual data items from the mean, or, in other words, it offered a mathematical picture of how near the data points are to the mean. Relative Absolute Error (RAE) is a method for calculating an analytical model's correctness. The absolute rate is abbreviated as RAE.

6 Experimental Setup

The experiment have been done through NeuroMax32 EEG headset with 32 channels, as per our previous work [10] to record the brain wave utilizing 32 electrodes put on the human skull. The connecting cables have two ends, one of which is attached to the NeuroMax board and the other to the electrodes. EEG signals will be obtained from the brain by an EEG headset through electrodes placed on the head and transferred to the NeuroMax board, which will then send the acquired signals to the connected computer via USB serial connection in the second step. The EEG signals are recorded by the computer's NeuroMax software, and the results are shown in a graphical perspective. The data in the graphical view is translated into an excel format in the third stage, which is then used for analysis by using the cross-validation technique to check the effectiveness of the model in the suggested work.

7 Result and Discussion

This sections presents the results of full training set evaluation and *k*-fold cross-validation for brain-actuated wheelchair model.

İn Fig. 2, when we get the data, after data cleaning, preprocessing the first step we do is to feed it to an outstanding model, so we can measure the effectiveness of our model to validate the proposed model. We have used the classifier as *X*-axis and no, of instances in *Y*-axis, respectively, no. of instances classified as low, medium, and high using approaches like *Decision Tree*, *SVM*, and *Naive Bayes algorithms*. We can say that, the results for *Decision Tree* algorithm closely matches with the no. of instances provided by proposed mathematical model as reflected in Fig. 2 and therefore, *Decision Tree* algorithm are more suitable than *Naive Bayes* and *SVM* algorithm for the proposed model.

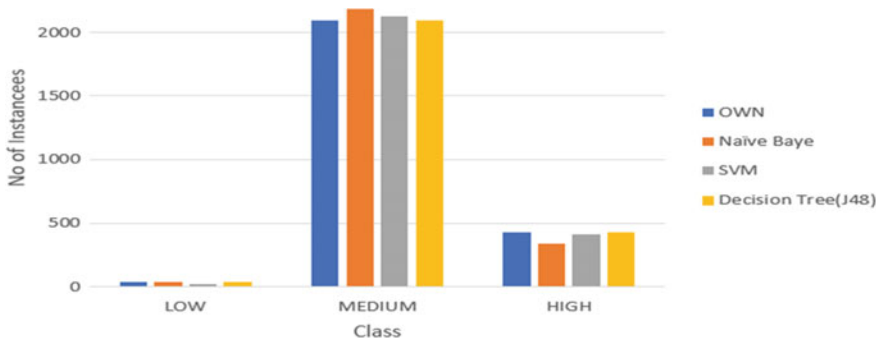


Fig. 2 Comparison of proposed mathematical model with various machine learning algorithms

From Fig. 3, we can infer that the *cross-validation* approach can correctly classify data instances with slightly lesser efficiency when compared with the full training set. In the same light, the amount of incorrectly classified instances in the full training set is close to zero, whereas for cross-validation, the same is slightly higher than the former value. The *root mean square error* and the Relative Absolute Error rate are slightly higher for the *cross-validation* approach when compared with the full training set. In the performance evaluation parameters, *k-statistics* for both parameters have a very less difference. Similarly, the *mean absolute error* and *the root mean square* for *cross-validation* are slightly more than that when used for the full training set. It is evident that the *TP, Precision, Recall, F-Measure, MCC, ROC Area, and PRC Area* values are almost same, and the *FP rate* is only negligibly higher than that of full training set when compared with *cross-validation* approach.

In Fig. 4, it is observed that the accuracy level of the *Decision Tree* is higher in comparison *to Naïve Bayes* and *SVM*. The accuracy level in Decision Tree is 98.98% whereas in SVM it is 98.44% and in Naïve Bayes is 95.23%. The incorrectly classified instance is negligible in both *SVM* and *Decision Tree* and a little in *Naïve Bayes* and also we find that in Decision Tree, we get the minimal error in comparison with the *SVM* and *Naïve Bayes*. It observed that in kappa-statistic value for Decision Tree is higher than SVM and Naïve Bayes, whereas Mean Absolute Error and *Root Mean Squared Error rate* is minimal. We have compared the *TP Rate*, *FP Rate*, *Precision*, *Recall*, *F-Measure*, *MCC*, *ROC*, and *PRC* area for *Naïve Bayesian, SVM*, and *Decision Tree* methods. We find that in the *Decision Tree* all the classes are the same except the *FP rate*. The experimental environment to evaluate and validate BCI system performance is illustrated using the training and testing accuracy of the system. Finding ways to improve and assure the overall performance of the system is critical given the limits of the restricted and unstable performance validation technique to improve the overall performance of the brain-actuated wheelchair.

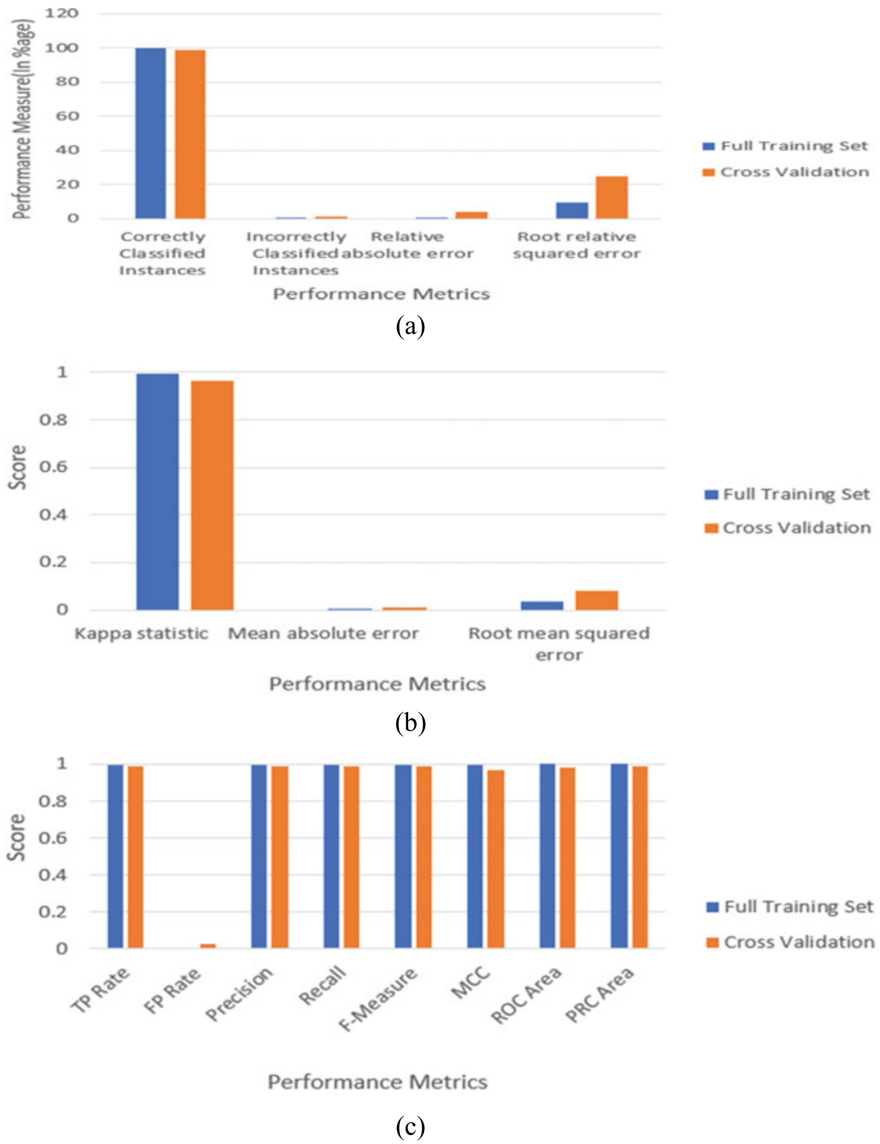
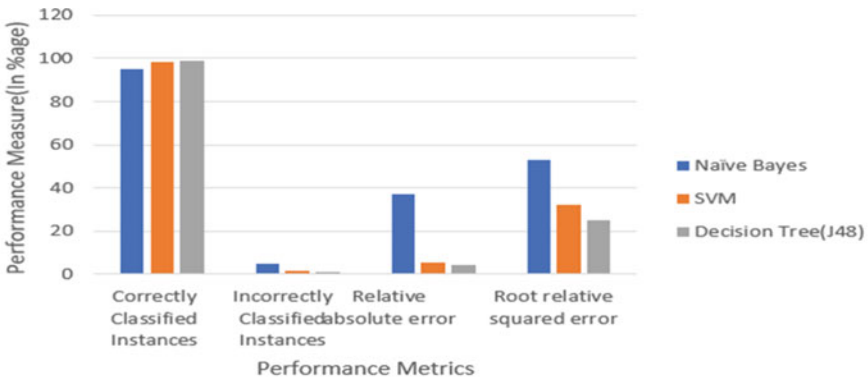


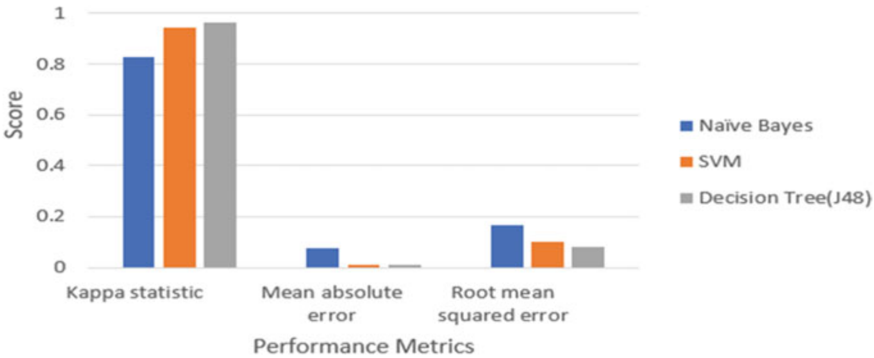
Fig. 3 Classification of proposed model using decision tree (J48) algorithm in full training set and cross-validation mode

8 Conclusion

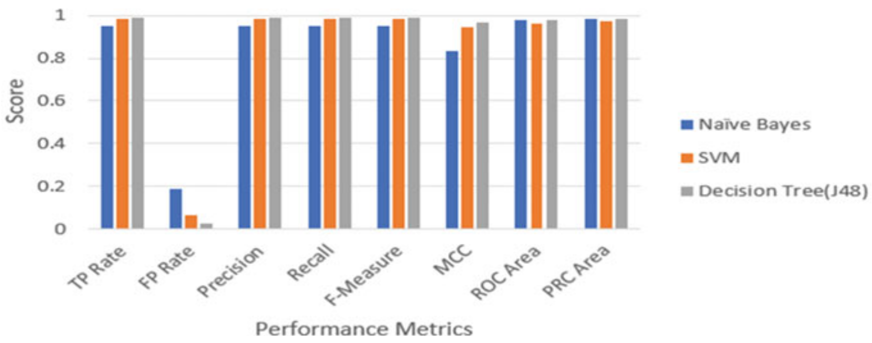
Smart wheelchairs can be used as utmost necessity for disabled people. İn this objective, we have tried to evaluate our machine learning model for improved accuracy when put to the test in a real or unknown environment. The entire dataset was first



(a)



(b)



(c)

Fig. 4 Classification of proposed model using various algorithms used to train our model for the functionality of our smart wheelchair. İn the next step, our model was used for cross-validation using the *k*-fold approach. Our results after using cross-validation are compared in Fig. 3a–c. From the analysis of Fig. 4a–c, we can mention that the cross-validation approach yields slightly higher root relative squared error, slightly high root mean square error, and a slightly high amount of false positive rate. The accuracy obtained from the cross-validation approach is slightly less than the efficiency received when the entire dataset was used for training. Under these circumstances, we can view that the machine learning model which has designed is relatively stable and can provide better functionality to end users when put to the test or used in a real environment. The researchers can be used cost–benefit analysis methods to improve the accuracy and efficiency of the model in the future.

References

1. Bi L, Fan X, Liu Y, EEG-based brain-controlled mobile robots: a survey. IEEE Trans Hum Mach Syst 43(2):161–176
2. https://docs.weka.io/. Retrieved as on 05 Dec 2021
3. Nekrasova J, Kanarskii M, Suareva A (2020) Comparison of classification algorithms for motor imagery brain-computer interface. In: IEEE Explore, 978-1-7281-9115-7
4. Zhang Y, Cai H, Nie L, Xu P, Zhao S, Guan C (2021) An end to end 3D convolutional neural network for decoding attentive mental state. Neural Netw 144:129–137, 0893–6083
5. Sahu PK, Sahoo RK, Sethi N, Sethi S (2020) Emotion classification based on EEG signals in a stable environment. In: 2020 international conference on computer science, engineering and applications (ICCSEA), pp 1–5. https://doi.org/10.1109/ICCSEA49143.2020.9132966
6. Tang J, Liu Y, Hu D, Zhou ZT (2018) Towards BCI-actuated smart wheelchair system. BioMed Eng Online
7. Qing C, Qiao R, Xu X, Cheng Y, Interpretable emotion recognition using EEG signals. In: Data-enabled intelligence for digital health. IEEE Access
8. Wu Y, Xie N, Attention optimization method for EEG via the TGAM. In: Hindawi computational and mathematical methods in medicine, vol 202, p 11
9. Ozerdem MS, Polat H (2017) Emotion recognition based on EEG features in movie clips with channel selection. Brain Inf 4(4):241–252
10. Badajena JC, Sethi S, Dash A, Rout P, Sahoo RK (2021) Smart wheelchair using brain waves through machine learning. In: 2nd international conference on machine learning, IoT and Bigdata (ICMIB-2021), intelligent system. ISBN 978-981-19-0900-9

Relationship LSTM Network for Prediction in Social Internet of Things

S. D. Mohana, S. P. Shiva Prakash, and Kiril Krinkin

Abstract Hyperscaling of IoT sensors leads to the connected device segments like smart phone, smart watches, smart home, smart city, and many more. These hyperscaling devices which are in a social relationship form a social IoT. The social relationship is based on object relationship in a smart environment. There are many challenges in social IoT, the major being object mobility, scalability, and pattern analysis in a smart environment. The smart objects have limited intelligence to take decision to predict the corresponding data. This work focuses on data of smart city environment to provide the services to the user in a given environment. The intelligence in the model using R-LSTM network is to determine the right data and predicting responding objects and relationship between the objects. The proposed work provides accuracy of 98.75% and loss of 0.37% to the SIoT smart environment.

Keywords SIoT-social internet of things · LSTM-long short-time memory network · Ml-machine learning · IoT-internet of things

1 Introduction

The Social Internet of Things (SIoT) is a network of interconnected heterogeneous or homogeneous objects with social relationships as in humans. The Internet of Things (IoT) are uniquely identifiable and provide data or information transferability without the interaction between human-to-computer or human-to-human. The social network of intelligent objects in SIoT is mapping between objects and the social network of humans. In SIoT, objects interact and behave like human relationships. SIoT has

S. D. Mohana · S. P. Shiva Prakash (✉)
Department of Information Science and Engineering, JSS Science and Technology University, Mysuru, Karnataka, India
e-mail: shivasp@jssstuniv.in

K. Krinkin
Department of Software Engineering and Computer Applications, Saint Petersburg Electrotechnical University "LETI", Saint Petersburg, Russia
e-mail: kirill@krinkin.com

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_11

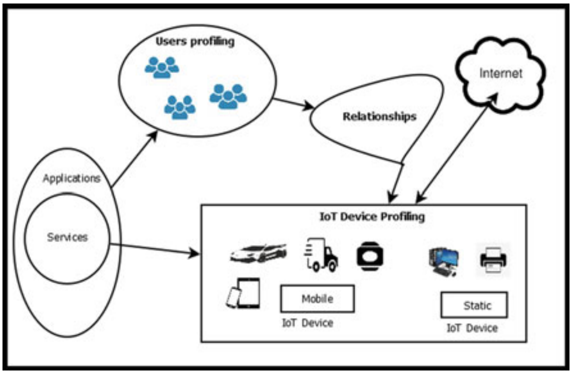


Fig. 1 SIoT environment

10 types of object relationships (O-R), namely parent-O-R, owner-O-R, guardian-O-R, social-O-R, sibling-O-R, guest-O-R, service-O-R, strange-O-R, co-location-O-R, and co-work-O-R. Every relationship has its own functionality to interact in the SIoT environment [1]. All these relationships will provide the reliable and efficiently to sharing resources and services. For example, IoT has heterogeneous objects (sensors, smart-phones, and actuators) in distinct features like OS (operating systems), software platforms, communication protocols, related standards, and independent identity. Thus, all these differences are ignored between communication in each devices. Thus, different objects cooperate effectively and securely to gratify end-users' desires such as reliability, safety, time, cost-efficiency, and availability. Social IoT provides a platform for interconnected objects to establish social relationship for common interest and provide services for users [2, 3].
The intelligent objects interact based on context of the users need. Hence in this work, proposed the context-based model using the artificial intelligence (Fig. 1).

This problem can be solved using long-term memory and different types of cells. Hence, memory cell is a black box; it has better performance at training and is faster like long-term dependencies that are detected in the data.

The organization of work is as follows: Sect. 2 related work, Sect. 3 problem statement, Sect. 4 provides system model and problem formulations, Sect. 5 introduced proposed relationship artificial neural network model and explains algorithm, Sect. 6 discusses on dataset, results, and Sect. 7 conclusion and future of the work.

2 Related Work

This section explains survey works of the SIoT. Han et al. worked on 5G/6G networks for smart city devices on ultra-reliable low-latency communications and massive machine-type communications for random access in a smart environment [4]. Jayaprakash et al. worked on LSTM for temperature models to New York City in USA [5]. Zhang et al. worked on Industrial Internet of Things (IIoT) to analyze equipment condition based on the sensor data using LSTM. Yao et al. worked on the predicting the wind speed using fuzzy rough-set in multi-dimensional LSTM [6]. Sundermeyer et al. worked on context of English and french language modeling using LSTM networks [7]. Zhao et al., worked on stock data using weighted LSTM to analyze the time series stock prediction [8]. Wang et al. worked on solar intensity using least absolute shrinkage and selection operator and LSTM for forecasting the linear relationship in data [9]. Liu et al. worked on abnormal detection in IoT stream data using LSTM model [10]. Mansour et al. worked on disease diagnosis model using cascaded LSTM for smart healthcare system [11]. Xia et al. worked on traffic flow prediction using window and normal distribution (WND)-LSTM for forecasting of traffic map reduce for large scale data [12]. Liu et al., worked on environmental water quality analysis using LSTM for water quality monitoring [13]. Shiva Prakash et al. worked on data aggregation in SIoT [14]. Shahab et al. worked on reviewing various SIoT applications [15]. Hence from the literature, it is found that no work that considers relationship as metric within LSTM for a SIoT-based applications.

3 Problem Statement

In the existing work, the services provided by the devices are recommended based on the user requirement within IoT. But in an SIoT environment, there is a need to consider the relationship between the devices, which possess certain challenges such as device mobility identification, change of relationship over a period of time due to the device mobility, etc. Thus, there is a need to develop a model that addresses the change of relationship between the devices. Hence, the work is on developing a relationship model based on device mobility that predicts the responding devices to provide services for a smart city environment application.

4 System Model

The SIoT system model S_M has various objects N which establish relationship R. The SIoT objects have homogeneous Object H_o and heterogeneous Object H_t in static S_t and mobility M_b nature. These independent objects D within the given set of networks N_x for the given set of objects in various distributions in a space D_{N_x} services S_e and applications A_{pp}. Thus, every individual object has applications, a set of services to get sensor data S_n through it. Hence, the random nature of distributions of objects in the system model is shown in Eq. (1).

$$S_M = \lim_{H_o, H_t} R \sum_{S_t}^{M_b} (D_{N_x}(D) : \forall(S, A)) = f\left(D_{N_x}, D(S, A), R(S_t + M_b)\right) \tag{1}$$

4.1 Problem Formulation

In SIoT, objects are distributed in randomly and obtain data from sensors. The application and services act as an interface to users from the objects. These objects have service with respect to time T for an application, to form a relationship among each object in SIoT.
Based on user–object interactions in a smart environment, an intelligent model is responsible for the rate of change of relationship and services. The knowledge model K_m considers activity of the user will change the relationship between the objects where it is indirectly proportional to the object service and applications. The objects form social relationships between the objects that come in contact in the given system network. So, data is analyzed through a knowledge model, hence the objective function as Eq. (2).

$$K_m = \lim_{H_o, H_t} D_{N_x} \left(\frac{1}{R(S_t + M_b)}, S, A, T \right) = f \left(\frac{A(S)}{R(S_t + M_b)} . T \right) \tag{2}$$

For the objective function can be defined as knowledge model (K_m), to find relationship (R) in network (D_{N_x}) of all the individual objects (D) for services (S) and applications (A) under the homogeneous or heterogeneous objects where in the static S_t and mobility M_b in nature, as shown in Eq. (3).

$$K_m = \lim_{n \to \infty} \sum_{n=H_o}^{H_t} f \left(\frac{A(S).T}{R(S_t + M_b)} \right) \tag{3}$$

subjected to applications and services $(A, S) = 1, 2, 3, \ldots, n$, devices D in static S_t and mobile M_b in nature $D = (S_t, M_b)$, the interactions with respect to relationships $R = 1, 2, 3, 4 \ldots 10$ and time $T = t_0, t_1, t_2, t_3, t_4 \ldots t_n$.

5 Proposed Model Design and Algorithm

The proposed service-oriented knowledge model is based on adding relationships between the user and objects. The relationship between the objects is based on user profiling in the context of SIoT applications. The proposed model investigates the relationship between a dependent (target) and independent (predictor) objects for forecasting services. It finds the rate of change of relationship between the objects (Fig. 2).

The LSTM model uses a hidden layer called the temporal loop and is connected through time. The short-term memory remembers previous observations and applies that knowledge in the going forward. Lower the gradient value, the process to update the weight is slower. To update the weights w with respect to time T is shown in Eq. (4).

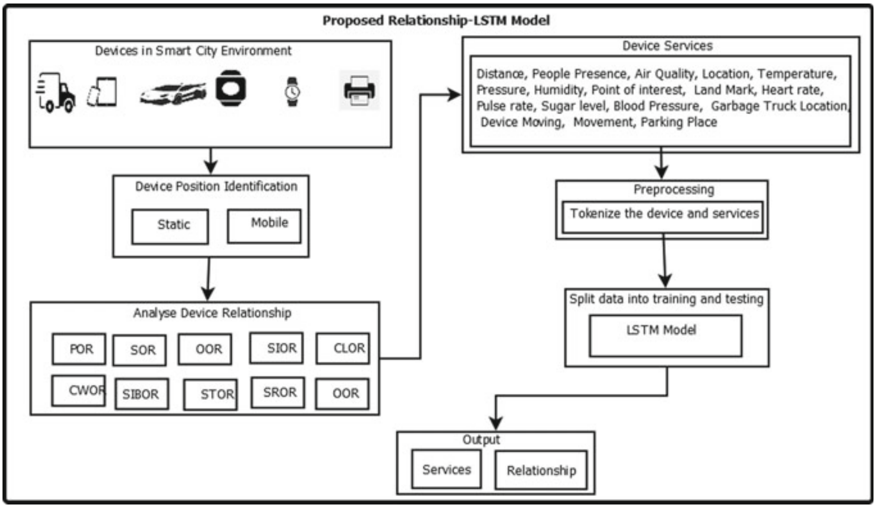


Fig. 2 Proposed model working mechanism

$$K_m = x.w_1+, \ldots, x.w_T. \tag{4}$$

The proposed LSTM Knowledge model is built using Eqs. (3) and 4 as shown in Eq. (5).

$$\begin{aligned} K_m = \frac{1}{R_s}((A_p(S_r))_1.w1 + (A_p(S_r))_2.w2 + \cdots \\ +(A_p(S_r))_{(t-1)}.w_(t-1) + (A_p(S_r))_T.w_n) \end{aligned} \tag{5}$$

LSTM cells manage two state vectors $h_s(t)$ and $c_s(t)$. The vector $h_s(t)$ as the short-term state and the vector $c_s(t)$ as the long-term state. According to the long-term state $c_s(t-1)$ traverses the network through a forget gate, dropping some memories in an order sequence of left to right, then in addition operation add new memories to short-term state $h_s(t-1)$. The result $c_s(t)$ is obtained at each time step in the output gate $g(t)$ through the tanh function. It produces the short-term state $h_s(t)$ which is equal to the cells output for time step $y(t)$. Hence, the LSTM gate controllers uses the logistic activation function through element-wise multiplication operations obtains range from 0 to 1. (Note: output gate 0s means close and output gate 1s means open.)

The process of cells in long-term state, short-term state, and also output at each time steps are shown in Eqs. (6), (7), (8), (9), (10), (11), and (12).

$$i_{(t_s)} = \sigma(Wgt^T_{(xi)}.x_t + Wgt^T_{(h_s i)}.h_s(t-1) + bs_i) \tag{6}$$

$$f_{(t_s)} = \sigma(Wgt^T_{(xf)}.x_t + Wgt^T_{(h_s f)}.h_s(t-1) + bs_f) \tag{7}$$

$$o_{(t_s)} = \sigma(Wgt^T_{(xo)}.x_t + Wgt^T_{(h_s o)}.h_s(t-1) + bs_o) \tag{8}$$

$$\text{(Memory cell candidate) } g_{(t)} = \tanh(Wgt^T_{(xg)}.x_t + Wgt^T_{(h_s g)}.h_s(t-1) + bs_g) \tag{9}$$

$$\text{(Memory cell) } c_s(t) = f_t \otimes c_s(t-1) + i_t \otimes g_t \tag{10}$$

$$\text{(Hidden state) } h_s(t) = o_t \otimes \tanh(c_s(t)) \tag{11}$$

$$\text{(Shadow state) } y_t = hs_t \tag{12}$$

where $Wgt_{(xi)}$, $Wgt_{(xf)}$, $Wgt_{(xo)}$ and $Wgt_{(xg)}$ are the weight for the input layer vector x_t. $W_{(hsi)}$, $Wgt_{(hsf)}$, $Wgt_{(hso)}$ and $Wgt_{(hsg)}$ are the previous short term state $hs_{(t-1)}$. The bs_i, bs_f, bs_o, and bs_g are the basis that initialize bs_f to a vector full of 1^s instead of 0^s.

Algorithm: R – LSTM network

```
While Objects in Network do
     identifyRelationship()
     Relationshipupdate()
     i= Tokenize()
  repeat
     for i:=max to min do
begin
     { LSTM Model }
end;
```

Characteristics of the proposed work as follows:

- The work is based on the relationships between the objects.
  - The relationships are established based on the distance between the objects.
  - The relationships are established based on the type of connection between the objects like private and public connections.
  - The relationships are established based on the type of protocols between the objects like Wi-Fi, Zigbee, Bluetooth.
  - The relationships are established based on the type of device types and device brands between the objects like the same brand and different brands.
- The work is based on the objects profiling in objects like owner ID, device ID, type of communication protocols available, type of connectivity like public device and private device.

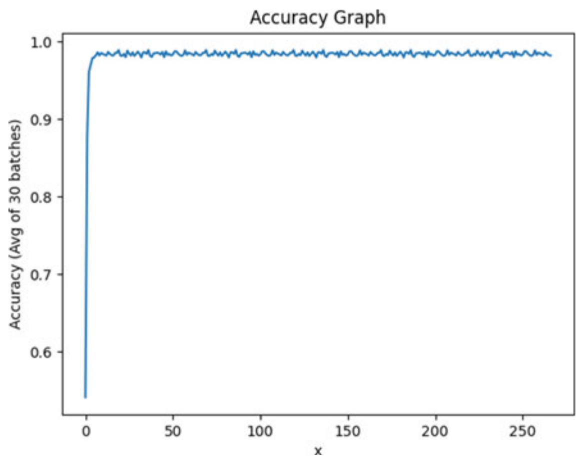


Fig. 3 R-LSTM accuracy graph

6 Results

The SIoT dataset contains 100,000 samples for public and private devices having both static and mobile devices. In private devices, there are few categories, namely smart phone (mobile), car (mobile), tablet (mobile), smart watch (mobile), and smart tbit (mobile). In public devices, there are few categories called data models namely weather, air quality, humidity, and temperature indicator. In private devices there are a few categories namely location, street light, parking and traffic monitoring. The proposed work is to predict the services using relationships between the objects using SIoT environment data. This experiment helps to find the relationship between the objects using semantic rule-based service using LSTM models. The LSTM is used to analyze the sensor data that is encoded with the service. The data is categorized into target and features; the services are target class, and remaining data are features corresponding to sensor values. These classes and features' data are split into two phases like training data of 70% and testing data of 30% obtained with an accuracy of 98.75% as shown in Fig. 3 and loss of 0.37% as shown in Fig. 4.

7 Conclusion and Future Work

The proposed work is carried out using relationships between the objects on the LSTM model. It predicts the time of forecasting services in terms of values each device obtains like closeness of classes in services like air quality, weather, temperature, tra c, people presence, and parking status within a smart city. The proposed R-LSTM-based service-oriented knowledge model uses the relationship between response device and requesting devices to predict the services to the users with accu-

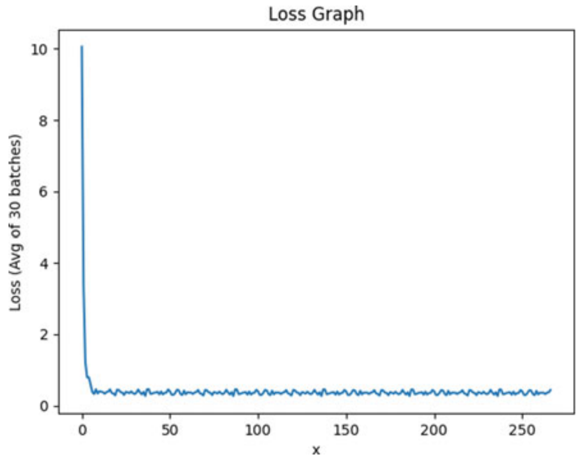


Fig. 4 R-LSTM loss graph

racy of 98.75% and loss of 0.37%. It requires an object profiling in every object otherwise relationship cannot be identified which is the only one limitation in this work. In future, analyze real-time objects using deep learning models.

Acknowledgements This work was carried out under the "Development program of ETU "LETI" within the framework of the program of strategic academic leadership" Priority-2030 No. 075-15-2021-1318 on 29 Sept 2021.

References

1. Kasnesis P, Toumanidis L, Kogias D, Patrikakis CZ, Venieris IS (2016) Assist: an agent-based SIoT simulator. In: 2016 IEEE 3rd world forum on internet of things (WF-IoT). IEEE, pp 353–358
2. Gupta H, Vahid Dastjerdi A, Ghosh SK, Buyya R (2017) iFogSim: a toolkit for modeling and simulation of resource management techniques in the internet of things, edge and Fog computing environments. Softw Pract Exp 47(9):1275–1296
3. Han SN, Lee GM, Crespi N, Heo K, Van Luong N, Brut M, Gatellier P (2014) DPWSim: a simulation toolkit for IoT applications using devices profile for web services. In: 2014 IEEE world forum on internet of things (WF-IoT). IEEE, pp 544–547
4. Han H, Zhao J, Zhai W, Xiong Z, Lu W (2021) Smart city enabled by 5G/6G networks: an intelligent hybrid random access scheme. arXiv preprint arXiv:2101.06421
5. Jayaprakash C (2021) Deep learning for improved heat index using IoT-based data. Turk J Comput Math Educ (TURCOMAT) 12(11):797–806
6. Zhang W, Guo W, Liu X, Liu Y, Zhou J, Li B, Yang S (2018) LSTM-based analysis of industrial IoT equipment. IEEE Access 6:23551–23560
7. Sundermeyer M, Ney H, Schlüter R (2015) From feedforward to recurrent LSTM neural networks for language modeling. IEEE/ACM Trans Audio Speech Lang Process 23(3):517–529
8. Zhao Z, Rao R, Tu S, Shi J (2017) Time-weighted LSTM model with redefined labeling for stock trend prediction. In: 2017 IEEE 29th international conference on tools with artificial intelligence (ICTAI). IEEE, pp 1210-1217

9. Wang Y, Shen Y, Mao S, Chen X, Zou H (2018) LASSO and LSTM integrated temporal model for short-term solar intensity forecasting. IEEE Int Things J 6(2):2933–2944
10. Liu J, Bai J, Li H, Sun B (2021) Improved LSTM-based abnormal stream data detection and correction system for internet of things. IEEE Trans Ind Inf 18(2):1282–1290
11. Mansour RF, El Amraoui A, Nouaouri I, Díaz VG, Gupta D, Kumar S (2021) Artificial intelligence and internet of things enabled disease diagnosis model for smart healthcare systems. IEEE Access 9:45137–45146
12. Xia D, Zhang M, Yan X, Bai Y, Zheng Y, Li Y, Li H (2021) A distributed WND-LSTM model on MapReduce for short-term traffic flow prediction. Neural Comput Appl 33(7):2393–2410
13. Chen Z, Xu H, Jiang P, Yu S, Lin G, Bychkov I, Liu Z (2021) A transfer learning-based LSTM strategy for imputing large-scale consecutive missing data and its application in a water quality prediction system. J Hydrol 602:126573
14. Meghana J, Hanumanthappa J, Shiva Prakash SP (2021) Performance comparison of machine learning algorithms for data aggregation in social internet of things. Glob Trans Proc 2(2):212–219
15. Shahab S, Agarwal P, Mufti T, Obaid AJ (2022) SIoT (social internet of things): a review. In: ICT analysis and applications, pp 289–297

Relationship-Based AES Security Model for Social Internet of Things

K. S. Santhosh Kumar, J. Hanumanthappa, S. P. Shiva Prakash, and Kirill Krinkin

Abstract Social Internet of Things is the rising contemporary of computer science and information technology. It combines the concept of a social network with the Internet of Things and improves distributed object discovery, selection, composition of services and information. Searching for services in the SIoT is based on the use of the social structure formed by the generation of relationships between the devices. Since SIoT has wide applications, security will have the greatest impact on IoT adoption. However, SIoT lacks a security model for its applications, where each device generates data. Thus in this work, a security model is proposed that considers the relationship between devices and generates relationship keys. The standard 256-bit Advanced Encryption Standard algorithm is implemented along with a relationship key to perform encryption and decryption on the data generated.

Keywords Social internet of things · Security · AES · Encryption · Decryption

1 Introduction

Social Internet of Things (SIoT) allows IoT objects to form a relationship with other independent devices profiled by the owners. SIoT incorporates intelligent IoT object's social behavior and allows them to create their own social network. It has numerous advantages. Firstly, SIoT fosters resource availability and enhances service discovery in a distributed manner [1, 2]. Secondly, the centralized method of searching for

K. S. Santhosh Kumar · J. Hanumanthappa
Department of Studies in Computer Science, University of Mysore, Mysuru, Karnataka, India
e-mail: santhosh@compsci.uni-mysore.ac.in

S. P. Shiva Prakash (✉)
Department of Information Science and Engineering, JSS Science and Technology University, Mysuru, Karnataka, India
e-mail: shivasp@jssstuniv.in

K. Krinkin
Saint Petersburg Electrotechnical University "LETI", Saint Petersburg, Russia
e-mail: kirill@krinkin.com

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_12

IoT objects creates a scalability problem, which SIoT solves by allowing each IoT object to navigate the SIoT network structure in a distributed manner to reach other objects. Third, things can inquire about the reputation of other things in the local neighborhood based on the social structure established among IoT objects [3, 4]. It includes a communication layer, an application layer, and a security layer. The components are shown above and discussed in the security framework and security algorithms [5]. As an example, the smart city concept has a huge challenge for IoT and the group of a social network called SIoT (Fig. 1).

The rest of the work is organized as follows: Sect. 1 Introduces the SIoT Architecture, the related work carried out is discussed in Sect. 2. Section 3 defines the problem Statement, Sect. 4 represents the System Model, the Proposed Design and Algorithm is presented in Sect. 5. Section 6 discuses the Results, Conclusion and Future work is carried out in Sect. 7.

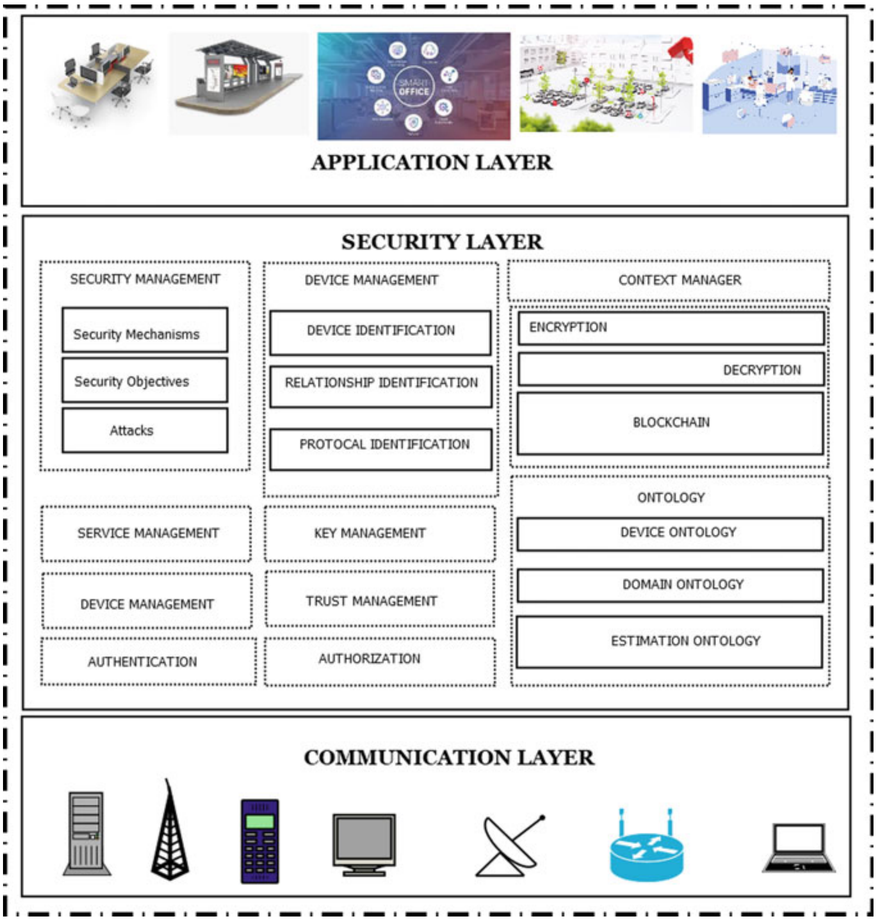


Fig. 1 SIoT security-based layer architecture

2 Related Works

This section presents the previous work carried out for the development of security models for IoT devices and also enhances the relationship-based security for SIoT, Fenz et al. [6] and a team of researchers have developed a way to increase the level of automation and cost-effectiveness of the information security compliance checking process. The technique can be applied to ISO 27002 information security procedures as well as other industry standard codes of practice, such as those issued by the US Department of Defense. Shaaban et al. proposed an ontology-based framework developed as a revolutionary automated approach for checking and validating security requirements [7]. Ekelhart et al. [8] invented a security ontology that can assist SMEs in establishing a firm foundation for adopting an IT security strategy that allows for low-cost risk management and analysis. Ramanauskait et al. [9] introduced a new security model that provides greater coverage of security standards than existing ontology and has superior branching and depth qualities for ontology visualization. factors, as well as the ability to address various security requirements of existing security ontologyOnto Metric factors and a new Security Model for Data Encryption Standard (CS5). Security patterns are created by mapping security knowledge from Guan et al. [10] Ontology written in the Web Ontology Language (OWL) were generated and then integrated into a security pattern search engine. Hachem et al. [11] concentrate on modeling a collection of ontology that define devices and their functions, as well as modeling the physical domain properly. Security rookies may reuse their security abilities to construct secure software systems using the same approach. Pereira et al. [12] present as it allows for the approximation and estimation of functions that are frequently offered by things, A security domain ontology's conceptual implementation model. Pedro et al. [13] proposed the use of an ontology to codify and express security information ideas that put current processes and procedures to the test. It uses a bottom-up approach to data security, favoring implicit knowledge representation over explicit knowledge representation. Simperl et al. [14] proposed a framework that enables the creation of many types of knowledge structures, such as ontology, taxonomies and folksonomies, as well as information management systems that use them. Rahman et al. [15] implemented the framework. The SN-IIoT's SPY-BOT system uses a hybrid data extraction approach to perform post-filtering that balances users' behavior polarity. Hussain et al. [16] proposed a security architecture, where CoAP is responsible for delivering an end-to-end security solution and operates alongside HTTP. Neisse et al. [17] explain the security toolbox based on models incorporated into an IoT device management framework. It aids in the specification and efficient assessment of security. Among those who have contributed to this work are Wang et al. [18]. The framework is to illustrate how it may be used in a Smart City, its viability and efficiency. By examining social trust, it offers secure content sharing.

3 Problem Statement

In the existing work, it can be found that the data generated by the device can be manipulated by an intruder at any given instant of time in an SIoT application. There is a need to secure this data that is used in information sharing between the devices. Hence, this work focuses on designing security models and proposing algorithms that secure SIoT device data using relationships as a key factor.

4 System Model

System Model S_m having objects $O = 0_1, 0_2 \ldots 0_n$ in a SIOT environment, it establishes a connection between the objects in a network N_x. The connected objects share the data D, based on 10 types of relationships $R = (R1, R2, \ldots R10)$ between the objects. Consider the system model S_m having the set of objects sharing information through security S, and it will interact through the relationship defined in Eq. (1).

$$S_m = \sum O\left\{N_x(S, D)|R : \forall N_x(D, R) = f(N_x, S, D, R)\right\} \tag{1}$$

4.1 Problem Formulation

Consider a network N_x is connected while sharing information through relationship R, the objective function is secured the data D while sharing between the objects O. Thus objective function can be defined as shown in Eq. (2).

$\forall N = N_x \in (D, R)$

$$\Longrightarrow \sum_{1}^{n} f(S, D, R) \tag{2}$$

$\forall N$, subjected to

$$S \in N_x(O) \tag{3}$$

where, $f(N_x)$ is connected between the object O, it shares the information based on the relationships between the objects shown in Eq. (3). Therefore the data is shared through relationships using security in the objects. The proposed solution is in Eq. (4).

$$\text{key (Nk)} = \text{Key Size}/32 \tag{4}$$

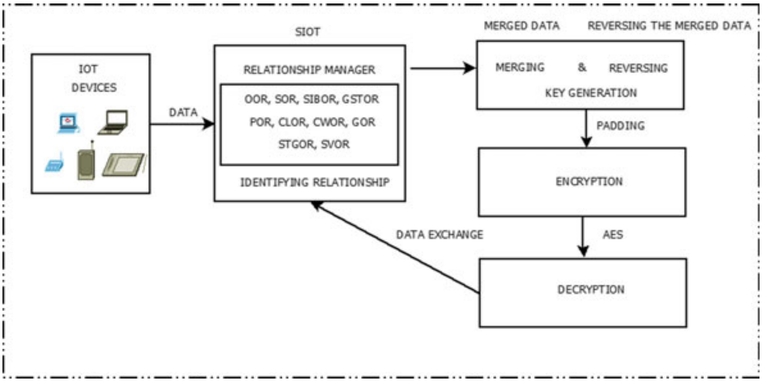


Fig. 2 Proposed model design

5 Proposed Security Model Design and Algorithm

The proposed model is based on encryption and decryption of objects data through relationships established between the objects. The working principle of the model is shown in Fig. 2. It contains the data generated by the SIoT objects that are encrypted by using the relationship between the devices as a key. Thus resulting in a 256-bit AES algorithm incorporated into the proposed security model. The keys are generated by reversing the relationship type using a transposition method that helps to provide more security to the SIoT data. Further these data can be decrypted by using the same key generated between the corresponding devices. The Advanced Encryption Standard key generation, encryption, and decryption methods for securing SIoT device data using relationships as a parameter to generate the key is as shown in Algorithm 1.

Algorithm : R=AES
Begin :
for (**Var** X)
If (X = rel_type.Reverse( )) **then**
As=[ ]
end;
For(each **char** **in** X)
If(As_Vs append(ord(**char**) $+$k)) **then**
end;
For(each Vs **in** (As_Vs) **Do**
convert each As_Vs **to** **char** ;
join each **char** **to** FS;
end ;
If(FS= G.ky) **then**
return ;
end ;

If(NBp = Bs = 1.en(Pt)Bs **and**
(As = chr(NBp)) **then**
(Ps = NoBp(As))
end ;
If((P_Pt =Pt + Psr) **and**
(Pt = P_pt)
then(iv = Rd.n( ).R(Bs))
end ;
If((Cp = AES.n(S.ky, AES.M_cbc, iv)) **and**
(En . t = Cp.En(Pt.Enc( ))) **then**
(b64enc (iv + En.t).decode(" ut f =8"))
return;
end;
end;

In Fig. 3, ontological model for an AES Encryption and Decryption Method, Encrypt data produced by SIoT devices in an environment from a dataset to identifiable data based on data type, relationship type, and device type using AES Encryption and Decryption Method.

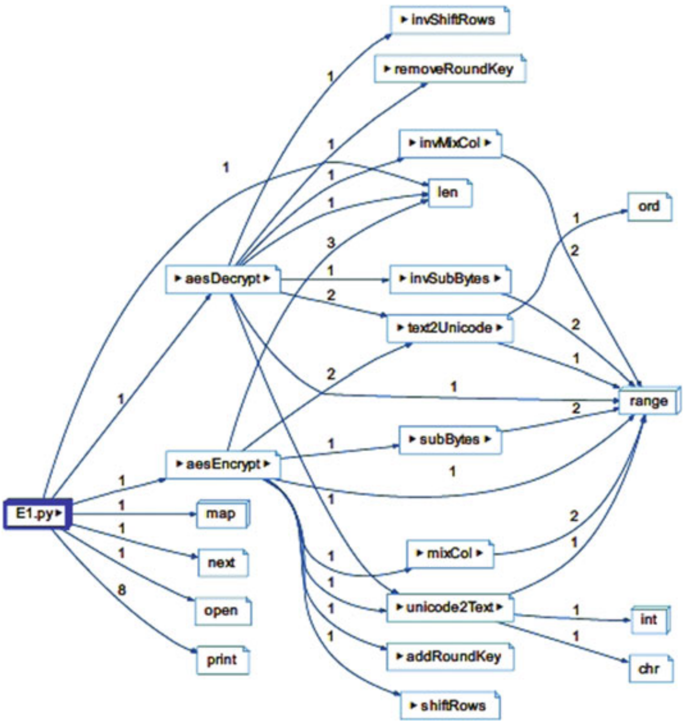


Fig. 3 Proposed model design

Table 1 Sample dataset

Source_obj_id	Relationship	Desti_obj_id	Trustworthy	Message
car_1	OOR	phone_2	1	Message transmitting
car_2	SOR	phone_1	1	Connection established
car_31	POR	car_13	1	Sharing the data
tv_1	CLOR	laptop_1	0	Received acknowledgement
tv_2	CLOR	laptop_2	1	Retry sending data
phone_1	OOR	housealarm_1	1	Data sent
phone_2	SOR	car_2	1	Off mode of car
phone_3	CWOR	phone_5	1	Start charging in 5 min

Table 2 Modified AES encryption and decryption result

Message before encryption	Key	Inverting key
Connection established	car_2SORphone_11	11_enohpROS2_rac
Message transmitting	car_100Rphone_21	12_enohpR001_rac
Sharing the data	car_31PORcar_131	131_racROP13_rac
Received acknowledgement	tv_1CLORlaptop_10	01_potpalROLC1_vt
Retry sending data	tv_2CLORLaptop_21	12_potpalROLC2_vt
Data sent	phone_100Rhousealarm_11	11_mralaesuohR001_enohp
Off mode of car	phone_2SORcar_21	12_racROS2_enohp
Encrpyted text	Decrpyted text	
Dóbka ñ ZûDdéób.%b“, o t £ÁCŨaÚ860ÈÊ	Connection established	
N, DUDÁ-00082006«| BòÇE DUBL60ÈÊ	Message Transmitting	
d]îy© öÊË-Ù»]Oh%a,	Sharing the data	
SÒP0h0jøc Da Dêd00+È00"? DÅûæåêDöß	Received acknowledgement	
RÑÆDD /Ö70v¥ ÖlíÖÙ Æ ÝÒÈÅûæåêDöß	Retry sending data	
Dîy200• ÈÌÚ: ÆÁūææ	Data sent	
nÑOÛ£ya döşçiy	Off mode of car	

6 Results and Discussion

The dataset contains object_id, relationship type, and destination object, as well as the trust factor, as shown in the sample dataset tabulated in Table 1. It has 100 000 samples with public object information and private object information such as tablets, phone, fitbit, car, alarm, laptop, door sensor, weather data models and air quality data models as shown in Table 2. It takes into account the source object id, relationship type, and destination object, as well as the trust factor.

The activity is to use the AES encryption and decryption process, with the results shown in Table 2.

7 Conclusion and Future Work

In this work, the issues related to SIoT data security are addressed by proposing a security model. It uses the relationship between the devices as a key for securing the SIoT device data within a 256-bit AES algorithm. The model is implemented on a SIoT dataset and tested by exchanging information between the sender and the receiver devices. The encrypted and decrypted data was found to be secured for a SIoT application. In the future, the model will be enhanced and tested by considering several types of attacks on the devices.

References

1. Tariqa N, Khanb FA, Asim M (2021) Security challenges and requirements for smart Internet of things applications: a comprehensive analysis. Proc Comput Sci 191:425–430 , 1877-0509
2. Bugeja J, Jacobsson A, Davidsson P (2021) PRASH: a framework for privacy risk analysis of smart homes. Sensors 21:6399
3. Logrippo L (2021) Multi-level models for data security in networks and in the Internet of things. J Inf Secur Appl 58:102778, 2214-2126
4. Malhotra P, Singh Y, Anand P, Bangotra DK, Singh PK, Hong W-C (1809) Internet of things: evolution, concerns and security challenges. Sensors 2021:21
5. Hughes-Lartey K, Li M, Botchey FE et al (2021) Human factor, a critical weak point in the information security of an organization's Internet of things. Elsevier, 2405-8440
6. Fenz S, Neubauer T et al (2018) Ontology-based information security compliance determination and control selection on the example of ISO 27002. Inf Comput Secur 26(5):551–567
7. Shaaban AM, Schmittner C, Gruber T, Mohamed AB, Quirchmayr G, Schikuta E et al (2020) Automated ontology-based security requirements identification for the vehicular domain. J Data Intell 1(4):401418
8. Ekelhart A, Fenz S, Klemen M, Weippl E et al (2007) Security ontologies: improving quantitative risk analysis. In: 40th annual Hawaii international conference on system sciences. https://doi.org/10.1109/HICSS.2007.478
9. Ramanauskaitėce S, Olifer D, Goranin N, Čenys A et al (2013) Security ontology for adaptive mapping of security standards. Int J Comput Commun 8(6):878–890, ISSN 1841-9836
10. Guan H, Yang H, Wang J et al (2016) An ontology-based approach to security pattern selection. Int J Autom Comput 13(2):168–182
11. Hachem S, Teixeira T, Issarny V et al (2011) Ontologies for the Internet of things. In: European community's seventh framework programme FP7/2007-2013 under grant by HAL hal-0064219, agreement number 257178
12. Pereira T, Santos H et al (2009) An ontology based approach to information security. In: MTSR 2009, CCIS 46. Springer, Berlin, pp 183–192
13. Gonzalez-Gil P, Martinez JA, Skarmeta AF et al (2020) Lightweight data-security ontology for IoT. Sensors 2020(20):801. https://doi.org/10.3390/s20030801
14. Simperl Elena, Bürger Tobias, Hangl Simon, Wörgl Stephan, Popovd Igor et al (2012) ONTOCOM: a reliable cost estimation method for ontology development projects. Sci Serv Agents World Wide Web 16:1–16
15. Rahman MA, Zaman N, Asyhari AT, Sadat SMN, Pillai P, Arshah RA (2021) SPY-BOT: machine learning-enabled post filtering for social network-integrated industrial Internet of things. Ad Hoc Netw 121:102588, 1570-8705
16. Hussain A, Ali T, Althobiani F, Draz U, Irfan M, Yasin S, Shafiq S, Safdar Z, Glowacz A, Nowakowski G et al (2021) Security framework for IoT based real-time health applications. Electronics 10:719

17. Neisse R, Fovino IN, Baldini G, Stavroulaki V, Vlacheas P, Giaffreda R (2014) A model-based security toolkit for the Internet of things. In: 2014 9th international conference on availability, reliability and security. IEEE, 978-1-4799-4223-7
18. Wang B, Sun Y, Duong TQ, Nguyen LD, Zhao N (2020) Security enhanced content sharing in social IoT: a directed hypergraph-based learning scheme. IEEE Trans Veh Technol 4412–4425

Intelligent Systems in Image Processing and Deep Learning

Face and Fingerprint Fusion Using Deep Learning

Saloni Ekal, Kunjal Wadke, Md. Altamash, and Rupali Kute

Abstract Biometrics recognition for individuals has been flourishing over the past few years, with the introduction of these systems in almost every sector of various industries. It makes keeping a track of large numbers of people easier at the administration level, eliminating the requirement of manual checks for every individual. Biometrics traits are individual characteristics like fingerprint, face, voice, etc., on the basis of which individuals are identified. Biometric technology and its use is not only limited to its application as a verification tool but also goes beyond that. Due to its secure, powerful and distinctive capabilities, its applications can also be extended as credentials tools and also digital authentication measures. The growth and pace of development in biometrics technology have been tremendous over the past few years and is considered to grow faster in the coming decades. The framework of the proposed model consists of performing feature-level fusion with the dataset used, and the second half consists of using various deep learning models to classify the labels correctly according to the original dataset.

Keywords Feature fusion · VGG16 · VGG19 · ResNet · Random forest · KNN · Decision tree

1 Introduction

Deep learning has been flourishing at a rapid pace for a few years now. It has been used on various facial image datasets and fingerprint datasets for biometric recognition, and with implementation of various deep learning models, it can be used in various advanced security systems for biometric recognition with multiple biometric characteristics.

S. Ekal · K. Wadke (✉) · Md. Altamash · R. Kute
School of Electronics and Communication Engineering, MIT WPU Kothrud, Pune, Maharashtra 411038, India
e-mail: kunjalwadke8888@gmail.com

R. Kute
e-mail: rupali.kute@mitwpu.edu.in

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_13

Biometrics technology is a way of establishing the identity of a person through automated techniques like deep learning models based on various behavioral traits like voice, signature or physical traits like face and fingerprint. This technology has been an active research topic which proposes various solutions to the modern day problems in the domains of authentication, identification, security and administrations systems. Biometrics are widely being used in various organizations, companies and educational institutes for individual identification where a person declares his identity so his recognition becomes a verification process when the data acquired by the sensor or input system in real time is matched with the data in the database.

Biometric identification is proving to be a lot more advantageous compared to the manual identification systems, as they are less prone to errors, improves the overall user experience and also is very authentic as there is no risk of loss. It has also gained popularity in the recent times due to digital security which is widely popularizing due to personal devices like smartphones, smart watches and other personal devices used by people which have an integral role played by biometrics.

Recent research and advancements in the field of deep learning like feature-level fusion, score-level fusion which accelerates the process of identification and reduces the room for error. Biometric recognition is not only used for identification and authentication but has also flourished in other domains of security protocols, law enforcement, health care and airport security.

2 Literature Survey

In recent times, the domain of feature level fusion has drawn a lot of attention with cutting edge results and its capability to process various levels of fusion. A lot of researchers have been doing their research in the field of both unimodal and multimodal biometric systems [1, p. 2]. Unimodal biometric systems are the traditional way of identification in use for a long time but the recent work in multimodal biometric systems is something that led us to choose this topic of research and implement it using deep learning. Multimodal systems are widely under research over the traditional unimodal systems because their combined scores give better insights over the individual scores by the latter.

These systems also have the ability to process multiple aspects of information to give a collaborative result involving all the input parameters of the biometric systems.

Multimodal biometric systems have a brighter future in the security sector because of its ability to process multiple characteristics of a person's identity which reduces the risk of error and increases the security in the identification process. The spectrum for development and implementation in this domain is vast which will bring in immense opportunities to be extended beyond just the biometric systems in commercial places. Various approaches to fusion have been in use like score-level fusion, sensor-level fusion and feature-level fusion. In the proposed system, we have implemented feature-level fusion, where the biometric system implements individual feature extraction of fingerprint and face datasets and then performs feature-level fusion of these features. The resultant classifier in our system classifies the correct identity of the person to whom the face and fingerprint biometric belong.

The approach proposed in this project aims at reducing errors in biometric recognition systems with multi-characteristic recognition. The identity of a person is identified with a two-stage recognition of different biometric, in this case, fingerprint and face. This makes the system more advanced and secure as even if one of the biometric characteristics fails to identify, that does not affects the overall failure of the system. It has the capability to be used in various educational institutes with a lot of students, organizations, companies with many employees and in advanced secure systems.

3 Recent Works

First multimodal biometric was proposed by Jain and Ross. Later, many scientists thrive for plenty of research in multimodal biometric systems. A great deal of academic research was devoted to it.

Csurka et al. used the Harris affine region detector to identify the key points in the images which are then described by SIFT descriptors. The proposed framework was mainly evaluated on the Xerox7 image set. The overall classification rate is reported to be 85% using linear SVMs [2, p. 2].

Chatfield et al. proposed different deep architectures: CNN-F, CNN-M and CNN-S, along with a comparison with previous state-of-the-art shallow representations such as the bag-of-features and the improved Fisher vector.

The experimental results are achieved using a linear SVM classifier applied to a feature representation of size 4096 extracted from a layer in the net. The authors strongly suggest that features obtained from deep learning with convolutional nets should be the primary candidate in most visual recognition tasks.

4 Methodology

Studies show that the use of deep learning technique helps us to achieve more precise models, and deep learning is among the novel detection techniques. Deep learning can be improved by adding different types of techniques like transfer learning, knowledge distillation, quantization, etc., among all these, we are using transfer learning as it helps us to enhance the accuracy just by training so layers and also this helps us to reduce our training time. The fusion technique is used in our experiment as we would be getting the fused vector of a person's finger and face features which will eventually add more security to our system of detection (Fig. 1).

Fetch the raw data: In the research work having the experimental dataset with variations plays vital role to have an advance and diverse results. In our research

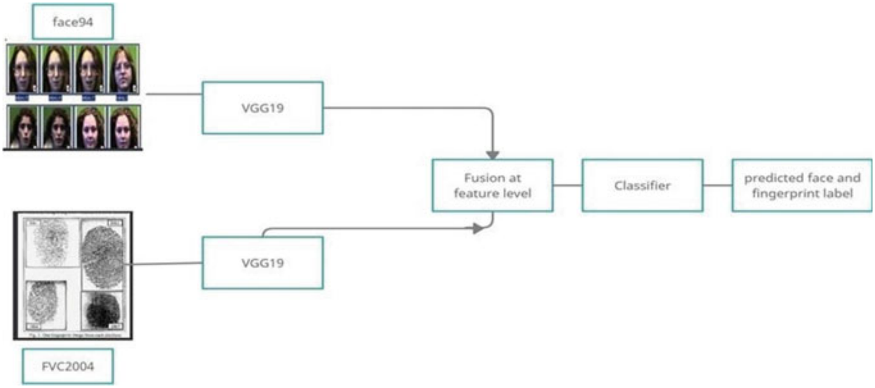


Fig. 1 Architecture of our experiment

work, we tried our hands on various dataset. Firstly, we tried our model on SOCOFing dataset for fingerprint and LFW for faces.

SOCOFing—This dataset contains 6000 fingerprint images from 600 people. This has functionality such as labels for genders, hand, finger name and much more attributes [3].

LFW—Label Faces Wild dataset is a frontal face dataset which was collected by University of Massachusetts which has a total of 13,233 images of 5749 subjects [4].

Our models failed on these datasets as our model needed the equal number of faces and fingerprints inside the folder of 1 single person. Hence, we shifted to the relevant dataset such as Face 94 for faces and FVC2004 for fingers.

Face dataset: The Face 94 dataset is from Libor Spacek's Facial Images Database is a frontal face database which had 153 subjects faces. Similar to Face 94, we tried our models on face 95 and face 96 [5, p. 3] (Table 1).

Fingerprint dataset: FVC2000 dataset came from the fingerprint competition which was held in 2000. 4 different directories, namely (DB1, DB2, DB3 and DB4) were collected by using the following sensors/technologies: Optical sensor, capacitive sensor and synthetic generator.

Similar to FVC2000, we tried our models on FVC2002 and FVC2004, and to use a final fingerprint dataset, we combined all 3 variants of FVC so that we can have a diversified dataset and can have robust outcomes [6].

Table 1 Face dataset comparison

Dataset	Subjects	Resolution
Face 96	152	196 × 196 pixel
Face 95	72	180 × 200 pixel
Face 94	156	180 × 200 pixel

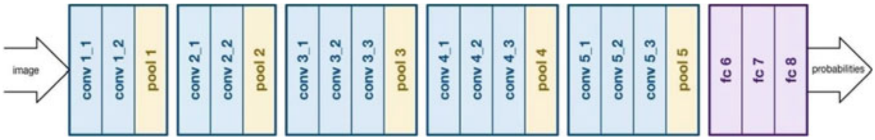


Fig. 2 Architecture of VGG19

4.1 Vggnet

VGG [7] is a convolutional neural network (CNN) architecture for image classification using deep learning [7].

Why VGG? It is currently the most preferred choice in the community for extracting features from images. It is considered to be one of the excellent vision model architectures till date. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. The weight configuration of the VGGNet is publicly available and has been used in many other applications and challenges as a baseline feature extractor.

The architecture is similar to the AlexNet. In this, it is essentially a network with 16 layers. The larger version is VGG19 which is a variant of that with 19 layers network architecture and it is something easy to implement. Architecture is a very large number of channels, i.e., 512. There are 496 base units but these are real large so if you have multiple of these add up right if you are using multiple of them. Each channel is essentially created by a different kernel, and each channel will learn different type of information because each channel is created by the same kernel. We use the same convolution to maintain the size and then he unread the max pooling layers to reduce the size by half and then we have yet another round of squeezing and so forth and you can see we are increasing the number of channels so the width increases, whereas, the height and width of the feature map decline. We are using the first layer of VGGNet, i.e., feature extraction layer to get the features of data, and we take the feature of data in multiple filters and create a feature map to better train a model (Fig. 2).

4.2 ResNet

Convolutional neural network is a residual neural network [8]. The idea behind the residual is that it is really hard to find the direct mapping from the input to the actual output where in the residual like the residual here. Instead we are trying to find difference between the input and the output and we can do that with a residual block over to the left. The main idea behind creating this residual neural network here is that we can stack more layers on top of each other or more of these residual blocks here on over top of the other, where in standard neural networks, if we just continue adding layers, we will just add more complexity and it will just take up too much performance if you are just stacking layer by layer but when we are using neural networks we can actually stack more. These residual blocks here are actually able to go deeper in the only way to create more complex and deeper neural networks without even like increasing that much in performance are like decreasing that much. The number here is actually just the number of residual blocks where one residual block is the block that we just went over in the previous line. We are able to train a new network with one thousand residual blocks stacked on top of each other. We are actually able to create a more complex neural network that can solve more complex tasks.

Feature level fusion: After the feature extraction through ResNet [8].

VGG16 andVGG19 [7] model, the obtain feature vectors were fused to have the fusion at feature level. Feature fusion is the technique in which we combine 2 or more feature vectors to get a single feature vector. Here, in our research, we combined the vectors of face and fingerprint features together so that for the single subject, we can have the single vectorized data of his finger and face combined. Due to this fusion, we can better recognition results. There are fusion with different techniques in biometric eg-fusion at raw data level (Sensor fusion), Decision level fusion, after understanding the need of our research we found feature level fusion to be best for our implementation.

Classifier model: Once we had our feature-level fusion, we put our fused vectors in different classifier models to check the accuracy and to get the perfect model for our experiment. The models on which we tried our hands on are as follows.

4.3 Random Forest

It creates decision trees on given data and the data passed into a number of weak models and trains these models to get the output from each model, then gets the prediction from each of them and finally selects the best solution by means of voting.

4.4 Decision Tree

The dataset will be made up of attributes (every now and then refer to as facial characteristics or typical features) and a class attribute. Other terms for class attribute would be target or outcome. This attribute is what we would like to predict.

4.5 KNN

K-nearest neighbors need supervised data to train on. With that data, KNN [9] can classify new, unidentified data by analysis of the closest data points. Thus, the variable K is taken to consider the number of neighbors you need to predict the unidentified data by majority voting.

5 Experimental Results

In our experiment, we tried our models on 3 different face dataset and a combine dataset which was the combination of 3 fingerprint datasets as mentioned earlier. For more variations in our results, we tried our datasets on 3 different classifiers, namely random forest, decision tree and KNN which were combined with the permutation and combination of 3 models like VGG16, VGG19 and ResNet with datasets the following are the results of our experiment.

5.1 VGG16

VGG16 gets its best accuracy with Face 96 for random forest with the accuracy of 95.67% as, when we compare our accuracies with KNN and decision tree, we have edge in random forest as it builds up many trees as compare to decision tree and it thus gives us a stable result as compare to decision tree as well as lazy algorithm KNN.

To find the optimal K value, we plotted the graph of K versus $F1$ score which gave us the solution as best k as $1k$ value and $3K$ value. We can use these findings to make a powerful effect on KNN performance (Tables 2 and 3; Fig. 3).

Table 2 Accuracy comparison of VGG16

	Face 94	Face 95	Face 96
Random forest	94.82	92.43	95.67
KNN	74	72.03	75.58
Decision tree	83.6	81	86

Table 3 Accuracy comparison of VGG19

	Face 94	Face 95	Face 96
Random forest	95.05	93	97.53
KNN	70	69.88	70.88
Decision tree	93	92.48	94.48

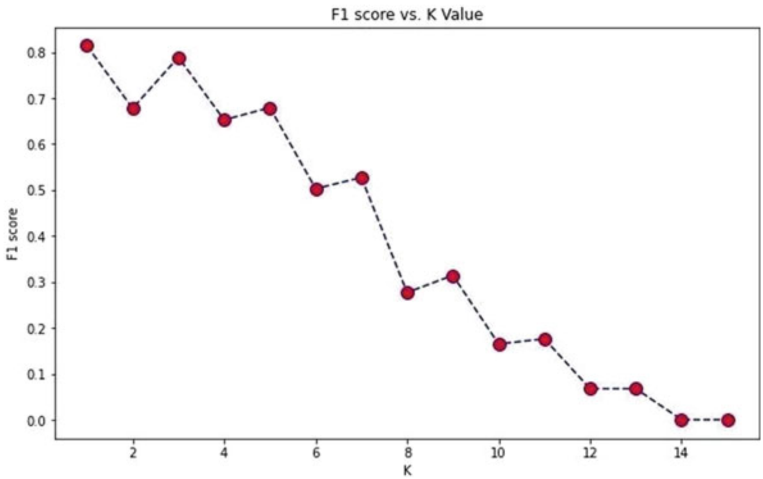


Fig. 3 VGG16 on KNN (for *K* value optimization)

5.2 VGG19

VGG19 acquired its highest accuracy with Face 96 which due to increase number of subjects and adequate pixels, it has the accuracy of 97.53 with random forest, 70.88 with KNN, decision tree has the accuracy of 94.48 as it is the minor version of random forest. The best performance of decision tree is acquire in this model.

In our research, we are getting a training set score of 95%, and cross-validation score is around 80% against the accuracy score and number of trees (Fig. 4; Table 4).

5.3 ResNet

ResNet model is used for the feature extraction and the pooling within is used to reduce the dimensions. ResNet is considered faster when compare to VGG, and in our experiment, we got our model's best accuracy with ResNet while trying it on Face 96 (98.28%). KNN has shown its highest accuracy in this model too (76%). Hence, by analyzing the experiment, we may comment that ResNet is considered best model for face and finger fusion in deep learning.

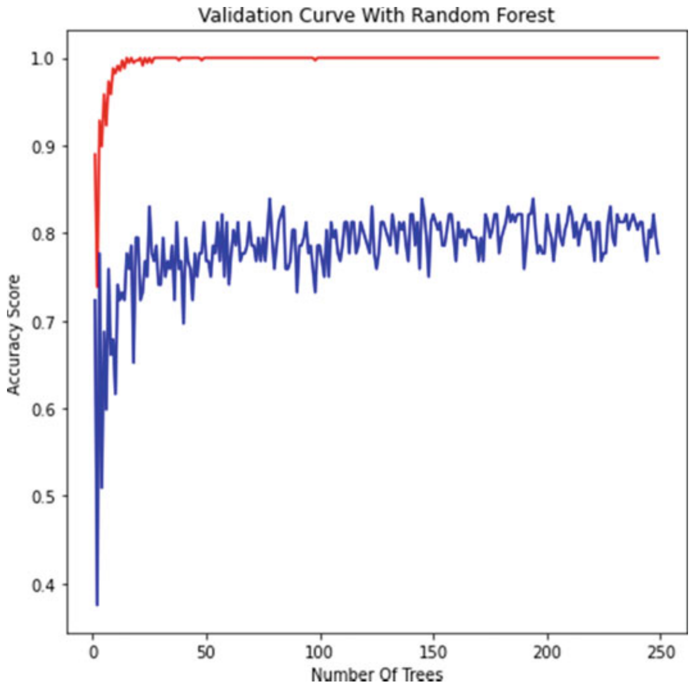


Fig. 4 VGG19 on random forest (accuracy score vs. number of trees)

Table 4 Accuracy comparison of ResNet

	Face 94	Face 95	Face 96
Random forest	97.06	96.79	98.28
KNN	75	74.68	76
Decision tree	89.28	89.94	91

6 Conclusion

In our work, we have implemented and evaluated performance of multimodal biometric systems using various methods and deep learning models. For the sake of reducing complications and errors, we have also worked out various techniques like converting all the images to gray scale images which made sure the dimension of the images was the same. The features for both the face and fingerprint dataset were extracted. Finally, the extracted features were successfully merged which is an important and highlighting part of the project. Experimental results of all the different classification models have showcased our testing process which evaluates the accuracy and the performance of each model.

This project can be extended to various different datasets and also various other deep learning models can be implemented too. The proposed idea of research has the potential to be implemented in real-time biometric systems and has scope for future use. It has the ability to revolutionize the future of biometrics with its multi-level recognition which reduces the error and chances of failure. These systems can be deployed where high security is of greatest priority and access is to be enabled only to the desired set of employees or students, etc.

The ideology of feature-level fusion can be extended beyond face and fingerprint to other biometric features such as iris, voice, signature and many more. It can be implemented using the same deep learning models used in this proposed approach. The scope of deep learning models used for implementing biometric systems can be broadened to other models for different sets of goals and outcomes depending on the different requirements.

References

1. Gawande U et al (2013) A novel algorithm for feature level fusion using SVM classifier for multibiometrics-based person identification. In: Applied computational intelligence and soft computing. Hindawi Publishing Corporation, vol 2013, p 11
2. Janani T, Ramanan A (2017) Feature fusion for efficient object classification using deep and Shallow learning. Int J Mach Learn Comput
3. Shehu YI et al (2018) Sokoto coventry fingerprint dataset: a biometric fingerprint database designed for academic research purposes
4. Huang GB et al, Labeled faces in the wild: a database for studying face recognition in unconstrained environments
5. Barnouti NH et al (2016) Face detection and recognition using Viola-Jones with PCA-LDA and square Euclidean distance. Int J Adv Comput Sci Appl (IJACSA)
6. Maio D et al (2004) FVC2004: third fingerprint verification competition. In: Biometric authentication, first international conference, ICBA
7. Simonyan K, Zisserman A, Very deep convolutional networks for large-scale image recognition
8. He K et al, Deep residual learning for image recognition
9. Huang R, Sun S (2010) An adaptive *k*-nearest neighbor algorithm, vol 1, pp 91–94

Artificial Intelligence for Satellite Image Processing: Application to Rainfall Estimation

Fethi Ouallouche, Karim Labadi, Yacine Mohia, Mourad Lazri, and Soltane Ameur

Abstract Meteorological satellites are widely used for collecting information about the atmosphere. Due to the indirect relation between satellite data and measurements of meteorological parameters, mathematical models, especially those based on artificial intelligence, have been developed for meteorological modeling. Indeed, in recent years, machine learning has enabled fundamental advances in the modeling of random systems. In this context, we will show the contribution of techniques based on artificial intelligence in the estimation of precipitation. Based on the expertise of our research laboratories in this field, the objective of this paper is to present our recent results and developments using machine learning, such as ANN, SVM, and RF. For the classification and estimation of rainfall intensities, satellite images were used for the implementation of these techniques. The training and validation was carried out by comparing the satellite images to the corresponding radar images. The results of these artificial intelligence-based techniques indicate very interesting performance.

Keywords Weather satellites · Intelligence artificial · Machine learning · Precipitation

1 Introduction

Artificial intelligence is used in a wide spectrum of fields to equip systems with an intelligence in operation. In recent years, we see an increasing use of artificial intelligence in the analysis, modeling, and prediction of random systems [2, 10]. This artificial intelligence based on mathematical and statistical approaches allows applications, from input data and after a learning phase, to generate reliable

F. Ouallouche · Y. Mohia · M. Lazri · S. Ameur
Laboratoire LAMPA (Laboratoire d'Analyse et de Modélisation des Phénomènes Aléatoires), Faculty G.E.I, University Mouloud MAMMERI of Tizi-Ouzou, Tizi-Ouzou, Algeria

K. Labadi (✉)
Laboratoire LR2E, ECAM-EPMI/Quartz-Lab, Cergy-Pontoise, France
e-mail: k.labadi@ecam-epmi.com

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_14

responses at the output of the system [11]. The implement of this concept has shown its effectiveness in data processing in several areas such as, pattern recognition, image segmentation, weather forecasting, classification, and modeling. A large number of techniques based on artificial intelligence have been developed, we can cite support vector machine (SVM), artificial neural network (ANN), Random forest (RF) or the Bayesian classifier (e.g., [5, 6, 9, 17]). Note that all these techniques use the principle of machine learning.

In this paper, we are interested in the contribution of artificial intelligence (AI) to classification, estimation of precipitation and weather forecasting. Indeed, the measurement of atmospheric parameters is fundamental for weather modeling and forecasting [14]. However, direct measurement instruments for collecting atmospheric data remain insufficient for a reliable evaluation of atmospheric parameters. The incorporation of satellite data therefore allows a wide field of observation and provides multispectral information at very large spatio-temporal scales. Mathematical models based on artificial intelligence are thus used to process these satellite data [8].

The main contribution in this paper is to make a state-of-the-art on the application of classification and estimation of precipitation intensities technics from satellite data using artificial intelligence.

2 Satellite Data

Since the introduction of meteorological satellites, information on atmospheric parameters is available in large quantities. This made it possible to respond reliably to the issues raised in meteorology, such as climate change, weather forecasts, and the measurement of climate variables, etc.

Meteorological satellites, both scrolling and geostationary, provide information in the form of images at regular intervals over large regions of the globe, allowing different meteorological parameters to be visualized (clouds, precipitation, winds, fog, etc.).

One of the most difficult and complex variables to measure is precipitation. To this end, several "research and development" satellite missions have been set up to estimate precipitation. We can cite, among others, tropical rainfall measuring mission (TRMM) and Meteosat Second Generation (MSG). These satellites mark an important step in the development of satellite precipitation estimation techniques. We are witnessing a massive use of satellite images, especially those from MSG. This is one of the European Eumetsat meteorological satellites, whose data are rich and complete, which observes Europe, Africa, and part of Asia. The MSG Spinning Enhanced Visible and Infra-Red Imager (SEVIRI) radiometer allows images to be acquired every 15 min. It has 11 channels using spectral bands ranging from visible to thermal infrared, with a spatial resolution of 3 km, and a panchromatic channel with high spatial resolution (1 km).

In order to make better use of this data, artificial intelligence was profusely used, which allowed a remarkable progress in meteorology, particularly in the estimation of precipitation.

3 Application of AI to the Rainfall Estimation

The scientific community has made massive use of artificial intelligence, particularly machine learning and deep learning, to solve problems related to the modeling of random and complex systems. Deep learning is a subset of machine learning, which is a subset of AI (Fig. 1).

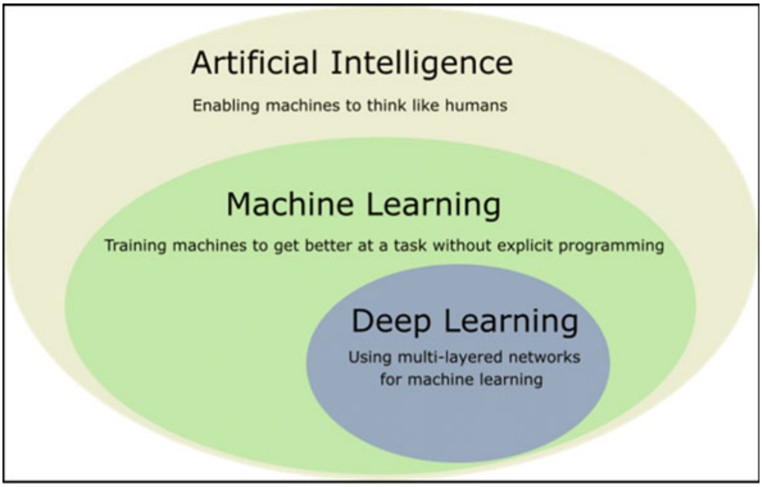


Fig. 1 AI versus machine learning versus deep learning

In meteorology, the wealth of high-quality satellite data offers the opportunity to replace traditional precipitation estimation techniques with machine learning techniques.

In the following, we will summarize some recently published machine learning techniques used in precipitation estimation.

3.1 Techniques Based on Machine Learning

In this section, we have divided the techniques according to the number of combined classifiers. All techniques use the same classification and estimation scheme, namely the classification of a precipitation scene into three classes (convective, stratiform, and not precipitating) and the estimation of precipitation during the rainy period based on classification results.

3.1.1 Mono-Classifier

We have implemented the classifiers: random forest (RF) [12], artificial neural network (ANN) [7], support vector machine (SVM) [8], naive Bayesian (NB) [4] and weighted *k*-nearest neighbors (WkNN) [1]. To show the superiority of artificial intelligence, we have also implemented the CS-RADST (Convective/Stratiform Rain Area Delineation Technique) method developed by Lazri et al. [6] which is a method based on multi-thresholding and enhanced the CST technique (ECST) to be applied to convective systems in the mid-latitude developed by Reudenbach et al. [15, 16]. The ECST which is similar to the CST but additionally includes the water vapor channel temperature for a more reliable deep convective/cirrus clouds discrimination [19]. All these techniques are applied to an instantaneous precipitation scene taken in northern Algeria during a passage of an anticyclone on December 29, 2010 (Fig. 2).

Convective cells are embedded in the anticyclone area. The classification results are shown in Fig. 3 (Table 1).

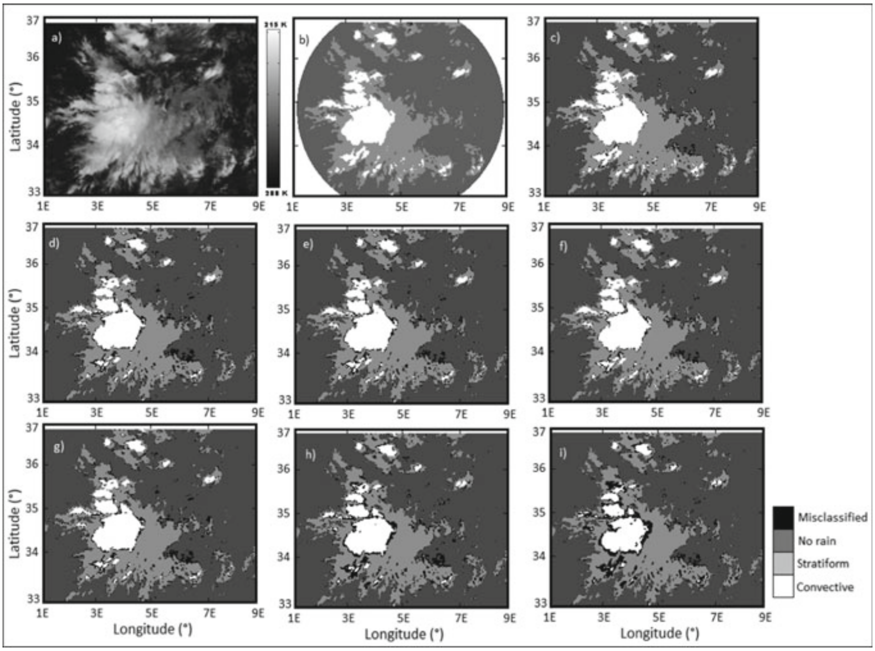


Fig. 2 Classification results, **a** IR10.8 image, **b** preprocessed radar image, **c** using RF, **d** using WkNN, **e** using SVM, **f** using BN, **g** using ANN, **h** using CS-RADST, **i** using ECST

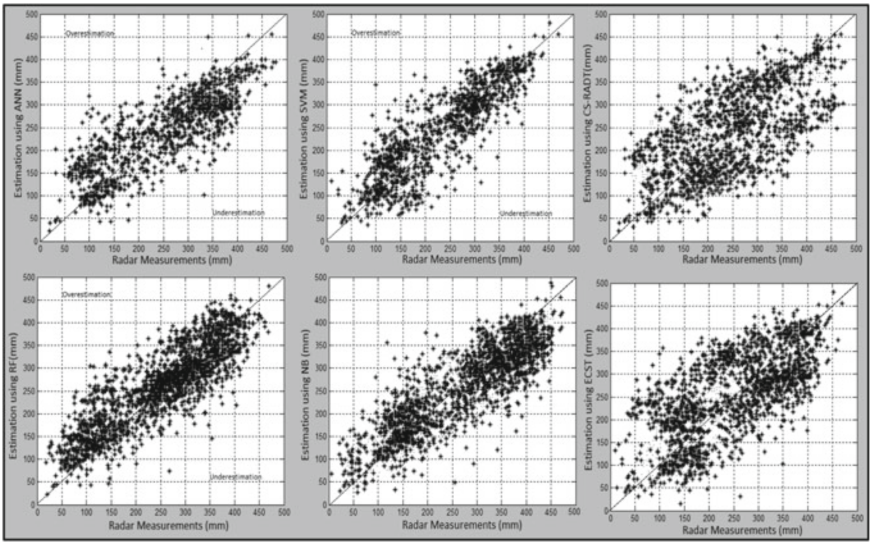


Fig. 3 Estimation by the different techniques

Table 1 Evaluation parameters results for classification and estimation

Technics	Classification				Estimation		
	POD	POFD	FAR	Bias	RMSD (mm)	Bias (mm)	Correlation coef.
RF	91	09	29	1.1	18.2	29.5	0.85
ANN	90	11	31	1.2	−20	33.5	0.67
SVM	90	12	30	0.9	18.5	32.6	0.70
BN	88	15	32	0.6	23	35	0.66
WkNN	89	14	32	1.4	24.3	36.2	0.65
CS-RADST	85	21	39	1.9	29.1	40.5	0.62
ECST	84	22	39	1.9	30.5	41.2	0.62
Optimal	100	0	0	1	0	0	1

The results show that machine-learning methods perform better than standard methods for both classification and estimation.

3.1.2 Bi-Classifiers

In this part, methods that combine two classifiers are also implemented. The first combines two classifiers (Artificial Neural Network and Support Vector Machine) based on Dempster-Shafer theory (DST-A/S) and the second is a new approach based on multilayer perceptron (MLP) and support vector machine (SVM) (Com-ANN/SVM) developed by Sehad and Ameur [17].

The results of precipitation classification and estimation are shown in Figs. 4 and 5, respectively (Table 2).

Clearly, the quality of precipitation classification and estimations are significantly better when the techniques combine machine-learning classifiers.

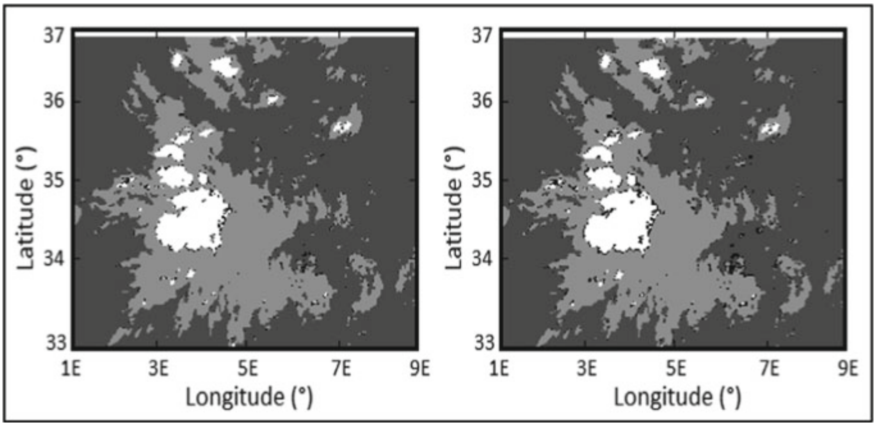


Fig. 4 Classification (right) using Com-ANN/SVM, (left) using DST-A/N

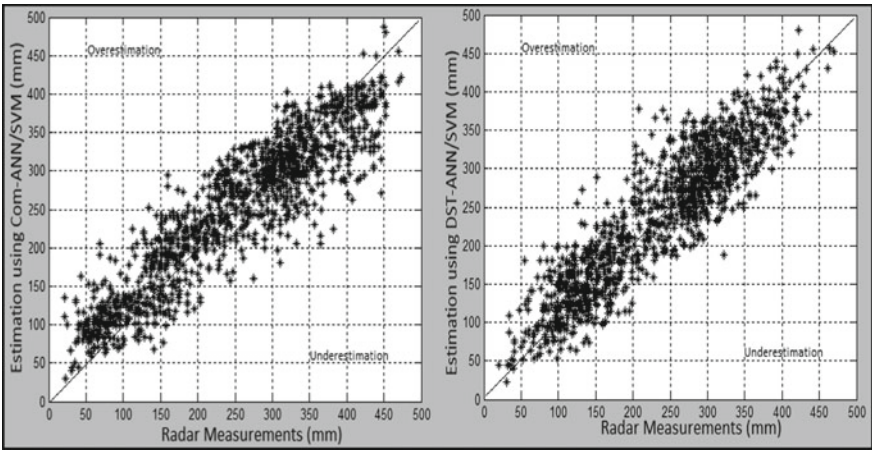


Fig. 5 Estimation results (left) using Com-ANN/SVM, (right) using DST-A/N

Table 2 Statistical values corresponding to the precipitation classification and estimation for bi-classifiers methods

Technics	Classification				Estimation		
	POD	POFD	FAR	Bias	RMSD (mm)	Bias (mm)	Correlation coef.
Com-ANN/SVM	92	08	24	1.1	15.3	26.3	0.87
DST-A/N	93	08	23	1.1	13.1	24.8	0.89
Optimal	100	0	0	1	0	0	1

3.1.3 Multi-classifiers

Two methods that use multiple classifiers are implemented. These are a model combining three classifiers, namely Support vector machine, Artificial neural network and Random forest (SAR) is designed for improving the classification of convective and stratiform rain [8] and the MMultic technique where six classifiers were first combined in order to exploit the full potential of each of these classifiers [9]. These are Random Forest (RF1), Artificial Neural Network (ANN), Support Vector Machine (SVM), Naive Bayesian (NB), Weighted *k*-Nearest Neighbors (WkNN), and the *K*means++ algorithm (*K*means).

As shown in the following, with the combination of several machine-learning classifiers, the results obtained are more interesting (see Figs. 6, 7 and Table 3). However, classification techniques based on artificial intelligence have all been exhausted and even combinations of classifiers have been tested. To further optimize the results, the deep learning mechanism is envisaged.

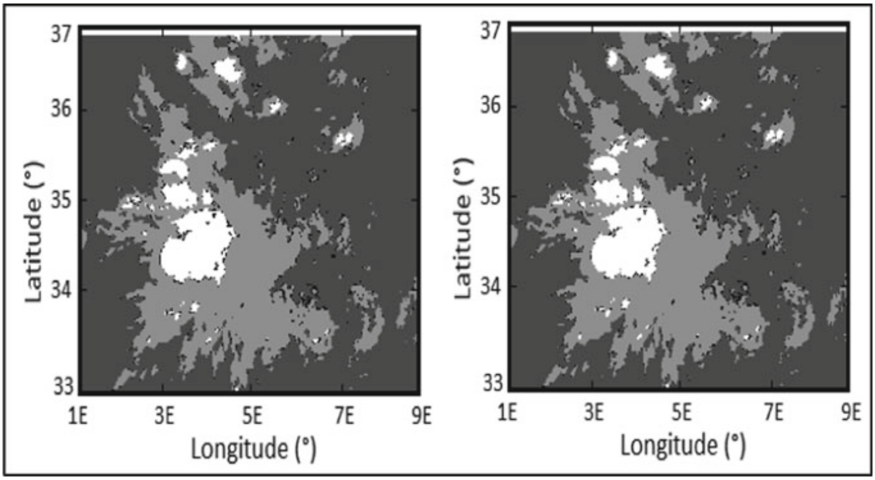


Fig. 6 Classification (right) using MMultic, (left) using ASR

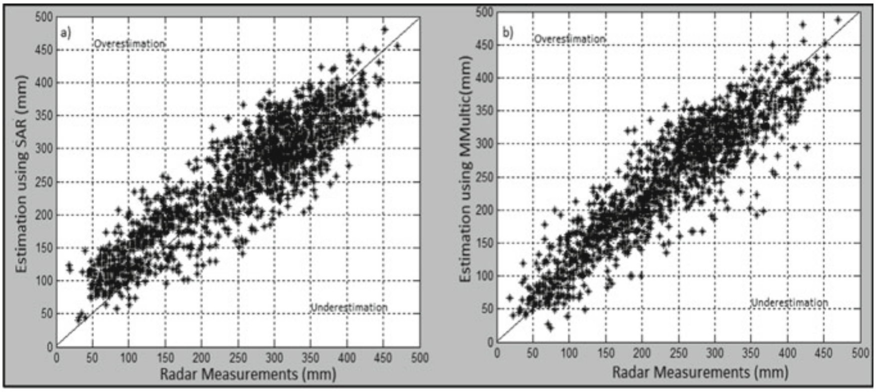


Fig. 7 Estimation results (left) using MMultic, (right) using SAR

Table 3 Statistical values corresponding to the precipitation classification and estimation for Multi-classifiers methods

Technics	Classification				Estimation		
	POD	POFD	FAR	Bias	RMSD (mm)	Bias (mm)	Correlation coe.
MMultic	96	06	20	1.05	15.2	9	0.94
SAR	95	07	22	1.06	24.6	12.5	0.90
Optimal	100	0	0	1	0	0	1

3.2 Contribution of Deep Learning

Very recently, deep learning has allowed significant progress due to the consideration of the complexity of random systems [18]. This concept makes it possible to learn new rules in order to be more reliable and efficient. To reliably take into account the precipitation variability, modern techniques based on deep learning, such as deep neural networks, convolutional neural networks, or even deep belief networks are increasingly used. The results by exploring deep leaning show much higher performance than the standard use of machine leaning [13]. These are made up of fairly deep neural architectures where a large number of neurons and layers are combined. It is therefore a field in the process of being explored in the estimation of precipitation after the successes achieved in classification, facial, voice recognition…. For more details on deep learning, the reader can refer to [3].

4 Conclusion

In this article, we highlighted the contribution of artificial intelligence in the classification and estimation of precipitation. To do this, we have implemented a set of machine learning-based methods, namely ANN, SVM, RF, NB, and WKNN. To compare, we also applied other standard methods on the same database.

First, we applied the different methods for classifying a precipitation scene in the presence of convective cells in a stratiform system. The results show superior performance for machine learning methods. The combination of several machine-learning classifiers yielded even more interesting results. Misclassified pixels were reclassified by combination methods. Thus, the rates of good classification have increased. Secondly, rainfall estimates were made. In the case of these estimates, we noted the same trend as the classification results. The best results were obtained for machine learning methods. The results are even better when several classifiers are combined. In addition, the implementation of deep learning will allow a better classification of the precipitation intensities, and consequently the precipitation estimate. Due to the deep architecture of neural networks used, it is possible to take into account the complexity of random systems.

References

1. Bensafi N, Lazri M, Ameur S (2019) Novel WkNN-based technique to improve instantaneous rainfall estimation over the north of Algeria using the multispectral MSG SEVIRI imagery. J Atmos Sol Terres Phys 183:110–119. ISSN 1364-6826. https://doi.org/10.1016/j.jastp.2018.12.004
2. Bibault J-E, Xing L (2021) Apports de l'intelligence artificielle en épidémiologie. Cancer Radiothér 25(6–7):627–629. ISSN 1278-3218. https://doi.org/10.1016/j.canrad.2021.06.004
3. Deng L (2014) A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Trans Sig Inf Process 3
4. Hameg S, Lazri M, Ameur S (2016) Using naive Bayes classifier for classification of convective rainfall intensities based on spectral characteristics retrieved from SEVIRI. J Earth Syst Sci 125:945–955. https://doi.org/10.1007/s12040-016-0717-7
5. Kuhnlein M, Appelhans T, Thies B, Nauss T (2014) Improving the accuracy of rainfall rates from optical satellite sensors with machine learning—a random forests-based approach applied to MSG SEVIRI. Remote Sens Environ 141:129–143
6. Lazri M, Ameur Z, Ameur S, Mohia Y, Brucker JM, Testud J (2013) Rainfall estimation over a Mediterranean region using a method based on various spectral parameters of SEVIRI-MSG. Adv Space Res 52:1450–1466
7. Lazri M, Ameur S, Mohia Y (2014) Instantaneous rainfall estimation using neural network from multispectral observations of SEVIRI radiometer and its application in estimation of daily and monthly rainfall. Adv Space Res 53:138–155
8. Lazri M, Ameur S (2018) Combination of support vector machine, artificial neural network and random forest for improving the classification of convective and stratiform-rain using spectral features of SEVIRI data. Atmos Res 203:118–129
9. Lazri M, Labadi K, Brucker JM, Ameur S (2020) Improving satellite rainfall estimation from MSG data in Northern Algeria by using a multi-classifier model based on machine learning. J Hydrol 584:124705

10. Le Guillou G, Malvache P, Himbaut S, Pham H (1985) Application de l'intelligence artificielle au projet de systeme de detection et de diagnostic base sur l'analyse des bruits. Progr Nuclear Energy 15:889–896. ISSN 0149-1970. https://doi.org/10.1016/0149-1970(85)90123-4
11. Malinetsky G, Vladimir S (2021) The artificial intelligence influence on real sociality. Proc Comput Sci 186:344–351. https://doi.org/10.1016/j.procs.2021.04.155
12. Ouallouche F, Lazri M, Ameur S (2018) Improvement of rainfall estimation from MSG data using random forests classification and regression. Atmosp Res 211:62–72. ISSN 0169-8095. https://doi.org/10.1016/j.atmosres.2018.05.001
13. Oukali S, Lazri M, Labadi K, Brucker JM, Ameur S (2019) Development of a hybrid classification technique based on deep learning applied to MSG/SEVIRI multispectral data. J Atmos Solar Terr Phys 193:105062
14. Plocoste T, Calif R, Jacoby-Koaly S (2019) Multi-scale time dependent correlation between synchronous measurements of ground-level ozone and meteorological parameters in the Caribbean Basin. Atmosp Environ 211:234–246. ISSN 1352-2310. https://doi.org/10.1016/j.atmosenv.2019.05.001
15. Reudenbach C, Heinemann G, Heuel E et al (2001) Investigation of summertime convective rainfall in Western Europe based on a synergy of remote sensing data and numerical models. Meteorol Atmos Phys 76:23–41. https://doi.org/10.1007/s007030170037
16. Reudenbach C, Asgard-Verlag (2003) Konvektive Sommerniederschlage in Mitteleuropa. Eine Kombination aus Satellitenfernerkundung und numerischer Modellierung zur automatischen Erfassung 25 mesoskaliger Niederschlagsfelder, Bonner Geographische Abhandlungen, vol 109, 152pp
17. Sehad M, Ameur S (2020) A multilayer perceptron and multiclass support vector machine based high accuracy technique for daily rainfall estimation from MSG SEVIRI data. Adv Space Res 65(4):1250–1262. ISSN 0273-1177. https://doi.org/10.1016/j.asr.2019.11.018
18. Dong S, Wang P, Abbas K (2021) A survey on deep learning and its applications. Comput Sci Rev 40:100379. ISSN 1574-0137. https://doi.org/10.1016/j.cosrev.2021.100379
19. Tjemkes SA, van de Berg L, Schmetz J (1997) Warm water vapour pixels over high clouds as observed by Meteosat. Beitr Phys Atmos 70:15–21

Cyber Security and Data Security

A Comprehensive Review of Various Approaches to Intrusion Detection Systems

Swati Shinde, Tejas Borde, Aditya Deo, Suraj Dhamak, and Shreyas Dungarwal

Abstract The extensive usage of the Internet to communicate and transfer data might succumb to various network related threats. Intrusions are one such threat, where the client/organization is at a risk of data theft. An intruder is someone who gains unauthorized access to our network or system. A network falling prey to an intrusion might result in loss of valuable data. A solution to intrusions is intrusion detection systems (IDS). This paper provides a comprehensive review of approaches to build IDS. The first section covers a review of the fundamentals of IDS, covering various intrusion types and IDSs, their strengths and their limitations. The next section discusses intrusions in wireless networks, followed by a review of a wireless approach to intrusion detection systems for IEEE 802.11 networks. The next section takes a look at various deep learning and machine learning approaches to intrusion detection systems that are currently being implemented. It summarizes the benchmark datasets that are currently being used to implement models for intrusion detection, followed by the results of a few machine learning models implemented on the NSL-KDD dataset.

Keywords Networks · Deep learning · Wireless networks · Machine learning intrusion detection systems

1 Introduction

Networks and computer systems are nowadays an integral part of the techno era. Computer networks connecting the organizations worldwide have numerous features which enhance the day-to-day functioning of tech-industries. A computer network is a virtual pathway that connects various networking devices. Networks being classified into two categories, namely wireless and wired, are used as per the requirement of the organization. With several uses and advantages, computer networks possess various security threats. Any technology can be a boon or a curse, it depends upon

S. Shinde · T. Borde · A. Deo (✉) · S. Dhamak · S. Dungarwal
Pimpri Chinchwad College of Engineering, Pune, Maharashtra 411035, India
e-mail: ahdeo2015@gmail.com

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_15

the user and the way that particular technology is handled. In a data-driven society, where data is the most precious resource, it is often vulnerable to threats if it is not protected. Unauthorized access or malicious use of information resources as well trying to find ways to gain unapproved access to data causing harm, or intrusion into other malicious content is called intrusion. Any organization—especially an IT organization—thrives on the way it manipulates its data (which is often sensitive and must be kept protected). If such data falls prey to an intrusion, or in more simpler words, unauthorized or illegal access, it has potential to cause large losses to the organization. Networks, if mishandled possess terrible threats to the organizations. People with malicious intent often act as intruders and intrude the network traffic and further satisfy their motive. Intruders often aim to destroy the integrity of the organizations data, and further, they may also plan to affect the availability of sensitive data. Intruders may have various intentions behind disturbing the integrity, availability as well as the confidentiality of the data of an organization. A solution to detect such intrusions that pose a threat to the secureness of the organization is intrusion detection systems (IDS).

2 Intrusion Detection System (IDS)

IDS is a system that enhances the security of a network and protects organizational information. IDS assists the network administrator to detect any suspicious activity and warns the controller to retrieve secure data by taking measures against such an activity. IDS can be host-based IDS (HIDS) and network-based IDS (NIDS). A NIDS exists on a device connected to the network and monitors network traffic to that part of the network, requiring continuous attack [1]. HIDS is installed on a server, known as a transmitter, and monitors activity only in that system. HIDS has two advanced types such as signature-based detection strategies namely detection by targeting patterns, such as byte sequencing, or malicious sequence instructions used by malware, and an anomaly-based detection process to detect both net Internet access and misuse by monitoring various aspects of the system and classifying it as undesirable or normal. Data collection which means transmitting data such as IDS, selecting a specific feature, analyzing data for accuracy, and describing system responses and attacks. The components of the IDS are sensors, detection, and reporting, retrospective collection and awareness, and the IDS control and control frontend. Various tools are available with IDS viz., SNORT, OSSEC-HIDS, KISMET, etc. Snort, being an open-source software that is small in size, uses descriptive legal language used to describe traffic coming from a specific IP address. OSSEC (open-source security) is an open-source software that works on large applications and apply the design for the client/server.

The three classes within Table 1 are HIDS, wireless-based IDS (WIDS), and NIDS. HIDS and NIDS are explained above, WIDS is analogous to NIDS, but considers network traffic from wireless devices and networks. There is another system called

Table 1 Types of IDS

Item	HIDS	NIDS	WIDS
Components	Agent: software Management server: 1 ~ n Database server: 1 ~ n (optional)	Sensor: n Management server: 1 ~ n Database server: 1 ~ n (optional)	Sensor: n Management server: 1 ~ n Database server: 1 ~ n (optional)
Detection scope of sensor	Singular host	Network subnet: n Host: n	WLAN: n WLAN client: n
Architecture	Managed network or standard network	Managed network	Managed network or standard network
Strengths	Can analyze communication functions that are end-to-end encrypted	Able to analyze wide scales of AP protocols	Can accurately monitor the activity of the wireless protocol
Technology limitations	Cannot obtain great accuracy as there is no contextual information Warnings and reports are generated late Consumes a lot of resources	Does not work for wireless environments False positives and false negatives are considerably more Bad performance under high loads	Unable to keep track of NL, AL, and TL protocol activities Evasion techniques cannot be avoided Sensors are vulnerable to physical jamming attacks. Extra security layers are needed for wireless environments

the NBA system which monitors network traffic to detect attacks of unexpected road flow.

A drawback of intrusion detection systems is that it is unable to detect to a great accuracy. False-positive (FP), occurring when IDS incorrectly identifies a harmless activity as dangerous activity, and false-negative (FN), which occurs when IDS fails to detect dangerous patterns, are two indicators of testing for accuracy.

3 Wireless Intrusion Detection System

Wireless networks, when compared with wired networks are considered to be more vulnerable to security threats. The major reason behind the high vulnerability rate in case of wireless networks is the openness of wireless networks. It is quite simple for the attackers to target wireless networks and exploit them to satisfy their motive. Wireless IDS plays an important role in securing the wireless networks worldwide. IDS are nothing but intrusion detection systems which constantly keep monitoring the network to trace out any kind of malicious activity in the premises of a network.

Scaled Wireless Networks

Rapid advancements in IT field and its alliances have led to vast scaling of networks. Networks are not only involved in the IT field but are an integral part of each field which uses computers or servers as its sub-components. Every field has adapted the concepts of modern computing to serve their purpose smoothly. When modern computing comes in play networks definitely come in action.

Data Security

Networks, either it be wired or wireless, involve data flow. Principle function of any kind of network is to transport data packets from one destination to another. Data can be casual or sensitive and it is the prime responsibility of the network manage to protect that data from people with malicious intent. Data security is nowadays becoming a major pillar supporting the modern computation.

Increased Cyber Attacks

Since past two decades, it's observed that the number of cyber-attacks is tremendously increasing. As they are open access, wireless networks are highly vulnerable to cyber-attacks. Number of cyber-attacks like SQL injection, malware attack, denial of service attack, man in the middle attack, password attack, cross-site scripting, etc., majorly target wireless networks.

4 Wireless Intrusion Detection, Prevention, and Attack

Figure 1 illustrates the process of intrusion detection in wireless systems. The threat identification is the initial step which involves identifying the internal and external threats to a network. After finding the expected threats, the next step is to collect and monitor all the data flowing through the system. When the data traffic is collected and monitored, the system uses the previous traces of such instances to compare the current data the current data and check if something is going fishy. If anything malicious is detected in the recorded data, the next step is to trace out the source of that attack and to provide a system alert to the network manager [2].

Wireless Network Security based on Centralized and Distributed Systems
Centralized Wireless IDS

This architecture involved wireless sensors and a central processing unit called analyzer. Sensors keep on recording and scanning the data flowing in the network

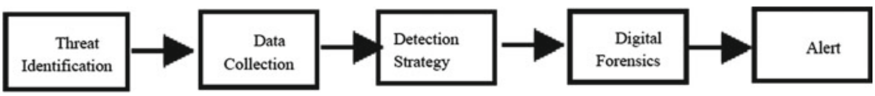


Fig. 1 Steps of wireless IDS

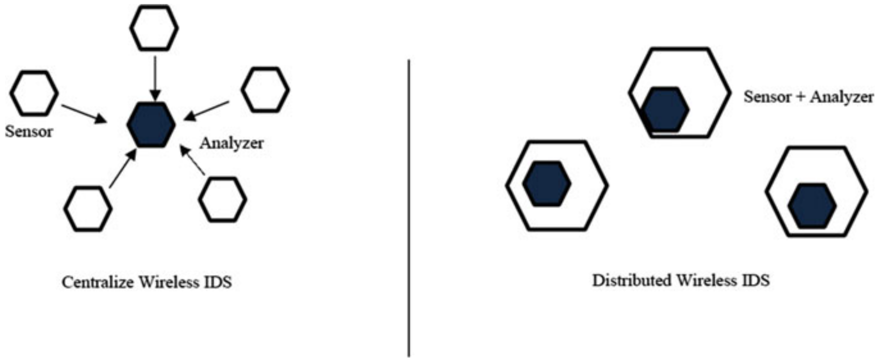


Fig. 2 Wireless IDS architecture

and report it to analyzer for further processing. Analyzer then detects if any anomaly is present in the data traffic and raises the alarm.

Distributed Wireless IDS

This architecture in [3] involved wireless sensors capable of scanning and processing the data on its own. These sensors were not dependent on the central analyzer. The complete process was handled by the intelligent sensors, and the sensor itself raised an alarm to inform the network manager regarding the anomalies (Fig. 2).

5 IDS for 802.11 Networks—A Review

This system intends to focus on the CIA triad to protect wireless networks from cyber-attacks. It functions particularly to protect against deauthentication attack and evil twin attack. Deauthentication attack is a scenario when an attacker sends false deauthentication frames to the server or the client to purposely terminate the established client–server connection. Evil twin attack is a scenario where the attacker generates rogue or false WAPs and attracts the legitimate user to its AP and then carries out the further attacks by creating a DoS scenario. Some parameters are considered to detect these attacks [4].

Parameters Considered for Deauthentication Attack

Number of frames: A fixed value is defined for number of deauthentication frames, if the number of frames is greater than the threshold value, an alert is raised, informing the administrator about the attack.

Number of duplicates: Each deauthentication frame is unique, and attackers generally follow a pattern where the deauthentication frames are duplicated after specific interval which is an attack scenario.

Reason for deauthentication: Whenever, the deauthentication frames are sent, the sender specifies the reason for deauthentication as a code, the attackers generally do not have this code in their deauth-frames and thus, an attack scenario is highlighted.

Data after deauthentication: If a legitimate deauthentication frame is sent, there is no data transfer after it. In case of false deauth-frames, some amount of data is still transmitted after the frames and again an attack scenario is highlighted, and an alarm is raised.

Parameters Considered for Evil Twin Attack

Beacon frames: Beacon frames are used by the AP to attract clients toward itself. The number of beacon frames has a threshold value. If the attacker crosses the threshold value, it is classified as an attack scenario.

Duplicate SSID: Every beacon frame has an SSID associated to it, and it is legitimate to the particular AP. The IDS monitors all the beacon frames and checks if the SSID is duplicated, if a duplicate SSID is detected, then it is an attack scenario.

Timestamp: The beacon frames carry a time stamp with them. In a valid case, the timestamp should increment gradually with each beacon frame. The attackers usually keep the time stamp constants. While scanning, if constant time stamps are noticed, the attack alarm is raised.

6 IDS Using Machine Learning (ML) and Deep Learning (DL)

The rapid growth and advancement of the Internet, services through the Internet and communication through the Internet brings consequent chances of threats to the networks. Many novel attacks on the network pose a risk of data leakage, data theft, and network security breaches. An IDS is a system that identifies such attacks and undertakes further actions to eliminate the threat. Recently, with the emergence of modern algorithms, the field of machine learning and deep learning are being seriously considered as a solution to detect intrusions in networks, and thus, prevent novel attacks.

Now is the best time to implement ML and DL approaches to IDS for the following reasons:

- Because of the advent of the Internet, more and more open-source data is available for researchers to research upon.
- Various algorithms in machine learning and deep learning are already showing promising results, but there is still a lot of scope for further research and implementation.

Evaluation Metrics

To test how well a deep learning method has performed, the following evaluation metrics are used. All evaluation metrics are based on the following parameters, which are the attributes of the confusion matrix [5]:

- True Positive (TP) indicates that the data instance is 'attacked' and is classified accurately;
- False Negative (FN) means that the data instance is 'attack' but is wrongly classified as 'normal';
- False Positive (FP) indicates that the data instance is 'normal' but is wrongly identified as 'attacked';
- True Negative (TN) means that the data instance is 'normal' and is correctly classified.

Based on these parameters, we have the following evaluation metrics:

Accuracy, precision, recall, True Negative Rate, and False Positive Rate. They are defined as

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{True Negative Rate} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

$$\text{False Positive Rate} = \frac{\text{FP}}{\text{TN} + \text{FP}}$$

The 'samples' mentioned in the above evaluation metric definitions are used from the benchmark datasets that are mentioned below in Table 2. Out of the mentioned datasets, the NSL-KDD, even though it is critically imbalanced, is observed to be the most used dataset for research purposes. Apart from the KDD Cup'99, the NSL-KDD dataset, and the Kyoto 2006 datasets are also found to be prominently used.

Machine Learning Approaches to IDS

The different machine learning approaches used for intrusion detection are mentioned in Fig. 3. All of those are described below in detail.

_K_-Means Clustering

K-means clustering is an unsupervised learning technique. There are '*K*' groups/clusters, each group having a specific set of qualities. Based on the dataset's features, the algorithm allocates the data points to one of the *K* clusters. The example points are grouped based on how similar the features are.

Table 2 Benchmark datasets

Dataset	Year	Attack types	Attacks
KDD Cup '99	1998	4	Denial of service, probe, remote to user, user to root
Kyoto (2006)	2006	2	Unknown attacks, known attacks
NSL-KDD	2009	4	Denial of service, probe, remote to user, user to root
UNSW-NB15	2015	9	Backdoors, shellcode, worms, Denial of Service, Port scans, fuzzers, exploits, generic, reconnaissance
CIC-IDS2017	2017	7	Brute force, HeartBleed, web, infiltration, Botnet, DoS, distributed Denial of service
CSE-CIC-IDS2018	2018	7	HeartBleed, Denial of service, Botnet, distributed Denial of Service, BruteForce, infiltration, web

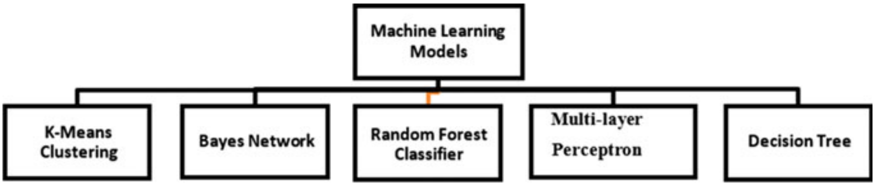


Fig. 3 ML approach to IDS

Decision Tree

The classification approach used to make excellent judgments involving high costs and risks is called decision tree. It uses the method to compare the competing options. It employs a top-down strategy. It is used to arrange records from root node to leaf node in order to separate them. In paper [6], the author had proposed two methodologies to generate decision trees (with and without pruning, where pruning is the method used to reduce decision tree size by removing the part of the dataset that is not a contributor to classifier accuracy). Decisions to build decision tree are taken using the gain ratio of each attribute. Calculation of gain ratio is done using the gain and the split-info of the attribute.

$$\text{Gain Ratio} = \frac{\text{Gain of Attribute}}{\text{Split info of Attribute}}$$

Random Forest Classifier

Random forest uses several techniques to classify data. On a random selection of data, they build numerous decision trees (as depicted in Fig. 4). It then either adds up

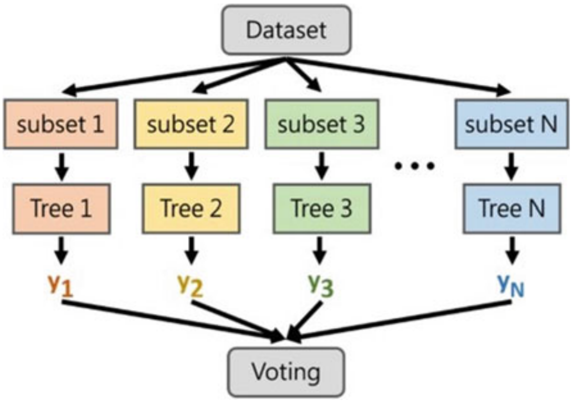


Fig. 4 Forest classifier

each tree's total votes to determine the test's class, or it gives weights to each tree's contribution [7]. It is an ensemble classifier.

Bayes Network

Bayesian networks are a graph model that shows the relationships between the nodes on the graph's edge. Bayesian networks are probabilistic graphical models that formulate dependencies on a directed graph's edge [8]. They use the fact that the nodes do not have any conditional reliance to create acyclic graphs.

Multi-layer Perceptron (MLP)

It is a feed-forward neural network. The input layer, the hidden layer(s), and the output layer are the three layers that make up the system. Weights or parameters are changed during training to reduce classification error. Each buried node introduces nonlinearity. Updation of the weights and biases is do not with the help of back propagation.

Table 3 shows the average accuracy given by the machine learning models in [9].

Deep Learning Approaches to IDS

The different deep learning algorithms used for intrusion detection are depicted in Fig. 5. All of them are described below in detail:

Table 3 Machine learning classifiers accuracy

Machine learning classifiers	Average accuracy (%)
Random tree	90.57
Random forest	93.77
Bayes network	90.73
Naïve Bayes	91.23
MLP	91.90
Decision table	92.44

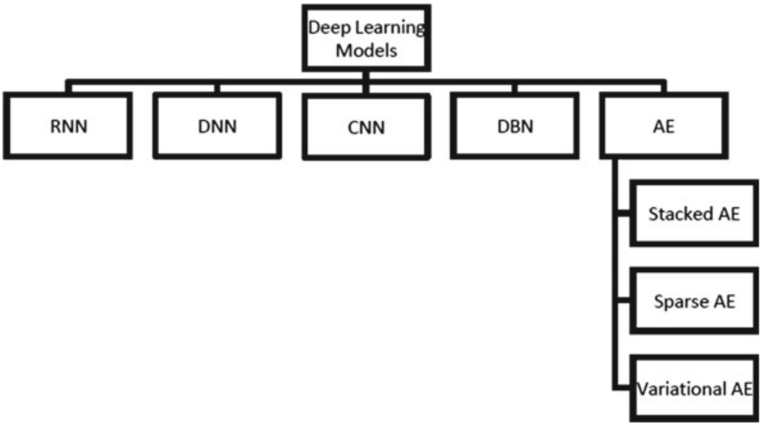


Fig. 5 DL approaches to IDS

Recurrent Neural Networks (RNNs)

A recurrent neural network uses time collection data. RNN extends the talents of the conventional neural network by adding additional memory factors. RNN is made from input, hidden, and output layers, wherein the hidden layers are considered to be the memory factors. The authors of [10] have devised an approach to detect intrusions using RNNs, called 'RNN-IDS'. The dataset used is the NSL-KDD dataset, and the evaluation metrics used are true positive rate, accuracy, and false positive rate. The RNN-IDS, with its great performance compared to traditional machine learning methods, proves to be very much suitable for modeling IDSs.

Deep Neural Networks (DNNs)

It is a type of artificial neural network consisting of an input layer, several hidden layers, and finally, an output layer. It is a classic supervised learning neural network architecture, and the model learns through the layers. The authors of [11] have used DNN as one of the models of their proposed ensemble machine learning model. The dataset used is NSL-KDD dataset, and the evaluation metrics are precision, accuracy, and recall.

Convolutional Neural Networks (CNNs)

A CNN is generally used to analyze, process, and classify image data. It consists of multiple layers of convolutional filters and pooling layers, which help extract specific features from the image/data. Convolutional neural networks are found to be very much successful for data with finite dimensions stored in arrays. For IDS, convolutional neural networks can be used to extract features from sample data and predict and classify intrusions on new input data [12].

Deep Belief Networks (DBNs)

It is a class of neural networks which consists of several hidden layers. The layers are connected to each other but the units within the layers are not. A DBN is obtained by the stacking of several restricted Boltzmann machines in layers and adding a SoftMax layer at the end for classification. In the context of IDSs, a DBN can be used for classification of data samples as well as for the purpose of feature extraction.

Autoencoders

As stated by the authors of the review paper [13], autoencoders are the most frequently used approach for implementation of IDSs using deep learning. An autoencoder focuses on learning through matching the output as close as the input. It consists of an input layer and an output layer (of the same size and dimensions) and several hidden layers, generally smaller in size than the input layer. The authors of [13] use 'self-taught learning', an unsupervised deep learning method which implements a sparse autoencoder for learning on the NSL-KDD dataset. The evaluation metrics used are precision, accuracy, and recall.

7 Experiment

For this experiment, we performed different classifiers (namely K-neighbors, random forest, logistic regression, decision tree, Gaussian NB, and SGDC) on the NSL-KDD dataset. The confusion matrices of different classifiers are shown in Fig. 6.

Table 4 shows the results of the experiments conducted. K-neighbor, random forest, and decision tree classifiers show reasonably good $F1$, precision, and recall scores on the NSL-KDD test dataset. Logistic regression, Gaussian NB, and SGDC classifiers show comparatively moderate to poorer scores. These results, however, do not provide any robust conclusion as to which of these models are best for intrusion detection systems, as the hyperparameters of these models when carrying out the experimentations were inadequately tuned. Moreover, the dataset used for experimentations is incredibly imbalanced.

8 Conclusion and Future Scope

Computer networks are being scaled and utilized to satisfy the needs of a variety of users and organizations, the need to secure such networks and preserve the integrity and confidentiality of the data flowing throughout the network is at its peak in the modern techno era. The intruders or the people with malicious intent use several modern and advanced techniques to get access to the network data. Similarly, we need to approach a major advancement in cyber security field. Considering the current scenario of the cyber world, the hackers always seem to be a step ahead of the security

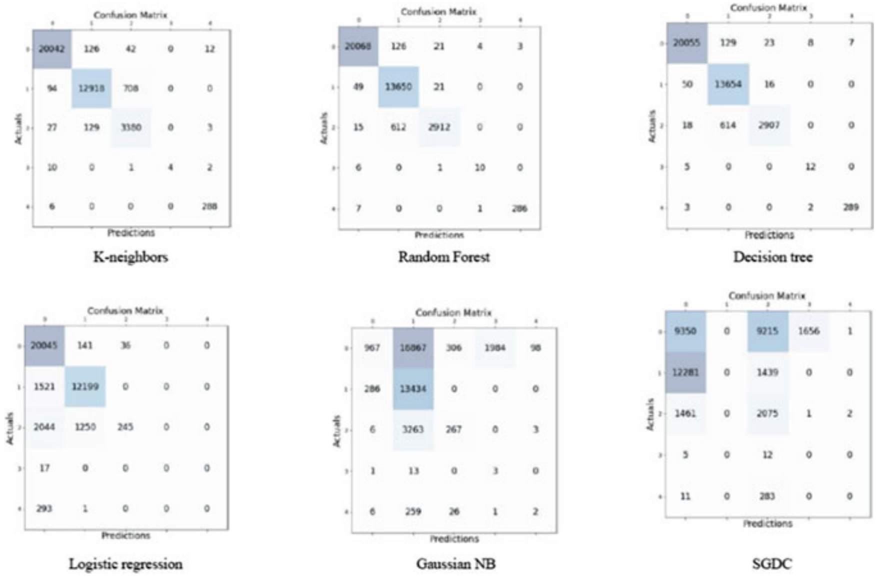


Fig. 6 Results

Table 4 Results

Classifier	$F1$-score	Precision score	Recall score
K-neighbors	0.83	0.95	0.82
Random forest	0.89	0.92	0.87
SGDC	0.14	0.11	0.21
Logistic regression	0.38	0.52	0.39
Gaussian NB	0.15	0.32	0.25
Decision tree	0.89	0.89	0.90

professionals. Major advancement is much needed to curb malicious activities. In the case of networks, intrusion detection systems are said to be constantly evolving to ensure data security. Along with the traditional open-source setups of IDS and the wireless IDSs, machine learning, and deep learning are adding their power to maintain the security of cyberspace. The combination of ML, AI, and the traditional IDS is somewhere a major growth in the field of cyber security. Still, with many loopholes in the system, there are major chances of improvement which if applied and implemented can be a boon for security researchers and a curse for the hacker. The future scope of this review would be to devise a hybrid IDS that makes use of the advantages of both traditional as well as modern intrusion detection techniques to accurately detect intrusion.

References

1. Kumar S (2007) Survey of current network intrusion detection techniques
2. Nada JA, Al-Mosa MR (2018) A proposed wireless intrusion detection prevention and attack system. In: 2018 international Arab conference on infor mation technology (ACIT), pp. 1–5. https://doi.org/10.1109/ACIT.2018.8672722
3. Ngai E, Liu J, Lyu M (2006) On the intruder detection for Sinkhole attack in wireless sensor networks. In: IEEE international conference on communications
4. Afzal Z, Rossebø J, Talha B, Chowdhury M (2016) A wireless intrusion detection system for 802.11 networks. In: 2016 international conference on wireless communications, signal processing and networking (WiSPNET), pp 828–834. https://doi.org/10.1109/WiSPNET.2016.7566249
5. Almseidin M, Alzubi M, Szilveszter K, Alkasassbeh M (2018) Evaluation of machine learning algorithms for intrusion detection system
6. Relan NG, Patil DR (2015) Implementation of network intrusion detection system using variant of decision tree algorithm. In: 2015 international conference on Nascent technologiesin the engineering field (ICNTE), pp 1–5. https://doi.org/10.1109/ICNTE.2015.7029925
7. Farnaaz N, Akhil J (2016) Random forest modeling for network intrusion detection system. Proc Comput Sci 89:213–217. https://doi.org/10.1016/j.procs.2016.06.047
8. Alqahtani H, Sarker I, Kalim A, Hossain S, Ikhlaq S, Hossain S (2020) Cyber intrusion detection using machine learning classification technique. https://doi.org/10.1007/978-981-15-6648-6_10
9. Biswas S (2018) Intrusion detection using machine learning: a comparison study. Int J Pure Appl Math 118:101–114
10. Yin C, Zhu Y, Fei J, He X (2017) A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access 5
11. Gao X, Shan C, Hu C, Niu Z, Liu Z (2019) An adaptive ensemble machine learning model for intrusion detection. IEEE Access 7
12. Ahmad Z, Khan AS, Shiang CW, Abdullah J, Ahmad F (2020) Network intrusion detection system: a systematic study of machine learning and deep learning approaches. In: Transactions on emerging telecommunications technologies. Wiley (2020)
13. Liao H-J, Lin C-HR, Lin Y-C, Tung K-Y (2013) Intrusion detection system: a comprehensive review. J Netw Comput Appl

Extensive Analysis of Intrusion Detection System Using Deep Learning Techniques

Nishit Bhaskar Patil and Shubhalaxmi Joshi

Abstract Intrusion detection systems (IDS) is a major cyber security approach that aims to observe the status of the software and hardware components operating in a system or network. Several IDS models have been available in the literature for tackling the security issues, which can be divided into two types, namely, signature-based IDS (SIDS) and anomaly-based IDS (AIDS). Regardless of recent developments, the present IDSs still needed to enhance the detection performance, minimize the false alarms, and recognizing unknown attack. For resolving these issues, several research works have dedicated on the design of IDS via machine learning (ML) and deep learning (DL) models. In this aspect, this study intends to perform a complete review of recently developed DL models for IDS. Besides, a detailed review of various DL models designed to identify the intrusions in the network take place. In addition, an extensive analysis of the reviewed approaches is performed in terms of different aspects such as objectives, underlying methodology, dataset used, and measures. Moreover, a brief discussion of the results obtained by the DL-based IDS models is also made. At last, the possible future developments and challenges involved in the IDS models are elaborated briefly.

Keywords Deep learning · Machine learning · Intrusion detection system · Cybersecurity · Attack detection · Data classification

1 Introduction

The transformation of malicious software shows crucial challenges to the development of intrusion detection system (IDS). The malicious attack has become increasingly complex and the primary problem is to find obfuscated and unknown malware

N. B. Patil (✉)
School of Computer Science, MIT-WPU, Pune, Maharashtra, India
e-mail: nsht.patil@gmail.com

S. Joshi
Department of Master of Computer Application, Faculty of Science, MIT-WPU, Pune, Maharashtra, India

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_16

[1]. Additionally, there has been increasing threats to security like zero-day attack developed to end internet user [2]. Thus, computer security is becoming very important as the usage of information technology has been an essential part of their day-to-day life. Intrusion is an act operated illegally or legally in transmission networks and systems that display variation from regular actions of that network. IDS is a well-equipped software or hardware-based system used to trap the intrusion at an early stage [3]. It is an integration of methods, tools, and resources that assist in identifying the intrusion and dissolving them at the right time. First line of defense, the prevention technique for intrusions like access control, authenticity, cryptography, and secure routing [4, 5]. Figure 1 illustrates the general process involved in IDS.

Initially, IDS was presented in 1980. Since then, many advanced IDS products have emerged. But still, several IDS suffer from a higher false alarm rate, generate several alerts for lower non-threatening situations that can cause severe harmful attacks to be ignored and increase the burden for security analysts [6]. Therefore, several authors aimed at designing IDS with reduced false alarm rates and high detection rates. Other challenges with current IDS are that they cannot identify unknown attacks. Since network environment changes rapidly, attack variants and novel attacks emerge continuously. Therefore, it is essential to design IDS that could identify unknown attacks. To resolve these above-mentioned issues, authors have focused on building IDS using machine learning (ML) techniques. The ML method is a type of artificial intelligence (AI) approach that could automatically determine valuable data from large data sets [7].

The ML-based IDS could attain a reasonable detection level once satisfactory training data is available, and ML model has adequate generalizability to identify novel attacks and attack variants [8]. Additionally, ML-based IDS don't depend largely on domain knowledge; hence, they are easier to construct and design. Deep learning (DL) is a subdivision of ML method which could attain remarkable performance. In comparison with conventional ML methods, DL approaches are good at handling big data [9]. Furthermore, DL techniques could automatically learn feature representation from raw information and then output results; they are practical and function in an end-to-end method. One prominent characteristic of DL is the deeper network which has numerous hidden layers. At the same time, conventional ML techniques [10], like k-nearest neighbors (KNN) and support vector machines (SVM), contains none or only one hidden layer. Hence, this conventional ML method is known as shallow model.

This study concentrates on the survey of recently presented DL models for IDS. This study examines the existing 35 DL based intrusion detection approaches based on the performance evaluation, determines the research gap, and highlights the reviewed works. All the articles selected for analysis are based on the performance outcome and its accuracy results. The reviewed methods are examined in different ways such as objectives, methodology used, datasets used, and performance measures. In addition, an expressive and comparison study of the surveyed DL enabled IDS models takes place by offering a tabular format. Then, the performance analysis of the different DL models takes place and identifies the optimal solutions. Lastly, the research issues and potential future scope of the research are highlighted

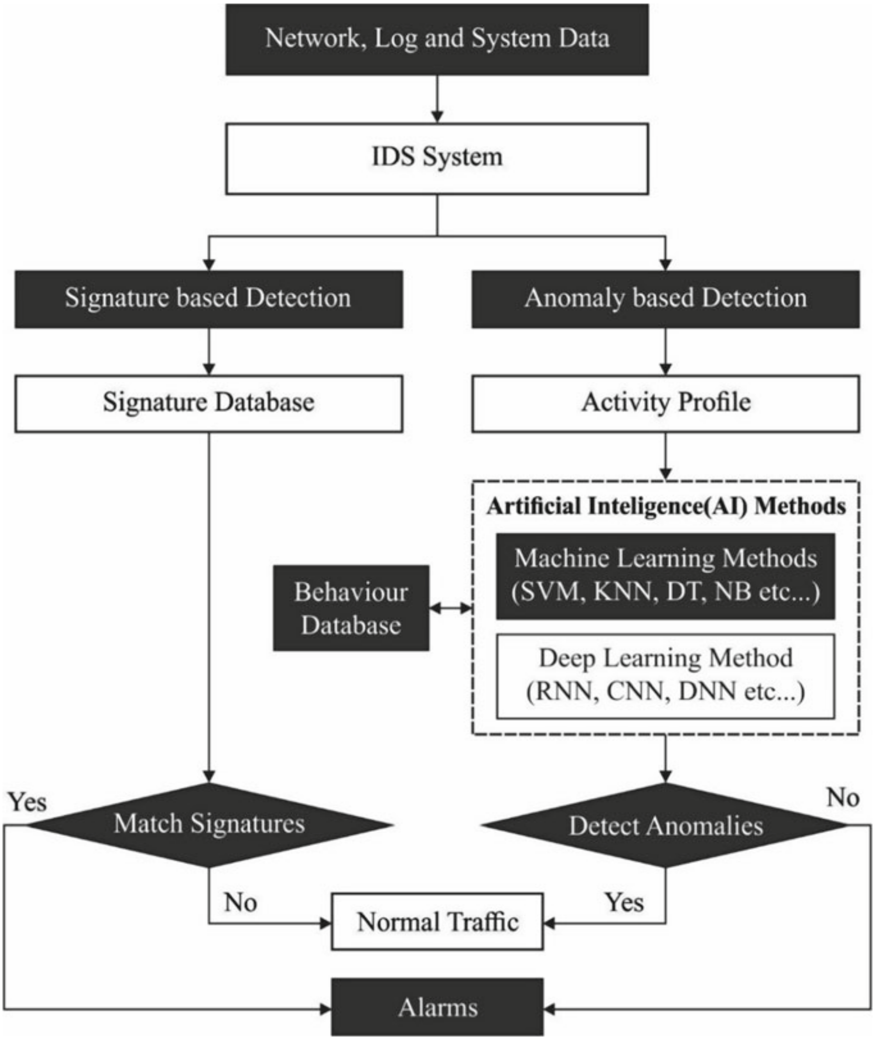


Fig. 1 .

with objectives such as evaluating an effective intrusion detection system model from the different datasets and assessing machine learning based optimization model that will be applied to detect and classify intrusions along with its performance evaluation on parameters such as Precision, Recall, F-Score, and Computational Time and Accuracy.

2 Review of Deep Learning Based IDS Models

In these subsection, a detailed review of various DL-based solutions developed for IDS is shown in Table 1. Farahnakian et al. [11] introduced a DL technique to IDS. This technique utilizes Deep Autoencoder (DAE) as most famous DL technique. The presented DAE technique was trained from a greedy layer-wise fashion for avoiding over-fit and local optimum. Hanselmann et al. [12] regarded a novel unsupervised learning approach called CANet. Familiar to us, an initial DL based IDS that manages the data structure of maximum dimension CAN bus, in which varying message kinds are sent at varying times. This technique was estimated on real and synthetic CAN data. An evaluation with preceding ML techniques demonstrates that CANet exhibits them by an important margin.

In Boukhalfa et al. [13], a novel idea to Network IDS (NIDS)-based LSTM for recognizing menaces and for obtaining long-term memory on them, for stopping a novel attack that is similar to the recent ones, and simultaneously, for containing a single mean to block intrusion. The simulated outcome is proved that novel technique LSTM is more effective, it is efficiently memorizing and differentiate among traffics: normal and attack, from both situations of classifications, binary and multiclassification. Yin et al. [14] presented a DL technique to IDS utilizing recurrent neural networks (RNN-IDS). Also, it can be analyzed the efficiency of the model from binary as well as multiclass classifications, and the amount of neurons and varying learning rate influence the efficiency of the presented technique. It can be related to individuals of J48, ANN, RF, SVM, and other ML techniques presented by the preceding researcher on the benchmark dataset.

Kunang et al. [15] introduced a DL-based IDS utilizing a pre-training approach with deep autoencoder (PTDAE) related to DNN. These techniques are established utilizing hyperparameter optimized methods. This investigation offers an alternative solutions to DL framework techniques with automatic hyperparameter optimized procedure which relates grid as well as random search approaches. The automated hyperparameter optimized technique uses defines the value of hyperparameter and an optimum categorical hyperparameter structure for improving detection efficiency. Fatani et al. [16] examined an effective AI based procedure to IDS from IoT systems. It can control the progress of DL and metaheuristics (MH) techniques which permitted its efficacy from resolving difficult engineering issues. It can be present the feature removal technique utilizing the CNNs for extracting relevant features. Besides, it can progress a novel FS technique utilizing a novel different of transient search optimization (TSO) technique is named TSODE, utilizing the operator of differential evolution (DE) technique.

In Kanna and Santhi [17], a precise IDS technique was presented by utilizing a unified method of optimized CNN (OCNN) and hierarchical multi-scale LSTM (HMLSTM). The presented IDS method carries out the pre-processed, feature removal with testing and training of network and last classification. In this method, the lion swarm optimization (LSO) was utilized for tuning the hyperparameter of CNN to an optimum configuration of learning spatial features. Aleesa et al. [18]

Table 1 Comparison of different DL-based IDS models

References	Year	Objective	Methodology used	Dataset used	Performance measures
Farahnakian et al. [11]	2018	To detect intrusions and avoid overfitting	DAE	KDD-CUP'99 dataset	Accuracy, detection rate and false alarm rate
Hanselmann et al. [12]	2020	To design unsupervised IDS for CAN	CANet	Real and synthetic CAN data	Accuracy, TPR/TNR
Boukhalfa et al. [13]	2020	To detect and block intrusions in the network	LSTM	NSL KDD	Accuracy, sensitivity, false positive rate, precision, and recall
Yin et al. [14]	2017	To design an IDS for binary and multi-class classification	RNN-IDS	NSL-KDD dataset	Accuracy, detection rate, and FPR
Kunang et al. [15]	2021	To develop IDS with hyperparameter optimization process	PTDAE-DNN	NSL-KDD, and CSE-CIC-ID2018	Precision, recall, accuracy, *F*-score, training time
Fatani et al. [16]	2021	To introduce a AI based solution for IDS	CNN + TSODE	KDDCup-99, NSL-KDD, BoT-IoT, and CICIDS-2017	Precision, recall, accuracy, *F*-score
Kanna and Santhi [17]	2021	To devise automated IDS using spatio and temporal features	OCNN + HMLSTM	NSL-KDD, ISCX-IDS, and UNSWNB15	Accuracy
Aleesa et al. [18]	2021	To improvise the UNSW-NB15 dataset for DL models	Deep-IDS	Improved UNSW-NB15 dataset	Accuracy
Lee and Park [19]	2021	To address imbalance data classification problem for IDS	GAN	CICIDS 2017 dataset	Precision, recall, accuracy, *F*-score

(continued)

Table 1 (continued)

References	Year	Objective	Methodology used	Dataset used	Performance measures
Liu et al. [20]	2021	To design a cascaded IDS with distributed *K*-means, RF, and DL models	CNN, LSTM + ADASYN	NSL-KDD and CIS-IDS2017 datasets	Accuracy, TPR, time cost
Ullah and Mahmoud [21]	2021	To devise a new DL based anomaly identification model for IoT environment	CNN	BoT-IoT, IoT network intrusion, MQTT-IoT-IDS2020, and IoT-23 intrusion detection datasets	Precision, recall, accuracy, *F*-score
Aldallal and Alisa [22]	2021	To develop a hybrid IDS for cloud platform	GA + SVM	CICIDS2017 dataset	Accuracy, detection rate
Abusitta et al. [23]	2019	To present a co-operative IDS for cloud platform	DA + DNN	KDD Cup'99 dataset	Accuracy
Zhou et al. [24]	2021	To propose a GNN based IDS for IoT environment	HAA with GNN + RWR	UNSW-SOSR2019	Precision
Al Jallad et al. [25]	2019	To project a DL based IDS model for big data	LSTM	MAWI dataset	Detection rate
Mighan and Kahani [26]	2021	To develop a hybrid ML and DL model for IDS	Stacked autoencoder	UNB ISCX 2012 dataset	accuracy, *f*-measure, sensitivity, precision, and time
Vinayakumar et al. [27]	2019	To develop an intelligent IDS model using DNN	DNN	NSL-KDD, UNSW-NB15, Kyoto, WSN-DS, and CICIDS 2017	Precision, recall, accuracy, *F*-score, TPR, ROC
Kasongo and Sun [28]	2020	To design a new IDS for wireless networks	FFDNN + WFEU	UNSW-NB15	Accuracy
Shone et al. [29]	2018	To introduce a DL model for NIDS	NDAE	KDD Cup'99 and NSL-KDD dataset	Precision, recall, *F*-score, accuracy, and time

(continued)

Table 1 (continued)

References	Year	Objective	Methodology used	Dataset used	Performance measures
Kasongo and Sun [30]	2019	To present a DL model with feature selection scheme for IDS	FFDNN	NSL-KDD dataset	Accuracy, precision-recall curve
Hu et al. [31]	2021	To design sensitive IDS model	IDSDL + CSI + DNN	Own dataset	Accuracy
Mendonça et al. [32]	2021	To develop a fast DL model for IDS	Tree-CNN hierarchical model	CICIDS2017 dataset	Detection accuracy, execution time
Toldinas et al. [33]	2021	To project NIDS model using multi-stage DL	ResNet50	UNSW-NB15 and BOUN Ddos	Precision
Khan [34]	2021	To introduce a hybrid IDS model	HCRNNIDS	CSE-CIC-DS2018 dataset	Precision, recall, $F1$-score, and DR
Ashiku and Dagli [35]	2021	To present a NIDS model using DL concepts	DNN	UNSW-NB15 dataset	Accuracy, detection rate
Wani and Khaliq [36]	2021	To provide a SDN based DL model for IDS	DL classifier	IDSIoT-SDL	Sensitivity, detection rate, accuracy, FAR
Jothi and Pushpalatha [37]	2021	To develop IDS for IoT	LSTM	CIDDS-001, UNSWNB15, and KDD datasets	Accuracy, precision, and recall
Haghighat and Li [38]	2021	To project voting based DL model	VNN	KDDCUP'99 and CTU-13	Precision, recall, F-score, accuracy, FPR, and FNR
Yousefnezhad et al. [39]	2021	To design ensemble model for IDS	KNN + SVM + DL model	UNSW-NB15, CICIDS2017, and NSL-KDD	Precision, recall, accuracy, F-score
Mayuranathan et al. [40]	2021	To detect DDoS in cloud environment	RHS model + RBM	KDD'99 dataset	Sensitivity, specificity, Kappa, accuracy, F-score

enhance UNSW-NB15 data set that utilized with DL as old ML approaches are taken as much time and the size of data set is not influence the efficiency of ML approaches, but the size of utilized data set affects the efficacy of DL approaches.

Lee and Park [19] resolved data imbalance by utilizing the generative adversarial networks (GAN) technique that is an unsupervised learning approach of DL that created novel virtual data same as recent data. It is also presented a model which is classified as RF for identifying detection efficiency after addressing data imbalance dependent upon GAN. In Liu et al. [20], an ID technique that relates ML with DL was presented. This technique utilizes the *k*-means and RF techniques as classifiers, and distributed computing of these techniques are executed on Spark platform for rapidly classifying normal as well as attack events. Next, with utilizing the CNN, LSTM, and other DL techniques, the event judged as abnormal is more classified as to varying attack type lastly. Currently, adaptive synthetic sampling (ADASYN) was selected for solving the unbalanced data set.

Ullah and Mahmoud [21] develop and design an anomaly-based IDS for IoT networks. Firstly, a CNN method is applied for creating a multi-class classification method. Then, the presented method is carried out through CNN models in 1D, 2D, and 3D. The generation and processing of features focus on the actual network traffic flow. They developed four data sets using this approach and then integrate them by rising the number of attack classes. Aldallal and Alisa [22], proposed a ML-based hybrid IDS. We integrated SVM and GA methods with an advanced fitness function designed for evaluating performance of the system. This scheme was investigated by the CICIDS2017 data set that has common and normal attacks. These two GA and SVM algorithms have been implemented in parallel to attain two ideal objectives: obtain the optimal set of features with the highest performance.

Abusitta et al. [23] present a ML-based IDS that effectively employs the past feedback data to make decisions. Especially, the presented method is depending on a Denoising Autoencoder (DA), i.e., employed as a fundamental element to create a DNN system. The power of DA exists in its capacity of learning the way to recreate IDS feedback from partial feedback. It enables to periodically decide suspected intrusion without comprehensive feedback from the IDS. Zhou et al. [24], introduced a hierarchical adversarial attack (HAA) technique, targeting the graphical neural network (GNN)-related IDS in the IoT system constrained resources. A hierarchical node election method based random walk with restart (RWR) is designed for selecting a group of susceptible nodes with higher attack importance, with the consideration of the overall loss changes and structural features within the targeted IoT networks.

Al Jallad et al. [25] present a resolution to identify new threats with lower false positive and high detection rates than previously employed IDS, as well as identify contextual and collective security attacks. They attain outcomes through networking Chatbot, a Deep RNN: LSTM on topmost Apache Spark Framework which contains input of traffic aggregation and traffic flow and the outcome is a language of two words, abnormal or normal. Mighan and Kahani [26] present a hybrid system that integrates the benefits of ML and deep network systems. At first, SAE system is utilized for latent feature extraction, which follows various classification-based IDSs, like SVM, RF, DT, and NB that is utilized for efficient and fast detection of intrusion in large network traffic data.

In Vinayakumar et al. [27], a DNN system, a kind of DL method, is used to design an effective and flexible IDSs for detecting and classifying unpredictable and unforeseen cyber-attacks. The rapid changes in network behavior and continues development of attacks make it essential for evaluating different data sets that are produced by the dynamic and static models. This kind of study facilitates the finding of the optimal model that could perform efficiently in identifying upcoming cyber-attacks. Kasongo and Sun [28] present a FFDNN wireless IDSs with a wrapper based feature extraction unit (WFEU). The WFEU employs the extra trees model for creating a best possible feature vector. The efficacy and efficiency of the WFEU-FFDNN are examined according to the AWID and UNSW-NB15 IDS data sets.

Shone et al. [29] proposed a Dl method for IDS called non-symmetric deep auto-encoder (NDAE). Moreover, the authors introduced DL classification technique via stacked NDAE. The presented classification method was executed in GPU-assisted TensorFlow and estimated by the standard NSL-KDD and KDD Cup'99 data sets. Kasongo and Sun [30] designed an IDS based DL method using FFDNN together with a filter-based feature selection method. The presented method can be estimated by the popular data mining (NSL-KDD) and NSL-knowledge discovery datasets and it is compared with the current ML algorithms.

Hu et al. [31] presented an IDS-based DL (IDSDL) with fine grained channel state information (CSI) for free the AP place. In CSI stage propagation modules decomposition technique was implemented for obtaining blurred elements of CSI stage on various paths as further sensitive detection signals. In CNN of DL was utilized for enabling the computer for learning as well as detecting intrusion without removing numerical features. In Mendonça et al. [32], a novel IDS dependent upon Tree-CNN hierarchical technique with Soft-Root-Sign (SRS) activation purpose was presented. To performance assessment, this method was executed in medium-sized company, analyzing the level of difficulty of presented solutions.

Toldinas et al. [33] present a novel manner to network ID utilizing multistage DL image detection. The network feature was altered as to 4 channel (Alpha, Blue, Red, and Green) images. Images are utilized for classification for testing as well as training the pretrained DL technique ResNet50. In Khan [34], a convolutional recurrent neural network (CRNN) was utilized for creating a DL-based hybrid ID structure that forecasts and categorizes malicious cyberattacks from the networks. Ashiku and Dagli [35] presents utilize of DL structures for developing a resilient and adaptive network IDS for detecting as well as classifying network attacks. The emphasis is DL or DNNs are enable flexible IDS with learning capacity for detecting and novel or zero-day network behavioral feature, consequently emitting the system intruders and decreasing the risks of compromises. In Wani and Khaliq [36], an SDN based IDS was presented that utilizes DL classifier to detection of anomaly from IoT. The presented IDS doesn't burden the IoT device with security profile. The presented work was implemented in simulated environments. The outcomes of the experimental test are estimated utilizing different matrices and related to other relevant techniques.

In Jothi and Pushpalatha [37], a novel IDS was presented utilizing powerful DL techniques. Motivated by LSTM benefits, whale integrated LSTM (WILS) network was presented for designing intelligent IDS for detecting the range of distinct states of threat on IoT networks. The system includes four important functions: (i) Data gathering unit that profiles the regular efficiency of IoT device linked from the network, (ii) identify the malicious device on the networks if an attack was happening, (iii) forecasts the kind of attacks utilized from the network. In Haghighat and Li [38], a novel voting-based DL structure is named VNN has presented for taking the benefits of some types of DL frameworks. Regarded as many methods generated by distinct features of data and many DL frameworks, VNN offers the capability for aggregating optimum methods for creating further accurate and robust outcomes. So, VNN uses security specialists for detecting further difficult attacks.

Yousefnezhad et al. [39] implement ensemble methods for enhancing the efficiency of ID and simultaneously, reduce the FAR. It can be utilized *k*NN tor multi-class classifier as well as SVM for approaching the classification issue from normally based detections. For combining several outcomes, it can be the Dempster–Shafer approach in that there are possibilities of explicit retrieval of uncertainty. In addition, it can be employed DL to remove features for training the instances, chosen by the instance selective technique dependent upon ensemble margin. Mayuranathan et al. [40] present an effective feature subset selection-based classification method to detect DDoS attacks. For detecting the DDoS attacks from IDS, an optimum feature set was chosen with maximal detection by utilizing of random harmony search (RHS) optimized method. If the features were chosen, a DL-based classifier method utilizing RBM was executed for detecting the DDoS.

3 Performance Analysis

This section inspects the recently developed DL models for IDS available in the literature. Table 2 offers the comparative analysis of the DL models interms of different measures. Figure 2 investigates the prec_n, reca_l, and F_{score} analysis of the DL models. The figure reported that the DNN model has obtained lower prec_n, reca_l, and F_{score} values of 0.81, 0.785, and 0.765, respectively. At the same time, the CCNN model has obtained slightly enhanced prec_n, reca_l, and F_{score} of 0.815, 0.791, and 0.76. Moreover, the conditional VAE, FNN, and MLESM-RNN techniques have resulted in moderate prec_n, reca_l, and F_{score} values. Furthermore, the RNN model has tried to accomplish merate prec_n, reca_l, and F_{score} of 0.831, 0.833, and 0.82, respectively. However, the DAE-DNN model has resulted in higher prec_n, reca_l, and F_{score} of 0.86, 0.833, and 0.82, respectively.

The accu_y analysis of the DL models is carried out in Fig. 3. The figure reported that the CCNN and DNN models have gained lower accu_y values of 0.791 and 0.785. In line with, the FNN, conditional VAE, RNN, and MLESM-RNN techniques have obtained moderate accu_y values of 0.803, 0.801, 0.813, and 0.804, respectively. But, the DAE-DNN technique has resulted in a maximum accu_y of 0.833.

Table 2 Comparative analysis of DL technique for IDS interms of different measures

Methods	Precision	Recall	Accuracy	$F1$-score
CCNN	0.815	0.791	0.791	0.76
FNN	0.821	0.803	0.803	0.815
RNN	0.831	0.813	0.813	0.793
Conditional VAE	0.816	0.801	0.801	0.791
Deep neural network	0.81	0.785	0.785	0.765
MLESM-RNN	0.819	0.804	0.804	0.792
DAE-DNN	0.86	0.833	0.833	0.82

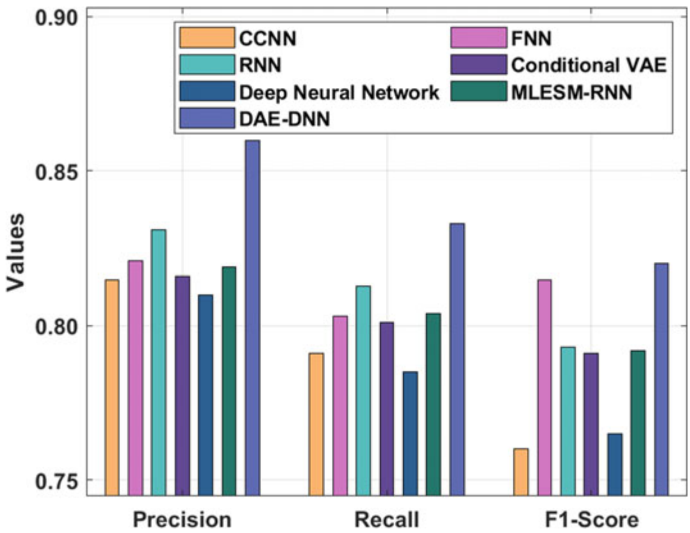


Fig. 2 Comparative analysis of DL techniques with varying measures

4 Challenges and Future Developments

The usage of an appropriate data set is one of the major problems in the development of deep learning-based IDS. The presented method doesn't offer reliable performance result, because they are based on the NSL- or KDD KDD99 benchmark data sets, that has older traffic, don't have real-time properties, and don't characterize current traffic behaviors and attack scenarios. Hence, attaining traffic from simulated environment could overcome this problem by investigating current data sets, like the N-BaIoT IoT [41], and CICIDS2017 IDS intrusion prevention system (IPS) datasets [42]. Also, the published dataset is available for several fields, like industrial control systems (ICS). As well, the comparison between distinct DL approaches that are carried out in isolation doesn't offer a reasonable comparison based on efficiency and effectiveness. This is because of differences in: (1) preprocessing (2) deep

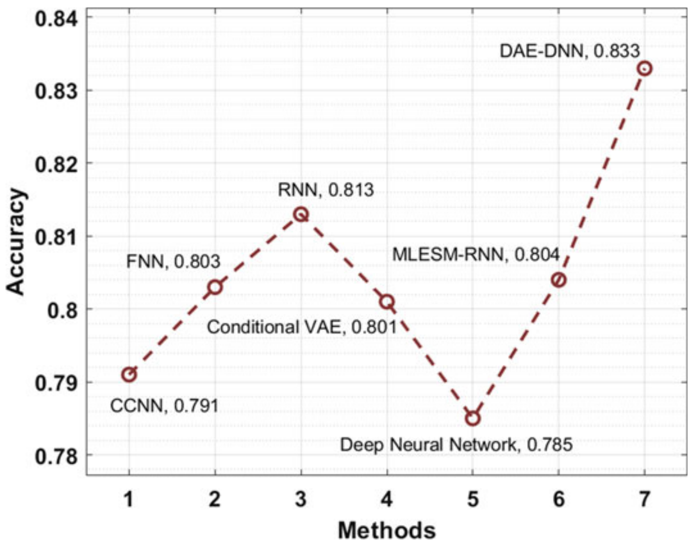


Fig. 3 Accuracy analysis of DL technique

network configuration, (3) hardware platforms (4) the used dataset, and (5) part of the data set, viz., adapted. Hence, it is necessary for further comparative analysis that uses common affecting factors and unified computing platforms for distinct DL frameworks to attain a reasonable result.

Yet, the DL methods does not cover intrusion detection in different fields. Thus, it is essential to reconsider the IDS problems in several fields like smart grids, 5G, several IoT platforms, and SCADA that have been previously analyzed by shallow ML and other anomaly detection methods. Extensibility to distinct fields requires a data set that truly reflects the targeted environments and attains remarkable outcomes. Several DL-based IDS based on GPUs and CPUs for intense off-line trained computation. In response to tremendous growth, chip vendor has created innovative AI accelerator; the AI chip markets are predicted to attain $66.3 billion in 2025 [43]. The more commonly used chips are application-specific integrated circuit (ASIC), the field programmable gate array (FPGA), and the neural network processing unit (NNPU), along with the Edge TPU, a tiny AI accelerator published in 2018 by Google for IoT device. Today's IoT devices and smartphones are armed with these innovative chips. Thus, leveraging this development to perform study will generate real-time prototype, instead of trusting offline data sets. Additionally, it will permit the advancement of innovative IDS for the restricted devices. Additional research of hybrid DL frameworks like GAN model is essential. It is valuable for leveraging DL method to change from collaborative IDS to collaborative DL-based IDS.

5 Conclusion

With the emergence of advanced technologies and the drastic increase in data generation, several research communities have investigated the design of DL models to detect intrusions. This survey extensively analyses and investigates the different DL based IDS models available in the literature. The reviewed methods are examined in different ways such as objectives, methodology used, datasets used, and performance measures. In addition, a comparative and descriptive analysis of the surveyed DL-based IDS models takes place by offering a side-by-side comparison in a tabular form. Then, the performance analysis of the different DL models takes place and identifies the optimal solutions. Lastly, the open research issues and future scope of the research are highlighted.

References

1. Sadreazami H, Mohammadi A, Asif A, Plataniotis KN (2018) Distributed-graphbased statistical approach for intrusion detection in cyber-physical systems. IEEE Trans Sig Inf Process Netw 4(1):137–147
2. Bhuyan MH, Bhattacharyya DK, Kalita JK (2014) Network anomaly detection: methods, systems and tools. IEEE Commun Surv Tutor 16(1):303–336
3. Shafi K, Abbass HA (2013) Evaluation of an adaptive genetic-based signature extraction system for network intrusion detection. Pattern Anal Appl 16(4):549–566
4. Pasqualetti F, Dörfler F, Bullo F (2013) Attack detection and identification in cyber-physical systems. IEEE Trans Autom Control 58(11):2715–2729
5. Meshram A, Haas C (2017) Anomaly detection in industrial networks using machine learning: a roadmap. In: Beyerer J, Niggemann O, Kühnert C (eds) Machine learning for cyber physical systems: selected papers from the international conference ML4CPS 2016. Springer, Berlin, pp 65–72
6. Hoque MAM, Bikas MAN (2012) An implementation of intrusion detection system using genetic algorithm. Int J Netw Secur Appl 4:2
7. Creech G, Hu J (2014) A semantic approach to host-based intrusion detection systems using contiguous and discontiguous system call patterns. IEEE Trans Comput 63(4):807–819
8. Alazab A, Hobbs M, Abawajy J, Khraisat A, Alazab M (2014) Using response action with intelligent intrusion detection and prevention system against web application malware. Inf Manag Comput Secur 22(5):431–449
9. Chebrolu S, Abraham A, Thomas JP (2005) Feature deduction and ensemble design of intrusion detection systems. Comput Secur 24(4):295–307
10. Koc L, Mazzuchi TA, Sarkani S (2012) A network intrusion detection system based on a hidden Naïve Bayes multiclass classifier. Exp Syst Appl 39(18):13492–13500
11. Farahnakian F, Heikkonen J (2018) A deep auto-encoder based approach for intrusion detection system. In: 2018 20th international conference on advanced communication technology (ICACT). IEEE, pp 178–183
12. Hanselmann M, Strauss T, Dormann K, Ulmer H (2020) CANet: an unsupervised intrusion detection system for high dimensional CAN bus data. IEEE Access 8:58194–58205
13. Boukhalfa A, Abdellaoui A, Hmina N, Chaoui H (2020) LSTM deep learning method for network intrusion detection system. Int J Electr Comput Eng 10(3):2088–8708
14. Yin C, Zhu Y, Fei J, He X (2017) A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access 5:21954–21961

15. Kunang YN, Nurmaini S, Stiawan D, Suprapto BY (2021) Attack classification of an intrusion detection system using deep learning and hyperparameter optimization. J Inf Secur Appl 58:102804
16. Fatani A, Abd Elaziz M, Dahou A, Al-Qaness MA, Lu S (2021) IoT intrusion detection system using deep learning and enhanced transient search optimization. IEEE Access 9:123448–123464
17. Kanna PR, Santhi P (2021) Unified deep learning approach for efficient intrusion detection system using integrated spatial-temporal features. Knowl Based Syst 226:107132
18. Aleesa A, Younis MOHAMMED, Mohammed AA, Sahar N (2021) Deep-intrusion detection system with enhanced unsw-Nb15 dataset based on deep learning techniques. J Eng Sci Technol 16(1):711–727
19. Lee J, Park K (2021) GAN-based imbalanced data intrusion detection system. Pers Ubiquit Comput 25(1):121–128
20. Liu C, Gu Z, Wang J (2021) A hybrid intrusion detection system based on scalable *K*-means+ random forest and deep learning. IEEE Access 9:75729–75740
21. Ullah I, Mahmoud QH (2021) Design and development of a deep learning-based model for anomaly detection in IoT networks. IEEE Access 9:103906–103926
22. Aldallal A, Alisa F (2021) Effective intrusion detection system to secure data in cloud using machine learning. Symmetry 13(12):2306
23. Abusitta A, Bellaiche M, Dagenais M, Halabi T (2019) A deep learning approach for proactive multi-cloud cooperative intrusion detection system. Futur Gener Comput Syst 98:308–318
24. Zhou X, Liang W, Li W, Yan K, Shimizu S, Kevin I, Wang K (2021) Hierarchical adversarial attacks against graph neural network based IoT network intrusion detection system. IEEE Int Things J
25. Al Jallad K, Aljnidi M, Desouki MS (2019) Big data analysis and distributed deep learning for next-generation intrusion detection system optimization. J Big Data 6(1):1–18
26. Mighan SN, Kahani M (2021) A novel scalable intrusion detection system based on deep learning. Int J Inf Secur 20(3):387–403
27. Vinayakumar R, Alazab M, Soman KP, Poornachandran P, Al-Nemrat A, Venkatraman S (2019) Deep learning approach for intelligent intrusion detection system. IEEE Access 7:41525–41550
28. Kasongo SM, Sun Y (2020) A deep learning method with wrapper-based feature extraction for wireless intrusion detection system. Comput Secur 92:101752
29. Shone N, Ngoc TN, Phai VD, Shi Q (2018) A deep learning approach to network intrusion detection. IEEE Trans Emerg Top Comput Intell 2(1):41–50
30. Kasongo SM, Sun Y (2019) A deep learning method with filter-based feature engineering for wireless intrusion detection system. IEEE Access 7:38597–38607
31. Hu Y, Bai F, Yang X, Liu Y (2021) IDSDL: a sensitive intrusion detection system based on deep learning. EURASIP J Wirel Commun Netw 2021(1):1–20
32. Mendonça RV, Teodoro AA, Rosa RL, Saadi M, Melgarejo DC, Nardelli PH, Rodríguez DZ (2021) Intrusion detection system based on fast hierarchical deep convolutional neural network. IEEE Access 9:61024–61034
33. Toldinas J, Venčkauskas A, Damaševičius R, Grigaliūnas Š, Morkevičius N, Baranauskas E (2021) A novel approach for network intrusion detection using multistage deep learning image recognition. Electronics 10(15):1854
34. Khan MA (2021) HCRNNIDS: hybrid convolutional recurrent neural network-based network intrusion detection system. Processes 9(5):834
35. Ashiku L, Dagli C (2021) Network intrusion detection system using deep learning. Proc Comput Sci 185:239–247
36. Kavitha T, Mathai PP, Karthikeyan C et al (2021) Deep learning based capsule neural network model for breast cancer diagnosis using mammogram images. Interdiscip Sci Comput Life Sci. https://doi.org/10.1007/s12539-021-00467-y
37. Cyril CPD, Beulah JR, Subramani N, Mohan P, Harshavardhan A, Sivabalaselvamani D (2021) An automated learning model for sentiment analysis and data classification of Twitter data using balanced CA-SVM. Concurr Eng Res Appl 29(4):386–395

38. Reshma G, Al-Atroshi C, Nassa VK, Geetha B et al (2022) Deep learning-based skin lesion diagnosis model using dermoscopic images. Intell Autom Soft Comput 31(1):621–634
39. Bhukya RR, Hardas BM, Anil Kumar TC et al (2022) An automated word embedding with parameter tuned model for web crawling. Intell Autom Soft Comput 32(3):1617–1632
40. Wani A, Khaliq R (2021) SDN-based intrusion detection system for IoT using deep learning classifier (IDSIoT-SDL). CAAI Trans Intel Tech
41. Berlin MA, Tripathi S et al (2021) IoT-based traffic prediction and traffic signal control system for smart city. Soft Comput. https://doi.org/10.1007/s00500-021-05896-x
42. Haghighat MH, Li J (2021) Intrusion detection system using voting-based neural network. Tsinghua Sci Technol 26(4):484–495
43. Yousefnezhad M, Hamidzadeh J, Aliannejadi M (2021) Ensemble classification for intrusion detection via feature extraction based on deep Learning. Soft Comput 25(20):12667–12683

Intelligent Systems for Social Welfare I

A Comprehensive Comparison Between Pre-trained and Custom Trained Object Detection Model for Indian Traffic Scenarios

Atharva Joshi, Abhishek Shelke, and Bhakti Paranjape

Abstract Artificial intelligence is an emerging field that is undergoing humongous development. Self-driving technology is a major part of that development. Various companies like Tesla, Waymo, Ford, and Audi are part of research and development in this field. The modern self-driving vehicles that work on Level 5 autonomy utilize lidar technology along with computer vision frameworks to get a complete autonomous driving experience. These companies are based in the USA or well-developed countries. Their implementation in India, which is a developing country, is to be determined. To get a fair idea of how these vehicles and their technology will perform in the Indian scenario, a testing plan is devised in this work. In this paper, a YOLO object detection model was custom trained on images taken in the Indian traffic scenario and compared with a pre-trained model which has been trained on COCO dataset which contains traffic images from various foreign countries like the USA and Europe. The comparison has been done on the metrics like accuracy in detection of classes, precision, recall, and $F1$-score.

1 Introduction

The development in autonomous driving technology is increasing day by day. Technology and automotive companies like Tesla, Ford, and Audi are the pioneers in researching and implementing self-driving technology in the industry. Testing the performance of this technology in Indian scenario is the scope of this research. It can be tested using the object detection framework and can be compared with a custom trained model. There are various CNN-based architectures like ResNet [1], faster R-CNN [2], written in Python and C++ (Caffe). They have higher detection quality than R-CNN. Histogram oriented gradients (HOG) [3, 4], which is a feature descriptor, is also used for object detection. It is easier to interpret the information it carries. Single shot detectors (SSD) [5] is also one of the popular techniques used in object detection. It combines detections from multiple feature maps with various

A. Joshi (✉) · A. Shelke · B. Paranjape
MIT World Peace University, Pune, India
e-mail: mjdjatharva@gmail.com

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_17

resolutions to handle objects of various sizes. SSD can be trained and integrated into the systems which require an object detection component. In this paper, you only look once (YOLO) [6], object detection algorithm is implemented. It is written in Python and the framework is DarkNet and is one of the fastest object detection algorithms giving good accuracy. Further, in this paper, YOLO algorithm is explored in detail. The YOLO [6, 7] model discussed in this paper has been custom trained on images taken on the Indian road conditions. The dataset was generated with 1200 images which are annotated based upon the classes. Then the model was trained on the custom dataset till the stop loss nears 2. The testing for this model and the pre-trained model was performed on images taken from the IDD dataset. The results obtained from were analyzed and presented in a graphical format.

2 Object Detection and YOLO

Object detection is one of the most widely used computer vision applications [8, 9], which is used to detect objects like humans, animals, cars, bikes, traffic signals, and more. The primary aim of object detection is to develop models that provide information to the users, such as "What objects are where?". It is used for various computer vision tasks such as image captioning, object detection, and object tracking. More specific to the use cases of object detection includes pedestrian detection, face detection, text detection, number plate detection, and people counting.

Object detection can be performed using two techniques

1. Image processing techniques
2. Deep learning techniques.

YOLOR (mAP 56.1) is currently one of the best detection algorithms for real time. It is based on accuracy and mean average precision and uses the MS COCO [10–12] dataset.

The algorithm is closely followed by YOLOv4 (MAP 55.4) and EfficientDet [12] (MAP 55.1).

Fastest real-time object detection algorithm based on inference time is the YOLOv4 algorithm with 12 ms, which is followed by TTFNet with 18.4 ms. YOLOv3 comes in a close third with 29 ms inference time. The one-stage detector that has been implemented in YOLO has improved inference timings as compared to some of the slower two stage methods such as Mask R-CNN (333 ms).

The convolution neural networks like R-CNN, region-based convolution neural networks, Fast R-CNN, and you only look once (YOLO) are also popularly used for object detection.

YOLO v4 is the successor of the YOLO v3 algorithm architecture, with a significant advancement in the mean average precision (mAP) of up to 10% and 12% improvement in detection speed. The architecture of YOLO v4 consists of four distinct blocks as shown: backbone, neck, dense prediction, and sparse prediction.

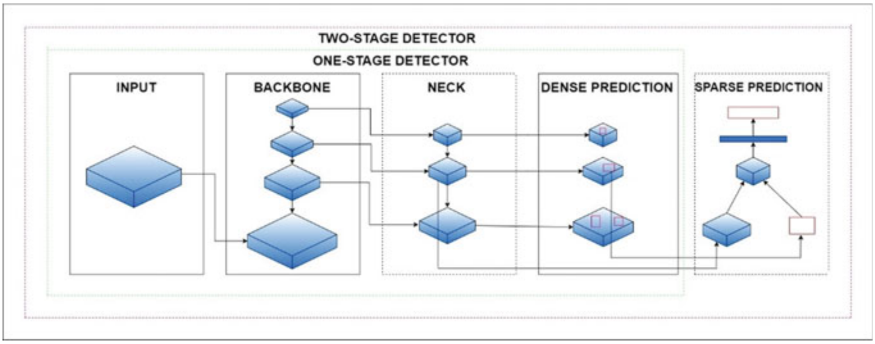


Fig. 1 YOLO v4 distinct blocks: backbone, neck, dense prediction, and sparse prediction

Backbone—CSPDarknet53 stands for cross-spatial-partial connections, it is the backbone shown in Fig. 1, which is the feature extraction architecture in the one-stage detector.

The current layer is divided into two parts, the first part is passed through the convolution layers while the other part is not. Further, both the results are aggregated.

Neck—The addition of layers between the backbone and the dense prediction block (head) is aided by the neck, which is similar to what is done by the ResNet architecture. The YOLOv4 architecture consists of three parts which are modified spatial pyramid pooling, a modified path aggregation network, and a modified spatial attention module, which aggregates the information and improves the accuracy.

Dense predictions (head)—The process of locating the bounding boxes and classification of the object is done by dense prediction. The process is similar to that of YOLOv3 where the bounding box coordinates are detected along with the score. It is used in one-stage detection algorithms such as YOLO and SSD.

Sparse predictions—Mostly used in two stage detection algorithms such as faster R-CNN, FPN, and Mask R-CNN.

The main task that YOLO algorithm performs is to split the image into numerous grids, each of them having equal dimensions and predicting the probability that a given cell contains an object using the defined anchor boxes. The output contains bounding box coordinates and the probabilities of the classes, in vector format. Figure 2 shows the outline of the YOLO v4 architecture.

3 Methodology

To get a fair judgment on how self-driving vehicles designed for other countries might work in India and what improvements they might need to perform efficiently in our country, we need to compare a model which has been trained on foreign data and a model which has been trained on the Indian Dataset. The Indian Dataset was collected and annotated based on eight classes of objects namely human, car, bus,

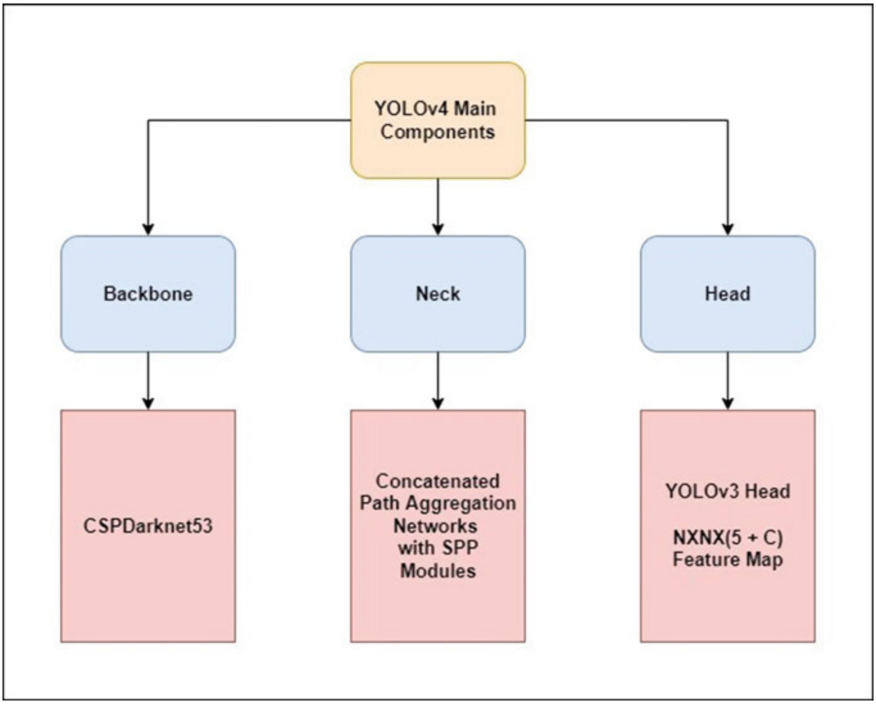


Fig. 2 Basic outline of YOLO v4

truck, animals, rickshaw, cycle, and bike that were chosen as they best represented and differentiated the Indian traffic scenario from other countries. After which the model was trained on the dataset until it reached a stop loss close to 2. The MAP and the IOU achieved during training were noted down, and then, it was compared with the pre-trained model based on comparison metrics such as precision, recall, and $F1$-score.

4 Dataset

The dataset [13] was collected by taking around 1200 images bifurcated into three parts—day, evening, and night to get the most scenarios required by our specifications. Also, additionally, the images from the Indian Driving Dataset (IDD) [14] were used for specific edge cases which would be utilized while comparing the COCO model. Image annotation is the first step in creation of a dataset for custom training of object detection model. Image annotation is the process of labeling images of a dataset to train an object detection model. Training a machine learning model with

Fig. 3 Sample images from the custom dataset used

labeled data is called supervised learning. The labels used for annotation are predetermined and are known as classes. After the training and successful deployment of the model, it is expected that the model identifies those classes as objects in an unannotated image.

Roboflow [15] was used for annotating the dataset with eight classes. The classes are

- bike, car, truck, cycle, human, animal, rickshaw, and bus.

Figure 3 shows some sample images from the custom-built dataset. The annotation data files were saved in the YOLO v4 DarkNet format, ".txt".

For the YOLO v4 [16] config file, specific values were used aligning with the classes.

- Batch size—64
- Subdivisions—16 (The subdivisions can be set to 32 in case of runtime issues)
- Image size—416 × 416 (can be a multiple of 32, increasing the image size may slow down the training process.)
- Max batches—16,000 (number of classes (8) * 2000)
- Steps—12,800, 14,400 (80% and 90% of max batches)
- Filters—39 ((number of classes (8) + 5) * 3)
- Iterations—6000.

Once the average loss is under 2.0, the training is stopped to avoid overfitting and overtraining of the model. The average precision obtained for each class was recorded and is presented in Table 1.

Table 1 Average precision attained during training

Class ID	Class name	Average precision (%)
1	Animal	100.00
2	Bike	81.65
3	Bus	83.88
4	Car	80.02
5	Cycle	57.14
6	Human	77.94
7	Rickshaw	90.39
8	Truck	84.85

Table 2 Compares accuracy for pre-trained and custom trained model

Class	Pre-trained (accuracy)	Custom trained (accuracy)
Human	0.75	0.72
Animal	0.69	0.84
Bike	0.65	0.89
Car	0.69	0.88
Truck	0.78	0.88
Rickshaw	0	0.89
Bus	0.75	0.95
Cycle	0.43	0.65

5 Testing

After training the model on the Indian data, the testing was done on the Indian Driving Dataset (IDD). The results were compared with the pre-trained model.

For testing the model, 30 unique images were selected from IDD, which, were completely new for both models. The detections given by both models for each class were noted. The precision and recall along with the $F1$-score for each class was calculated and is shown in Table 3 for pre-trained model and in Table 4 for custom trained model. The accuracy for each class for both the models is shown in Table 2.

6 Evaluation of Results

The performance of the pre-trained model and the custom trained model was evaluated based upon how accurately they predicted each class along with the precision and recall scores exhibited by them on the test dataset. Table 2 shows the accuracy attained by both the custom trained model and the pre-trained model in detecting the

Table 3 Precision, recall, and $F1$-score for pre-trained model

Class	Precision	Recall	$F1$-score
Human	0.27	0.46	0.34
Animal	0.19	0.16	0.17
Bike	0.67	0.56	0.61
Car	0.60	0.73	0.66
Truck	0.41	0.46	0.43
Rickshaw	0	0	0
Bus	0.22	0.26	0.24
Cycle	0.11	0.13	0.12

Table 4 Precision, recall, and $F1$-score for custom trained model

Class	Precision	Recall	$F1$-score
Human	0.23	0.24	0.23
Animal	0.18	0.16	0.17
Bike	0.71	0.74	0.72
Car	0.47	0.48	0.47
Truck	0.41	0.42	0.41
Rickshaw	0.32	0.32	0.32
Bus	0.16	0.16	0.16
Cycle	0.42	0.44	0.43

classes. It was observed that the accuracy of the custom trained model was higher in all classes except in the human class. The higher accuracy of the pre-trained model in detecting the human class is attributed to, the model being trained on COCO Dataset [17]. The pre-trained model was not able to accurately identify rickshaws, pertaining to the fact that rickshaws are mostly found in the Asian sub-continent and the west rarely sees them. Figure 4 shows the accuracy attained by both the models on the test dataset.

Table 4 shows that the custom trained model was able to detect indigenous vehicles like rickshaw. The ability to detect animals was also better. Table 3 shows FI-score, mean precision and recall for all the classes for pre-trained model.

By examining the tables above, it can be noted that the pre-trained model was precisely able to detect humans, cars, trucks, and buses. It struggled to detect the bike and rickshaw class which points toward the fact that availability of these classes of vehicles in the foreign land is scarce. Thus, as the pre-trained model has not been explicitly trained on these classes it is not able to detect them accurately and precisely. The harmonic mean of precision and recall gives us the $F1$-score. Preferably it should be nearly 1. In Table 4, the highest $F1$-score obtained by the custom trained model is 0.72 for the class bikes and the highest $F1$-score obtained by the pre-trained model is

Fig. 4 Bar chart depicting accuracy attained by both the models for all classes

0.66 for the car class. The visual representation for precision, recall, and the $F1$-score for the pre-trained and the custom trained model is shown in the Figs. 5, 6, and 7, respectively.

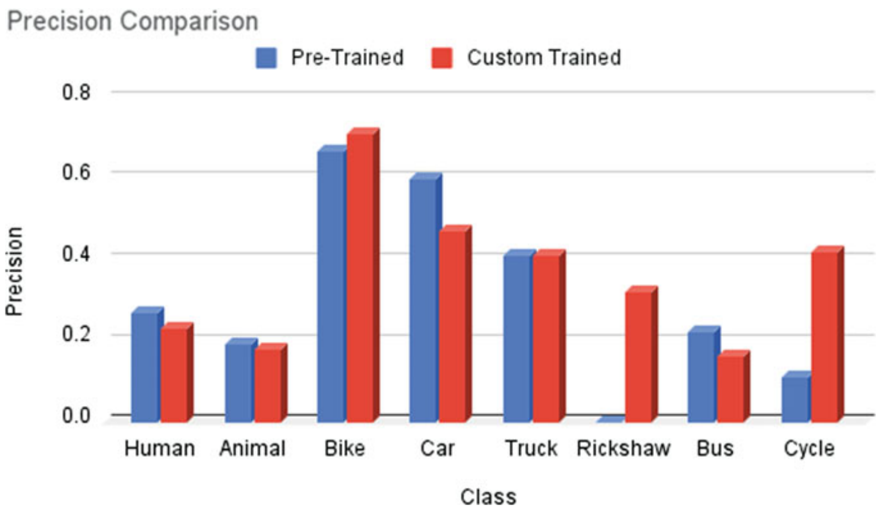


Fig. 5 Bar chart depicting precision attained by both the models for all classes

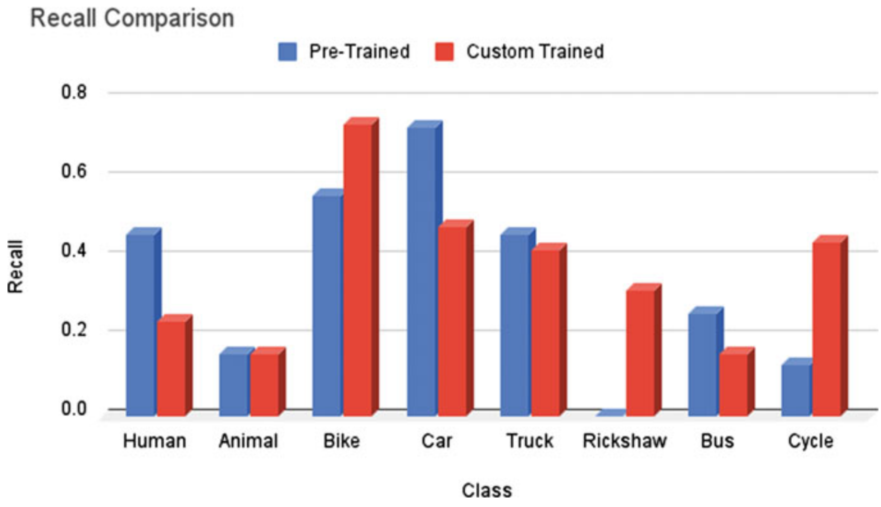


Fig. 6 Bar chart depicting recall attained by both the models for all classes

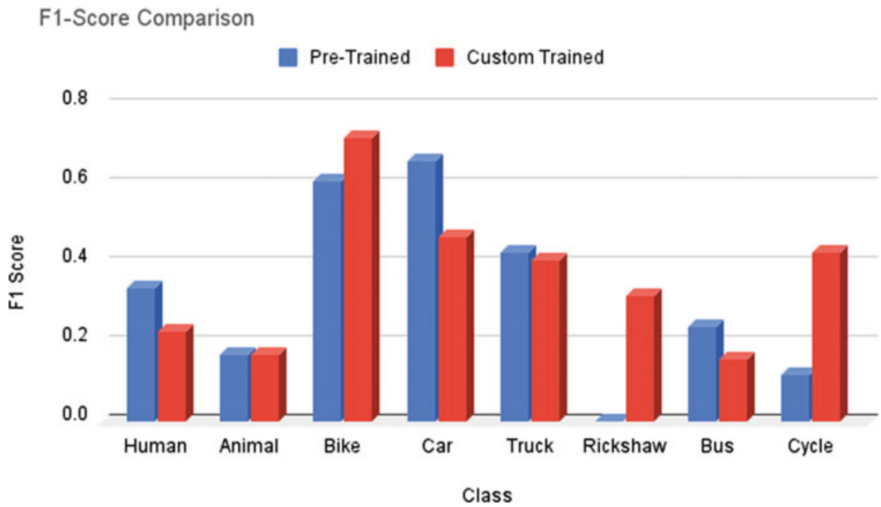


Fig. 7 Bar chart depicting $F1$-score attained by both the models for all classes

7 Conclusion

After evaluating the results of the testing, it has been observed that the pre-trained model performs well in some classes than in others and thus the model needs to be custom trained according to the country where it must be deployed. This will ensure higher accuracy in detection of the classes by the model, and hence, we could also conclude that autonomous cars which are designed for a specific location or country can be used in a different country but the discrepancies in their detection would be life-threatening and hence the model will need to be adjusted to achieve optimum safety and autonomy.

References

1. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Computer Society conference on computer vision and pattern recognition, vol 2016. http://doi.org/10.1109/CVPR.2016.90
2. Sharma V, Mir RN (2019) Saliency guided faster-RCNN (SGFr-RCNN) model for object detection and recognition. J King Saud Univ Comput Inf Sci. https://doi.org/10.1016/j.jksuci.2019.09.012
3. Surasak T, Takahiro I, Cheng CH, Wang CE, Sheng PY (2018) Histogram of oriented gradients for human detection in video. http://doi.org/10.1109/ICBIR.2018.8391187
4. El-Sayed RS, El-Sayed MN (2020) Classification of vehicles' types using histogram oriented gradients: comparative study and modification. IAES Int J Artif Intell 9(4). http://doi.org/10.11591/ijai.v9.i4.pp700-712
5. Li MM, Lei JY, Zhao CJ (2019) An improved objective detection of road scenes based on SSD model. http://doi.org/10.1145/3366194.3366317
6. Redmon J, Farhadi A (2017) YOLO V2.0. Cvpr2017
7. Zuo X, Yu J, Xian T, Hu Y, Hu Z (2021) Road obstacle object detection based on improved YOLO V4. Int J Adv Netw Monit Controls 6(3). http://doi.org/10.21307/ijanmc-2021-023
8. Liu L et al (2020) Deep learning for generic object detection: a survey. Int J Comput Vis 128(2). http://doi.org/10.1007/s11263-019-01247-4
9. Chandan G, Jain A, Jain H, Mohana (2018) Real time object detection and tracking using deep learning and OpenCV. http://doi.org/10.1109/ICIRCA.2018.8597266
10. Solawetz J (2020) An introduction to the COCO dataset. Roboflow
11. Kim D-H (2019) Evaluation of COCO validation 2017 dataset with YOLOv3. J Multi Eng Sci Technol (JMEST) 6(7)
12. Tan M, Pang R, Le QV (2020) EfficientDet: scalable and efficient object detection. http://doi.org/10.1109/CVPR42600.2020.01079
13. Paranjape B (2021) DATS_2020. Mendeley Data, V1. http://doi.org/10.17632/nfc34n8svj.1
14. Varma G, Subramanian A, Namboodiri A, Chandraker M, Jawahar CV (2019) IDD: a dataset for exploring problems of autonomous navigation in unconstrained environments. In: IEEE winter conference on applications of computer vision (WACV 2019)
15. Alexandrova S, Tatlock Z, Cakmak M (2015) RoboFlow: a flow-based visual programming language for mobile manipulation tasks. In: Proceedings—IEEE international conference on robotics and automation, vol 2015. http://doi.org/10.1109/ICRA.2015.7139973
16. Dewi C, Chen RC, Liu YT, Jiang X, Hartomo KD (2021) YOLO V4 for advanced traffic sign recognition with synthetic training data generated by various GAN. IEEE Access 9. http://doi.org/10.1109/ACCESS.2021.3094201
17. Lin T-Y, Patterson G, Ronchi MR, Cui Y, Maire M (2017) COCO dataset. COCO Consortium

Detection of Outdoor Traffic and Kids Playing Scene for Visually Impaired People

Swati Shilaskar, Amey Kadam, Rupesh Kadam, Tanuja Jadhav, Shripad Bhatlawande, and Jyoti Madake

Abstract In this paper, a novel outdoor scene detection technique for visually impaired people has been proposed. This system focuses on the detection of traffic jam scenes and playing kids scenes. Visually impaired people (VIP) face more difficulties walking alone outdoor. As they cannot prevent themselves from unexpectedly hitting objects, avoiding stray animals, and bumping into people. It inspired us to work on a solution for the mobility of blinds. This paper proposes an effective outdoor scene detection system for improving their lifestyle. Traditional solutions are available for visually impaired people like white cane and guide dogs. Traditional solutions address obstacle detection and path guidance, but scene-cognition is somehow a neglected domain. The proposed system can detect two scenes. SIFT and K-means clustering machine learning algorithms have been used in this system along with the following classifiers with accuracy-decision tree (84%), KNN (84.75%), random forest (88.58%), SVC linear kernel (85.08%), SVC polynomial kernel (85.33%), SVC radial basis function kernel (87.08%), and linear regression (85.08%).

Keywords Assistive aid · Visually impaired · Computer vision · Machine learning · Traffic scene detection · Kids playing scene detection

S. Shilaskar (✉) · A. Kadam · R. Kadam · T. Jadhav · S. Bhatlawande · J. Madake
Department of Electronics and Telecommunication, Vishwakarma Institute of Technology, Pune 411037, India
e-mail: swati.shilaskar@vit.edu

A. Kadam
e-mail: amey.kadam20@vit.edu

R. Kadam
e-mail: rupesh.kadam20@vit.edu

T. Jadhav
e-mail: tanuja.jadhav19@vit.edu

S. Bhatlawande
e-mail: Shripad.bhatlawande@vit.edu

J. Madake
e-mail: jyoti.madake@vit.edu

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_18

1 Introduction and Literature Survey

Being able to see the beauty of the world through our eyes is a blessing for a human being. But unfortunately, there are visually impaired people around the world with visual disorders, they may have partial vision loss or complete blindness, or legal blindness. According to the World Health Organization (WHO) report of 2021, at least 2.2 billion people are having near or distance vision impairment and at least 1 billion people have a visual disability due to certain causes like unaddressed refractive error (88.4 million), Cataract (94 million), diabetic retinopathy (3.9 billion), corneal opacities (4.2 million), glaucoma (7.7 million), and trachoma (2 million) as well as effects like near vision impairment caused by unaddressed Presbyopia (826 million) [1]. Considering age statistics of visually impaired people (VIPs), a survey published by John Elflein in the year 2020, 55% of women are living with blindness, this amounts to 24 million- and 75%-men having blindness were 50 years and over which amounts to 19 million [2]. For normally sighted people these are only numbers. Visually impaired people had to face many difficulties just for living a simple life like normal people as they cannot travel long distances particularly independently, traveling alone outdoor might cause them harm unlike roaming in the known surrounding, in their own house. Most of the time they must depend on others physically to do trivial things (clothing, cooking, and seeking). They do not have the freedom of enjoying social life (socializing, shopping, and working) also they find difficulties in reading and writing which result in a lack of self-confidence. Considering all these facts and problems. This paper proposes a novel scene detection system for visually impaired people to promote a better lifestyle.

In today's world, there are different approaches available that take VIPs' lives to next level. White canes are used widely by visually impaired people for walking (detects obstacles on the ground) because of their low cost and lightweight nature. Along with it, there come several disadvantages as it can detect objects only at knee level and which are up to 1-m distance. So, it could be dangerous for users as it cannot detect obstacles at eye level. Some tactile display-based techniques mentioned in [3, 4] and sound beacon system [5] works as a solution for visually impaired people.

The following literature contains different aid for visually impaired people. One of the aids is a haptic device which is similar to the white cane device. Proposed with enhanced sensing part and active handle. It detects objects and makes vibrations on the cane device [6, 7]. A guide dog may also use to detect obstacles in the path while walking. However, these dogs are need training, and trained guide dogs are expensive. Along with it is risky for visually impaired people to depend on another living being for safety. Furthermore, visually impaired people also need to get training to handle and take care of these dogs which are costly, time-consuming, difficult, and guide dogs can detect only short-range obstacles like white canes. Most of the literature includes electronic travel aid for visually impaired people as, Villanueva and Farcy [8] developed an electronic travel aid for blind people which helps them to detect a path by using radiometric calculation, LEDs, and a photodiode for implementation. This system can detect roads and obstacles by opening side panels, front panels, and post. But, these systems have several degradations due to their limitations of capabilities and mobilities. Further, a microwave radar on the white cane is published it used to detect the obstacle in a wider range. After testing this system gives better performance, noise tolerance, and reduced dimensions than previous electronic travel aids [9] along with miniaturization of antennas [10]. Another electronic travel aid was introduced smart instructive device. This device is in the shape of eyeglasses that guide people efficiently which include both depth sensor and ultrasonic sensor for detecting small and transparent obstacles [11]. Likewise, Chang et al. [12] proposed a wearable glass system for visually impaired people which helps to recognize a drug pill using deep learning.

This electronic travel aid can be classified into wearable or portable assistive devices and camera-based or sensor-based devices. Considering portable assistive devices, developed a camera-based portable system that helps blind people to recognize multiple indoor objects at a time. The first step is to have a representation of compact sequence coefficients using compressive sensing (CS) representation. Used a semantic-based image similarity measure for going through a statistical prediction model [13]. In most cases, VIPs find wearable systems as a convenience.

Ramadhan AJ. et al. Proposed a smart wearable system for the convenience of visually impaired people which includes various sensors, micro-controller, Bolt GPS module, and cellular communication, tracking the path and detect the obstacle in front of the person done by the sensors, alerts people in the surrounding and track the location and send a phone message to register mobile phones of family members [14] along with person get alert because of buzzer sound and a wrist vibration on the hand [15]. Another literature contains effective wearable navigation design which helps the user to avoid obstacles and follow the virtual blind road while walking outside and which can also locate the user location, they used SLAM (e.g., ORM SLAM oriented brief slam) system for virtual blind road and for localization instead of GPS based system as it has several disadvantages like it is mostly dependent upon internet connectivity, atmospheric condition, and quality of the device. This system gives VIPs visual hints and sound feedback [16, 17]. Following are camera-based systems, proposed a camera-based automatic recognition system for visually impaired people. This system provides high accuracy, efficiency, robustness, and ease of use as it focuses on targeting the image capture for blind people. Using speeded-up robust features they proposed a component-based framework. This system detects false recognition and guides users for the bill to be recognized based on positive and negative image datasets [18]. Furthermore, developed a system of co-robotic cane

(CRC) which uses the three-dimensional camera. Inactive mode, it gives instructions to users about the desired detection and in passive mode, it works as a white cane with advanced computer vision [19]. Made mobile robot obstacle avoidance system [20]. Most of the systems used RGB-D consumer cameras for navigation systems [21]. For sensor-based assistive devices, sensors would be, ultrasonic sensors [10, 22], infrared sensors [23], sonar modules [24], which keep VIPs safe along with right and quick detection. For making indoor navigator system used range and color sensors that detect distance up to 3 m and larger than 3 m, respectively [10]. But, this typical region sensor has limitations like they don't provide better quality for input. In further as technology gets vast and to overcome this disadvantage of a typical region sensor proposed a binocular vision sensor over typical vision sensors for providing a better quality as input and they also use cloud computing for the further processor which includes detection and requested automatic result [25]. This detection and automation of the system are done by enhanced machine learning algorithm CNN [26, 27], and DCNN [28]. In this enhanced technology mobile-based navigation systems are also invented [29], features with the online map used for navigation [30], similarly, developed a CNN-based DEEP SEE FACE mobile system dedicated to visually impaired people [31]. Along with these all solutions, Training sessions are also conducted for visually impaired people [32], reports about the auditory orientation training system. By based on an acoustic VR along with an HRTF solution. It produces a virtual training environment, and, in this place, visually impaired people can easily walk while listening to sounds such as stores, vehicles, and noise through headphones. This system reproduces sound reflections and insulation, so users can experience both sound location and obstacle detection skills [33].

Scene detection is not much highlighted domain as compared to obstacle detection. Weijian Hu et al. proposed sonification of scene information with three different approaches, i.e., depth image sonification, obstacle sonification and last one is path sonification. Using these three methods they find pathway scene information for guiding VIP [34]. Ruiqi Cheng et al., have proposed the scene recognition and visual localization based system for assistive navigation for visually impaired people. They have developed a unified deep neural network by combining scene description and scene recognition [35]. Tian et al. [36] have proposed a novel dynamic crosswalk scene understanding method that helps to detect the objects and tells where and when to cross the road for visually impaired people. This work contains three sub-datasets. The proposed module distinguishes pedestrian traffic lights (PTL) and vehicle traffic lights (VTL).

Mainly visually impaired people face difficulties when they go outside of the home (crossing road, avoiding street animals, and traffic) so we come up with an effective outdoor scene detection solution which detects a traffic jam scene and kids playing the scene as it will help while walking on the road and they can enjoy the movement by analyzing the kids playing scene, respectively (Table 1).

Table 1 Comparison of techniques discussed in literature

Refs.	Technique	Algorithm	Datasets	Performance
[19]	Pose estimation and object recognition	Gaussian-Mixture-Model (GMM)	Custom made dataset	86.7% accuracy
[21]	RGB-D camera for visual and range information	RANSAC (RAN-dom SAmple Consensus	Custom made dataset	85% precision
[25]	Scene recognition using binocular image sensor	CNN, ResNet	Custom made dataset	76.6% precision
[35]	Scene recognition and scene description	CNN, MobileNet V2, ShuffleNet V2	Places-365	80% precision

There are various scene detection approaches but there still exist some limitations with frequently appearing scenes in day-to-day life. The scene detection-related problems faced by blind people and the limitations of the current aids described in the preceding section encourage the creation of a novel aid presented in this paper. The aid utilizes popular machine learning and computer vision algorithms. The proposed blind assistance system is used as a scene detection with voice alert. The system includes a monocular camera followed by an embedded controller along with a pre-trained machine learning model developed using SIFT and K Means Clustering algorithm and several supervised machine learning classifiers (decision tree, random forest, KNN, SVC, and logistic regression) and compared their performance in this paper.

This paper is organized in three sections named as methodology, result and conclusion. Methodology section includes description of used classifiers with their algorithm and results section gives performance evaluation of each classifier.

2 Methodology

In this paper, a method for scene detection is proposed for blind people. The scenes are traffic and kids playing. This is one of the approaches to overcome the difficulty faced by blind people at outdoor places. Figure 1 illustrates a block diagram of the proposed system in which the camera module is used as input. Input can be an image or video from which images can be extracted as per frame rate per second and given to the controller. A controller can be Raspberry Pie or Jetson nano because they

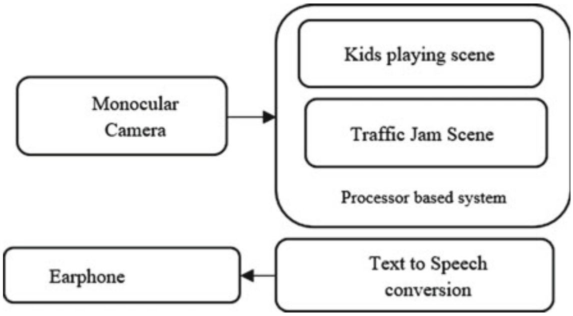
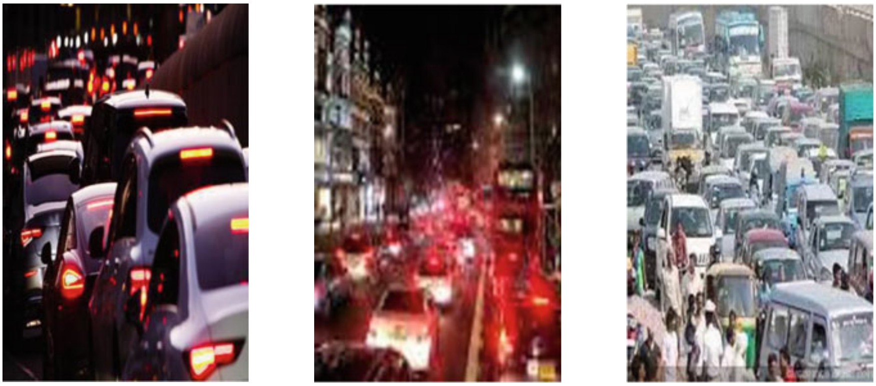


Fig. 1 System level block diagram

support camera modules and have a minimum required memory. For the training of the machine learning model nearly 6000 images are collected out of those 3000 images are of traffic images and 3000 images are of kids playing images. This trained model needs to be stored in the embedded controller and according to the input from the camera the model will predict the scene based on input, frame output is either traffic scene or kids playing which is mapped with the audio signal, and from that output, signal user can be able to understand which scene is currently present in the outside world which helps the user (blind people) by minimizing some difficulty at the time mobility. The methodology for the training module has been discussed below in detail.

2.1 Dataset and Image Pre-processing

The proposed system uses datasets from two different sources for training and testing purposes. Each class contains 3000 images. Approximately 1100 images of dense traffic are used from the Traffic-Net dataset and 1900 images are taken from the Google search results. Also, all 3000 images of the class kids playing are taken from Google search results. The image database has a very diverse set of images and contains images of weather conditions like rainy and sunny days, lighting conditions such as daylight, night, and evening. Figure 2 contains some sample images from the dataset of class traffic. The dataset also includes vehicles of distinct color, shape, size, types, and from different viewpoint angles. To create traffic jams class images of traffic jam scenes taken from a diverse group of cities around the world. This class contains images of Mumbai, Pune, Bangalore, London, Queens, and New York City. The entire dataset is classified into two classes: traffic jams and kids playing manually. Below are some examples of images from the created dataset.



Traffic at night Class = traffic | Traffic on rainy day + night class = traffic | Traffic at daytime class = traffic

Fig. 2 Sample images from traffic dataset with different lighting conditions and density

As mentioned in the previous sections. The camera module captures images, and these images are analyzed by the highly efficient machine learning model (also presented in this paper) this model is developed to recognize distinct scenes. Currently, the proposed system is in the prototyping phase and capable of recognizing only two scenes i.e., kids playing and a traffic jam.

Figure 3 represents pipelined diagram of the proposed work. Images captured from the monocular camera are used for scene detections. Then the captured images are pre-processed after pre-processing generation of the features vector taken place using scale invariant feature transform (SIFT). To eliminate feature bias normalization is applied over the feature vector. SIFT produces a feature vector of $n \times 128$ (n is in order of thousands) those features are clustered together into five clusters (determined by the Elbow method). After creating the k-means clustering model, descriptors of each image are put into five clusters (bins) and in the next step to reduce the dimensions, principal component analysis (PCA) has applied. After dimensional reduction there comes the classification step. There are two phases of the proposed work: training and detection phase. In the training phase, a model to detect the scenes is trained and in the detection phase scene is detected using the trained model. These two phases and their steps are illustrated in Fig. 4.

It represents two main phases: training and testing. In training phase, novel machine learning model is developed to recognize required scenes, and in testing phase model trained from the 1st phase is utilized to recognize the scene

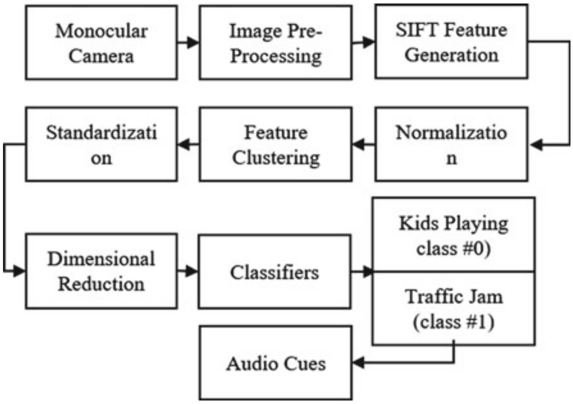


Fig. 3 Pipelined architecture of the proposed system

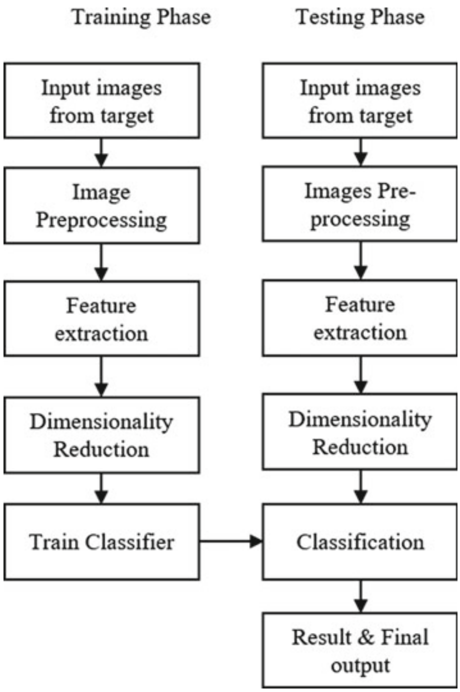


Fig. 4 Training and testing phases of proposed system

Algorithm 1 Creation of master feature vector file

1: **Input**: a path to main folder of dataset (e.g., Traffic)
2: **Output**: master feature vector consisting of feature detected for each image
3: Initializations of variables: $i = 1$, $n =$ total count of images in dataset
4: Create a blank feature vector file (.CSV)
5: **For** (i to n) do // iterate through all images
6: Read ith image
7: Resize image to 150×230
8: Convert resized image to grey scale image
9: Detect SIFT descriptors
10: Normalize detected descriptors
11: Append normalized descriptors to feature vector file
12: Increment i by 1
13: **End for**
14: Repeat from step 1 to 13 for second folder of dataset
15: Merge all feature vector files to create one master feature vector file
16: **End of Algorithm**

2.2 *Feature Vector Extraction*

Feature extraction allows the identification of the unique features of the images. Features from the entire dataset is extracted and a machine learning model is trained to recognize these unique features and map those features with the features from an input image. If a good match is found, then that scene can be classified into two classes (traffic jam and kids playing). Algorithm 1 mentions the entire procedure followed in this work to extract features from both classes and to generate a master feature vector file. And, Fig. 5 illustrates a flowchart for the same algorithm. Here in this work, scale-invariant feature transform abbreviated as SIFT is used. SIFT is very robust to scaling and rotations. SIFT descriptor is used because in the real world, the system will be used by visually impaired people and there is no possibility that they will

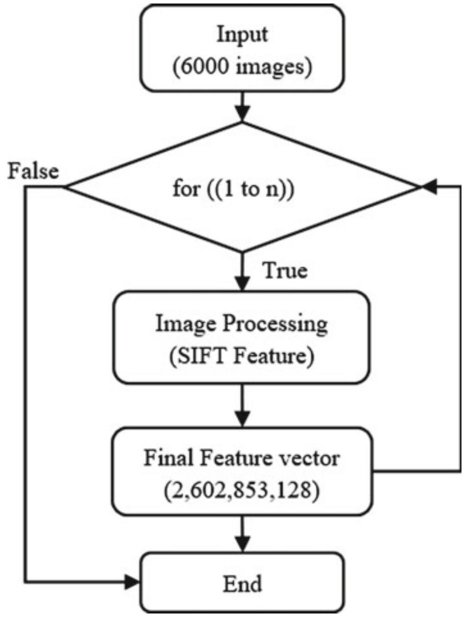


Fig. 5 Flowchart for master feature vector creation

direct the camera system toward the target. The images which the proposed system will be receiving would be scaled and rotated. In that situation also the proposed system should detect features without fail. Therefore, SIFT is highly preferred in this case. As mentioned in Fig. 5. In the training phase, SIFT is applied over the entire dataset consisting of 3000 images of each class. SIFT generates a descriptor vector with 128 columns and several distinct rows depending on the input image. To use the proposed module on low power embedded module, there is a need for dimensionality reduction and the same is explained in the next section.

2.3 Feature Clustering and Dimensionality Reduction

To reduce the number of rows, i.e., unique feature extracted by SIFT of the master feature vector. The proposed system uses k-means clustering. This will reduce the number of rows by grouping similar features into the same clusters. The exact steps involved in this is thoroughly explained through Algorithms 2, 3, and Fig. 6. This was about reducing the number of rows, but the master feature vector also contains 128 columns. To reduce the columns principal component analysis (PCA) is used in the proposed models. After reducing dimensions, the entire feature vector is split into 80/20 for training and testing purposes.

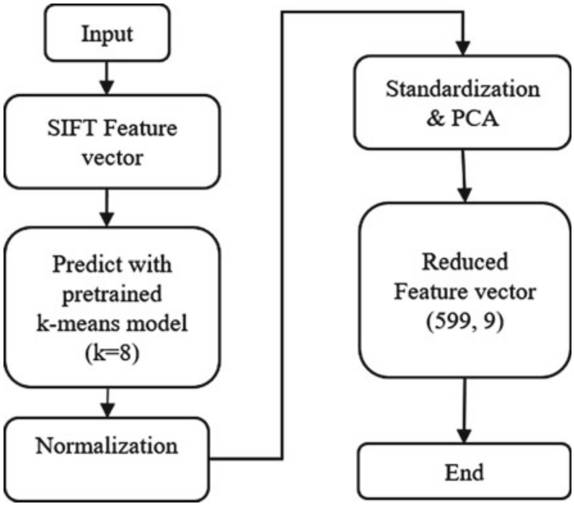


Fig. 6 Flowchart for K-means model and dimensionality reduction

Algorithm 2 Create K-means clustering model

1. **Input**: Master feature vector file (output of algorithm 1)
2. **Output**: K-means trained model for 8 cluster size
3. Apply K-means clustering with 8 clusters on master
4. Feature vector file save output of k-means on the disk
5. **End of algorithm**

Illustrated in Fig. 7 this data is given to different classifiers named as decision tree, K nearest Neighbor, random forest, support vector machine, and linear regression to train the model. In the results, the section performance of their classifier is compared.

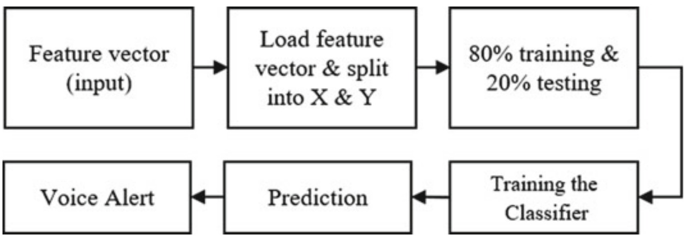


Fig. 7 Flow diagram for recognition and predication

Algorithm 3 K-means predictions and dimensionality reduction

1: **Input**: a path to K-means trained model and a path to dataset (e.g., Traffic)
2: **Output**: master feature vector file with reduced dimensions
3: Initializations of variables: $i = 1$, $n =$ total count of images in dataset
4: **For** (i to n) do
5: Read ith image
6: Resize image to 150×230
7: Covert resized image to grey scale image
8: Detect SIFT descriptors
9: Normalize detected descriptors
10: Predict cluster number of descriptor using K-means model
11: Put the descriptor into predicted cluster/bin
12: Assign class label to descriptor
13: Append descriptors to new feature vector file
14: Increment i by 1
15: **End for**
16: Repeat from step 1 to 15 for second folder of dataset
17: Merge all files generated to one new master feature vector file
18: Apply standardization on new master feature vector File
19: Apply PCA on new master feature vector file
20: Save feature vector with reduced dimensions file to the disk
21: **End of Algorithm**

Algorithm 4 Recognition and prediction of Traffic jam and Kids playing scene

1. Input: feature vector with reduced dimensions file and input image
2. Output: Prediction of Class of input image (0 for Kids playing and 1 for traffic jam)
3. Load the feature vector file
4. Split the feature vector data and labels (last column) from the file
5. Store feature vector into x and labels into y
6. Split data into 80/20 for training and testing
7. Train classifier model
8. Check accuracy and evaluation score of the model
9. Save classifier model onto disk
10. Generate final feature vector for input image
11. Predict class of input image using final feature vector of image and trained classifier model
12. End of Algorithm

2.4 *Classification and Recognition of Traffic Jam and Kids Playing Scene*

After dimensionality reduction and splitting data as 20% for testing and 80% for training there comes this step to train the classifiers to classify the features from the master feature vector file into two classes, i.e., traffic jam and kids playing. The steps involved in this stage are mentioned in Algorithm 4. One of the supervised machine learning is the decision tree it takes decisions by using a set of rules like we human takes a decision. The idea behind the decision tree is that it the split dataset of all points belonging to each class until it gets separated and uses those features to form yes/no questions. Entropy is used to determine the given attribute. It indicates randomness in data. Entropy values range between 0 and 1. If the value of entropy is less, it means that it is trusting able.

$$H(s) = -\text{probablity of} \log_2(q+) - \text{probability of } \log_2(q-) \tag{1}$$

where $(q+)$ is % of positive class and $(q-) =$ % of negative class.

The Random Forest is also a machine learning classifier used to solve classification and regression problems. One of the approaches for solving complicated problems is combining several machine learning algorithms and this is also called ensemble learning. By Bagging and bootstrap aggregating, the random forest algorithm is trained. For improving accuracy in machine learning bagging algorithm is used. This study considers meaning squared error (MSE) at the time of solving regression problems in a random forest algorithm for identifying data branches from every node.

$$\mathrm{MSE} = \frac{1}{M}\sum_{j=1}^{M}\left(f_j - y_j\right)^2 \tag{2}$$

In Eq. 2 M is the number of data points, F_j is the value returned by the model, and y_j is the actual value for data points j.

K-nearest neighbor (KNN) is one of the simplest supervised machine learning techniques. KNN establish similarity between new data and already present data and place new data into the category which has the most similarity between current and available data. For most classification problems in machine learning, KNN is used. At training it stores data and when we give new data then it classifies new data into a category that is similar to new data. In KNN distance between two points in two clusters will be calculated by using Euclidean distance following Eq. (3).

$$a = \sqrt{\left((b2 - b1)^2 + (d2 - d1)^2\right)} \tag{3}$$

Support vector classifier is a subtype of support vector machine (SVM) algorithm. SVM provides more accuracy than logistic regression and decision trees. It is well known for handling nonlinear input spaces. SVM is measurably used for outliers' detection, classification, and regression. SVC is a support vector classifier and its implementation is dependent on libsvm. Based on one versus one scheme it manages multiclass. It uses three types of kernels linear, radial basis function, polynomial kernel decide hyperplane type and that used to separate data. Linear uses linear hyperplane and RBF and Polynomial uses a nonlinear hyperplane.

Logistic regression is also a classification method in machine learning. To module dependent data it uses logistic function and dependent data has only two possible classes, so this technique is used while dealing with binary data. From the linear regression equation, we can obtain a logistic regression equation. In logistic regression, x can be between 0 and 1. The mathematical equation for logistic regression is given in Eq. 4.

$$\log\left[\frac{x}{1-x}\right] = a_0 + a_1 y_1 + a_2 y_2 + a_3 y_3 + \cdots + a_n y_n \tag{4}$$

3 Result

Performance Evaluation

All the machine learning classifiers used are compared by their performance evaluation. From Table 2, it is seen that the random forest classifier provides the greatest accuracy which is 88.58% for the problem statement under study. Second, the highest accuracy is 87.08% provided by SVC with radial basis function kernel (Fig. 8).

There are some evaluation parameters are available in machine learning out of that table contains a precision, recall, and $f1$ score for performance evolution. Precision is the ratio between true positive and total predicated positive means precision gives analysis and tells how accurate model is out of predicated positive and how many of them are positive. Recall is as similar as precision but there is a slight difference in that instead of total predicated positive it considers total actual positive, so it shows out of true positive how many actual positive consider by our module. $F1$ score is

Table 2 Performance evaluation of classifier

No	Classifier	Precision	Recall	$F1$ score	Accuracy
1	Decision tree	0.877	0.81	0.842	0.84
2	KNN	0.93	0.76	0.83	0.84
3	Random forest	0.91	0.86	0.88	0.88
4	SVC linear kernel	0.89	0.81	0.85	0.85
5	SVC polynomial kernel	0.91	0.80	0.85	0.85
6	SVC radial kernel	0.91	0.82	0.87	0.87
7	Linear regression	0.89	0.81	0.85	0.85

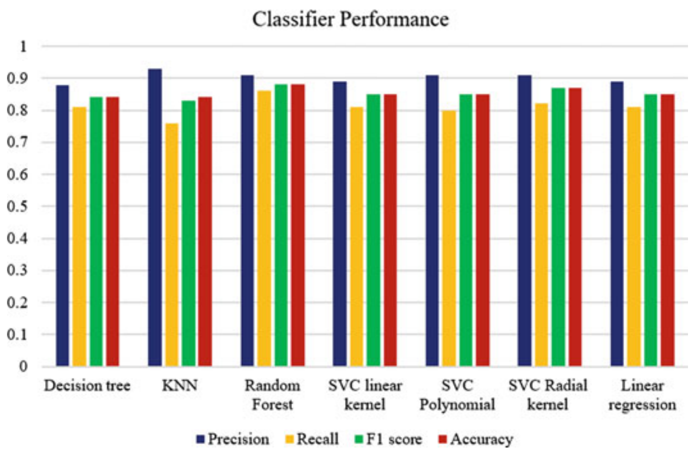


Fig. 8 Comparison of classifier performance

dependent on precision and recall it is a function of both recall and precision. For balance between precision and recall, we need an $f1$ score. By comparing all results of the above-mentioned classifier KNN gives the highest precision which is 0.93 and the second highest is shared between random forest and SVC with the radial kernel which is about 0.91. The highest recall and $f1$ scores are given by random forest as 0.86 and 0.88, respectively. Results of all classifiers with their precision, recall, and $f1$ score is mentioned in the table.

4 Conclusion

In this paper, a novel solution in scene understanding has been proposed to aid visually impaired people in mobility. The paper presents a scene detection system that can be attached to any ordinary white cane and can detect scenes using images captured by the camera module of the system and produces audio cues for blinds. The proposed system can detect two scenes, viz, kids playing and traffic jams. The system can be further extended to detect additional activities in the environment. A stand-alone, lightweight, and low-cost embedded system can be developed as a mobility solution for visually impaired people.

Acknowledgements We express our sincere gratitude to the visually impaired participants in this study, orientation and mobility (O&M) experts and authorities at The Poona Blind Men's Association, Pune. The authors thank the Rajiv Gandhi Science and Technology Commission, Government of Maharashtra, Mumbai, and Vishwakarma Institute of Technology Pune for providing financial support (RGSTC/File-2016/DPP-158/CR-19) to carry out this research work.

References

1. Online web resource: https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment, last accessed on: 01-01-2022
2. Online web resource: https://www.statista.com/statistics/1237876/number-blindness-by-age-gender/, last access on: 01-01-2022
3. Yang W, Huang J, Wang R, Zhang W, Liu H, Xiao J (2021) A survey on tactile displays for visually impaired people. IEEE Trans Haptics 14(4):712–721. http://doi.org/10.1109/TOH.2021.3085915
4. Pissaloux EE, Velázquez R, Maingreaud F (2017) A new framework for cognitive mobility of visually impaired users in using tactile device. IEEE Trans Hum-Mach Syst 47(6):1040–1051. https://doi.org/10.1109/THMS.2017.2736888
5. Blenkhorn, Evans DG (1997) A system for enabling blind people to identify landmarks: the sound buoy. IEEE Trans Rehabil Eng 5(3):276–278. http://doi.org/10.1109/86.623019
6. Andò SB, Marletta V, Valastro A (2015) A haptic solution to assist visually impaired in mobility tasks. IEEE Trans Human-Mach Syst 45(5):641–646
7. Barontini F, Catalano MG, Pallottino L, Leporini B, Bianchi M (2021) Integrating wearable haptics and obstacle avoidance for the visually impaired in indoor navigation: a user-centered approach. IEEE Trans Haptics 14(1):109–122. http://doi.org/10.1109/TOH.2020.2996748

8. Villanueva J, Farcy R (2012) Optical device indicating a safe free path to blind people. IEEE Trans Instrum Meas 61(1):170–177. https://doi.org/10.1109/TIM.2011.2160910
9. Cardillo et al (2018) An electromagnetic sensor prototype to assist visually impaired and blind people in autonomous walking. IEEE Sens J 18(6):2568–2576. http://doi.org/10.1109/JSEN.2018.2795046
10. Jardak et al (2016) Detection and localization of multiple short range targets using FMCW radar signal. In: 2016 global symposium on millimeter waves (GSMM) & ESA workshop on millimetre-wave technology and applications, pp 1–4. http://doi.org/10.1109/GSMM.2016.7500332
11. Bai SL, Liu Z, Wang K, Liu D (2017) Smart guiding glasses for visually impaired people in indoor environment. IEEE Trans Consum Electron 63(3):258–266. http://doi.org/10.1109/TCE.2017.014980
12. Chang W, Chen L, Hsu C, Chen J, Yang T, Lin C (2020) MedGlasses: a wearable smart-glasses-based drug pill recognition system using deep learning for visually impaired chronic patients. IEEE Access 8:17013–17024. https://doi.org/10.1109/ACCESS.2020.2967400
13. Mekhalfi L, Melgani F, Bazi Y, Alajlan N (2015) A compressive sensing approach to describe indoor scenes for blind people. IEEE Trans Circuits Syst Video Technol 25(7):1246–1257. https://doi.org/10.1109/TCSVT.2014.2372371
14. Ramadhan AJ (2018) Wearable smart system for visually impaired people. Sensors 18(3):843
15. Patil QJ, Shu FC (2018) Design and construction of electronic aid for visually impaired people. IEEE Trans Hum Mach Syst 48(2):172–182. http://doi.org/10.1109/THMS.2018.2799588
16. Xiao J, Joseph SL, Zhang X, Li B, Li X, Zhang J (2015) An assistive navigation framework for the visually impaired. IEEE Trans Hum Mach Syst 45(5):635–640. https://doi.org/10.1109/THMS.2014.2382570
17. Bai SL, Liu Z, Wang K, Liu D (2018) Virtual-blind-road following-based wearable navigation device for blind people. IEEE Trans Consum Electron 64(1):136–143. http://doi.org/10.1109/TCE.2018.2812498
18. Hasanuzzaman EM, Yang X, Tian Y (2012) Robust and effective component-based banknote recognition for the blind. IEEE Trans Syst Man Cybern Part C (Appl Rev) 42(6):1021–1030. http://doi.org/10.1109/TSMCC.2011.2178120
19. Ye, SH, Qian X, Wu W (2016) Co-robotic cane: a new robotic navigation aid for the visually impaired. IEEE Syst Man Cybern Mag 2(2):33–42
20. Shoval JB, Koren Y (1998) Auditory guidance with the Navbelt—a computerized travel aid for the blind. IEEE Trans Syst Man Cybern Part C (Appl Rev) 28(3):459–467. http://doi.org/10.1109/5326.704589
21. Aladrén A, López-Nicolás G, Puig L, Guerrero JJ (2016) Navigation assistance for the visually impaired using RGB-D sensor with range expansion. IEEE Syst J 10(3):922–932. http://doi.org/10.1109/JSYST.2014.2320639
22. Khan PP, Rashid M, Hossain M, Ahad MAR (2020) An AI-based visual aid with integrated reading assistant for the completely blind. IEEE Trans Hum-Mach Syst 50(6):507–517. http://doi.org/10.1109/THMS.2020.3027534
23. Ando (2008) A smart multisensor approach to assist blind people in specific urban navigation tasks. IEEE Trans Neural Syst Rehabil Eng 16(6):592–594. http://doi.org/10.1109/TNSRE.2008.2003374
24. Ando B, Graziani S (2009) Multisensor strategies to assist blind people: a clear-path indicator. IEEE Trans Instrum Meas 58(8):2488–2494. http://doi.org/10.1109/TIM.2009.2014616
25. Jiang JY, Lv Z, Song H (2019) Wearable vision assistance system based on binocular sensors for visually impaired users. IEEE Internet Things J 6(2):1375–1383. http://doi.org/10.1109/JIOT.2018.2842229
26. Cheng KW, Bai J, Xu Z (2020) Unifying visual localization and scene recognition for people with visual impairment. IEEE Access 8:6428464296. http://doi.org/10.1109/ACCESS.2020.2984718
27. Islam MM, Sadi MS, Zamli KZ, Ahmed MM (2019) Developing walking assistants for visually impaired people: a review. IEEE Sens J 19(8):2814–2828

28. Chen X, Xu J, Yu Z (2019) A 68-mw 2.2 Tops/w low bit width and multiplierless DCNN object detection processor for visually impaired people. IEEE Trans Circuits Syst Video Technol 29(11):3444–3453. https://doi.org/10.1109/TCSVT.2018.2883087
29. Abu Doush I, Alshatnawi S, Al-Tamimi A, Alhasan B, Hamasha S (2017) ISAB: integrated indoor navigation system for the blind. Interact Comput 29(2):181–202. http://doi.org/10.1093/iwc/iww016
30. Wang Z, Li N, Li B (2012) Fast and independent access to map directions for people who are blind. Interact Comput 24(2):91–106. http://doi.org/10.1016/j.intcom.2012.02.002
31. Mocanu B, Tapu R, Zaharia T (2018) DEEP-SEE FACE: a mobile face recognition system dedicated to visually impaired people. IEEE Access 6:51975–51985. https://doi.org/10.1109/ACCESS.2018.2870334
32. Cappagli G et al (2018) Assessing social competence in visually impaired people and proposing an interventional program in visually impaired children. IEEE Trans Cogn Dev Syst 10(4):929–935. https://doi.org/10.1109/TCDS.2018.2809487
33. Seki Y, Sato T (2011) A training system of orientation and mobility for blind people using acoustic virtual reality. IEEE Trans Neural Syst Rehabil Eng 19(1):95–104. http://doi.org/10.1109/TNSRE.2010.2064791
34. Hu W, Wang K, Yang K, Cheng R, Ye Y, Sun L, Xu Z (2020) A comparative study in real-time scene sonification for visually impaired people. Sensors 20(11):3222
35. Cheng R, Wang K, Bai J, Zhijie Xu (2020) Unifying visual localization and scene recognition for people with visual impairment. IEEE Access 8:64284–64296
36. Tian S, Zheng M, Zou W, Li X, Zhang L (2021) Dynamic crosswalk scene understanding for the visually impaired. IEEE Trans Neural Syst Rehabil Eng 29:1478–1486. https://doi.org/10.1109/TNSRE.2021.3096379

A Real-Time Detection of Indian Traffic Signs for Visually Impaired People

Jyoti Madake, Mahesh Badade, Mrunal Barve, Shripad Bhatlawande, and Swati Shilaskar

Abstract Outdoor navigation on crowded roads is the biggest challenge for visually impaired people. Safe and independent travel for blind people is possible if they are able to recognize the traffic signs. Active and real-time traffic sign recognition is a very crucial requirement for the safety of visually impaired people. The existing blind assistive aids do not support this problem. The aim of this paper is to recognize the Indian traffic signs with the help of SIFT as a feature descriptor. The most frequently used 7 traffic signs were used for identification in the scope of this paper. PCA is used for the optimization of the feature vector. Three classifiers were used for performance analysis and classification which were random forest, K-nearest neighbor, and decision tree. The trained model precisely detects the road traffic signs. The proposed system is an electronic travel aid for blind people with low power requirements and reduced latency. The results state that the recognition using random forest, KNN, and decision tree gives 73%, 71%, and 65% accuracy.

Keywords Computer vision · Indian traffic signs · Traffic sign detection · Random forest · SIFT · Visually impaired people

J. Madake (✉) · M. Badade · M. Barve · S. Bhatlawande · S. Shilaskar
Vishwakarma Institute of Technology, Pune, India
e-mail: jyoti.madake@vit.edu

M. Badade
e-mail: mahesh.badade20@vit.edu

M. Barve
e-mail: mrunal.barve19@vit.edu

S. Bhatlawande
e-mail: shripad.bhatlawande@vit.edu

S. Shilaskar
e-mail: swati.shilaskar@vit.edu

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_19

1 Introduction

The World Health Organization (WHO) has reported 2 billion people across the world to have a visual impairment or complete blindness. The loss of vision is the biggest challenge for carrying out any daily living activities efficiently. The visually impaired people are usually accompanied by a sited guide or a guide dog for the outdoor excursion. They face many challenges while walking on the road. Outdoor travel involves walking around familiar places such as temples, garden, the marketplace, hospitals, and a few more. This travel includes sidewalks walking, using public transport a bus or train. Sometimes, the visually impaired people are accompanied by their sited friends or family members for travel assistance. There have been many assistive solutions developed for blind people outdoor travel. It is challenging for them to understand the traffic signs without any external help [1]. While crossing the road, accurate detection of the traffic signs is most crucial. This paper details a low-power and affordable solution for the real-time detection of traffic signs.

Many studies have been carried out in the area of detecting and classifying real-time traffic signs. There is a possible way to detect and identify traffic signs in real time in three stages: detection, feature extraction, and then classification [2]. Saadna et. al. proposed different method for detecting and recognizing traffic signs using neural vectors and support vector machines. SVM is used for the classification of the signs [3]. Sugiharto and Harjoko [4] proposed a color-based detection method using HOG (Histogram Oriented Gradient) followed by SVM for classification. Many authors also propose detecting the signs using a combination of various classifiers and random forest methods [5]. The authors proposed the feature extraction by using a local binary pattern which is the input to the extreme learning machine network which is used for the classification as well as recognition [6]. The authors also proposed a method that is based on the extreme learning machine [ELM]. This method includes feature extraction with the help of HOG, and then, it is followed by using a single classifier [7]. There are several models used for the shape features, and one of them is FOSTS [8]. The authors also developed various new datasets which consist of more than 10,000 images which are then classified, on the basis of robust end-to-end CNN [9]. The authors also suggested the multiscale convolution network which is majorly used for local traffic sign detection [10]. The authors proposed another method that works in real-time detection and recognition of the traffic signs. This proposed system was carried out using deep convolutional neural networks and making the system by varying the different parameters in a given proposed system [11]. Another approach similar to this was proposed for detecting the speed sign by using shape-based features. Distance to border vectors is used to detect the borders and the artificial neural network for the classification [12]. Some authors also proposed a hybrid approach for traffic sign detection which included the color detection-based system. The color of the given sign was detected and classified by using a pixel classifier, and then, the results are generated according to the hypothesis [13]. Some of the authors used a combination of algorithms to detect the signs. One of the proposed methods used feature combination and random forests in which the image enhancement was achieved using HSV and they were classified by using a random forests classifier [14]. The authors proposed an algorithm on the basis of multi-feature and multi-classifier features in which traffic signs are divided into subsets, and SVM and HOG were used in the method [6]. Another proposed system introduced a new traffic sign dataset of 105 thousand street-level images which covered 400 manually annotated traffic signs classes [15].

The authors proposed a new approach for the detection and recognition of traffic signs by using hybrid features formed by 2 robust feature descriptors HOG and SURF (speeded up robust features) [16]. Another approach was proposed which had an adaptive approach that mainly emphasized adaptability to various roads and traffic sign environments [17]. A real-time approach using SVM and color segmentation was also proposed in which a small computer-mounted car was developed which was used to diagnose the highway maintenance and sign inventory [18]. Some authors proposed algorithms based on single classifiers like random forests, SVM by using German, and Swedish datasets [19, 20]. The authors proposed another approach based on multi-block LBP features with uses SVM normalization. This system had 2 normalization techniques that are min–max and *z* score [21]. Some researchers used more than 2 classifiers like decision trees, k-d trees, and random forests [22]. The authors used another method to improve the accuracy and precision of detection and classification which was random forest and fisher's criteria. They were used to lower the feature space and also accelerate the classification [23]. Tree classifiers were widely used for designing the detection system which also uses HOG, SVM, and distance transforms [24]. The authors proposed another method using a GTSRB dataset (German traffic sign recognition benchmark) which included more than 40 classes and 50,000 images [25]. A new proposed approach was detecting traffic signs using spatially weighted HoG trees. This system implemented image segmentation which used red color enhancement to reduce search space [26]. Another approach suggested traffic sign recognition using grid search and HOG-SVM. Grid search techniques were used to optimize the various parameters of SVM [27]. Many of the approaches included a real-time system on the basis of HOG and convolutional neural networks [28]. As per recent research, most of the detection systems are real-time and also focus on being quick and fast [29]. The authors suggested an approach that resulted in being 20 times faster than the existing methods which were based on the color probability model and color detection [30]. Some authors proposed a color-fused multiple features approach for traffic sign detection in which a gentle AdaBoost classifier was used to separate traffic signs from the background [31]. Some approaches suggested SVM-based approach to recognize the traffic sign which gave better performance [32]. Some authors suggested an approach using a novel permutation-based local image feature which gives an advantage of fast and illumination invariant. After the survey of all the proposed techniques for traffic sign detection systems using CNN, researchers found that the challenges and the limitations faced by the convolutional neural networks and all the systems working with CNN which were calculated in research in the terms of time complexity and accuracy [33]. Some papers proposed CNN and fast RCNN-based approaches for the detection of traffic signs in which GTSRB (German Traffic Sign Recognition Benchmark) datasets were used [34].

2 Methodology

This paper proposes a traffic sign recognition system. A block diagram of the given proposed system is shown in Fig. 1. It consists of a monocular camera, a processor-based system, and an earphone. The camera captures real-time details of the surrounding environment. The processor-based system recognizes traffic signs in the detected scene. It converts traffic signs into audio feedback and conveys them to visually impaired people by earphones.

A machine learning approach has been implemented to recognize the traffic signs, and the same approach is shown in Fig. 2.

2.1 Dataset Collection and Pre-processing

The system is implemented using Indian traffic sign dataset. It consists of 14,832 images of 7 classes namely, (i) speed limit 30, (ii) speed limit 50, (iii) speed limit 80, (iv) hump road, (v) no horn, (vi) turn left, and (vii) turn right. The complete dataset

Fig. 1 Block diagram of the proposed system

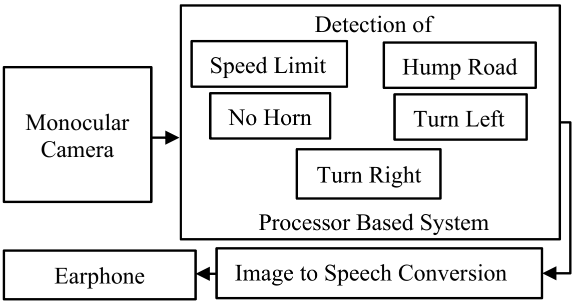


Fig. 2 Machine learning approach for proposed system

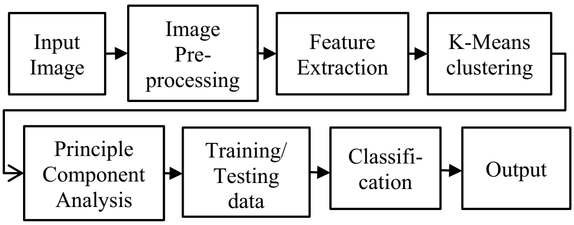


Fig. 3 Sample images of the traffic signs

was obtained from the Internet [35–39]. The sample images of the dataset are shown in Fig. 3.

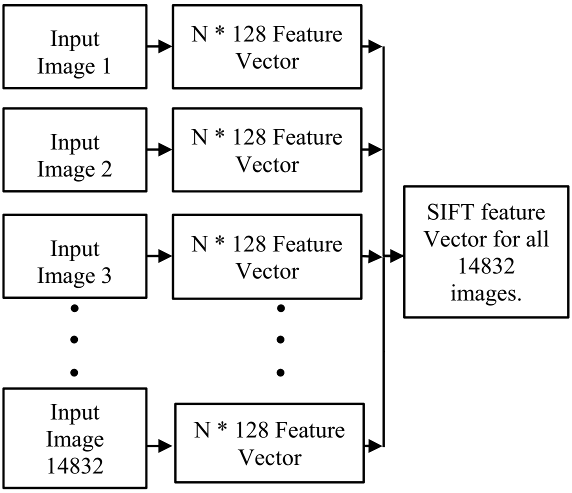
Image processing techniques such as cropping, rotating, and flipping were performed on the images. All the images of the dataset were resized into 200 × 200 pixels and then converted to grayscale.

2.2 Feature Vector Compilation

Scale Invariant Feature Transform (SIFT) was used for feature extraction. SIFT is a machine learning algorithm used for detecting, describing, and matching local features in images. SIFT algorithm generates a feature vector in dimensions of (Y, 128), where Y is the n number of key points generated for the particular image. The process of feature extraction using SIFT is shown in Fig. 4.

The SIFT algorithm was used to extract the features from the 14,832 grayscale images. The (530,786, 128) features were obtained for 14,832 grayscale images.

Fig. 4 Feature extraction using SIFT

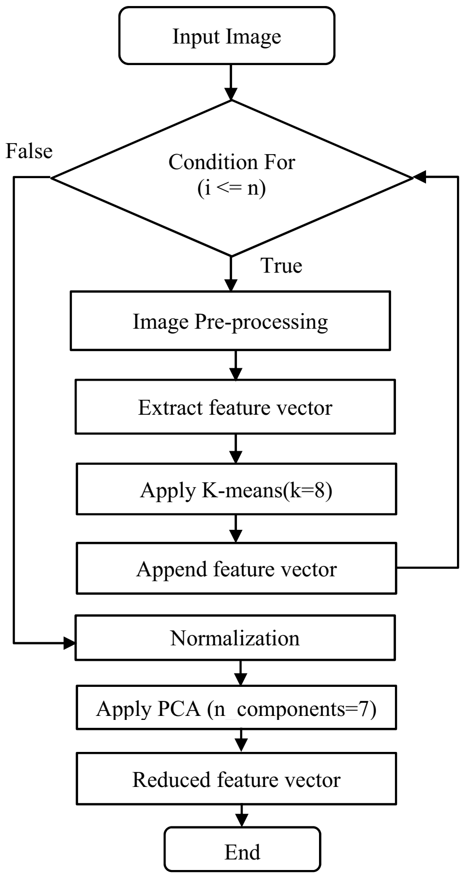


2.3 Dimensionality Reduction

The dimension of the given extracted feature vector was too large. K-means clustering and Principal Component Analysis (PCA) were used to lower the size of an extracted feature vector. K-means clustering is an unsupervised machine learning algorithm that groups the unlabeled dataset into different clusters. Principle component analysis (PCA) is an unsupervised, non-parametric statistical technique that is mainly used for dimensionality reduction in machine learning. The flow diagram for dimension reduction is shown in Fig. 5.

The extracted feature vectors were applied to the K-means clustering. The $K = 8$ was chosen based on the elbow method. All the feature vectors were grouped into 8 clusters. The dimensions of the feature vector were converted into (14,832, 8). The individual predicted SIFT vectors were normalized to reduce the feature bias. Min–Max scaling has been used to normalize the data. Equation (1) represents Min–Max Scaling.

Fig. 5 Flow diagram for dimensionality reduction



$$A' = \frac{A - A_{\min}}{A_{\max} - A_{\min}} \tag{1}$$

where $A_{\max}$ is the maximum value and $A_{\min}$ is the minimum value of the feature.

Further, the principal component analysis (PCA) was used to reduce the dimensions. The size of the feature vector was converted from (530,786, 128) to (14,832, 7) after complete dimensionality reduction. Algorithm 1 describes the step-wise process for dimensionality reduction of the feature vector.

Algorithm 1 Feature Vector Optimization

Input: SIFT feature vector (530786, 128)
Output: Reduced feature vector (14832, 7)

Initialization:
Loop process

1: **for** each image in dataset **do**
2: Extract SIFT features (530786, 128)
3: Reduce size using K-means($K = 8$)
4: Append feature vectors
5: Normalize using Min-Max Scaler
6: **end for**
7: Fit PCA (n_components=7)
8: Transform using PCA
9: **return** reduced feature vector (14832, 7)

2.4 Classification and Recognition of Traffic Sign

The dataset of 14,832 traffic sign images was divided into an 80:20 ratio, where 80% of images were used for training and 20% images were used for testing. Three different classifiers were used to assess the performance of the system, namely (i) random forest, (ii) decision tree, and (iii) KNN (K-nearest neighbor).

The first classifier was random forest. It is widely used as it gives much more accurate and robust noise results than a single classifier. There are numerous simple trees in it. The data from each tree determines the class assignment based on a single vote. Also using this method gives the advantage that they are more compatible and user-friendly. Equation (2) represents mean squared error (MSE).

$$\mathrm{MSE} = \frac{1}{X}\sum_{i=1}^{X}(p_i - q_i)^2 \tag{2}$$

where X is the number of data points, p_i is the value returned by the model, and q_i is the actual value for data point i.

The second classifier was decision tree. It is mostly used for classification and regression. It provides a flowchart-like structure in which internal nodes are tested on attributes, each branch denotes results of the test, and leaf node indicates class label. It is capable of handling both numerical and categorical data as well as problems involving multiple outputs. This is a very specific type of classifier to make a decision.

The third classifier was KNN. It is a supervised learning technique that stores all the data and also classifies them into various categories. This method can also be used for regression problems, but it is mostly used for classification problems. This is a non-parametric algorithm, meaning that no assumptions are made when it analyses data. The distance is calculated by various methods like Euclidean distance. Equation (3) represents Euclidean distance.

$$\mathrm{dist}((p, q)(r, s)) = \sqrt{(p - r)^2 + (q - s)^2} \tag{3}$$

where d is distance and (p, q) and (r, s) is 2 points in the dimensional plane.

The process for traffic sign recognition is described in Algorithm 2.

Algorithm 2 Traffic Sign Recognition

Input: Traffic sign image
Output: Predicted class

Initialization:

Fit model with training data for classification
Predict output using a pre-trained model
for every predicted output **do**:
  if(output==0)
    return “The traffic sign is of Compulsory Ahead or Turn Right”
  elseif(output==1)
    return “The traffic sign is of Hump Road”
  elseif(output==2)
    return “The traffic sign is of No Horn”
  elseif(output==3)
    return “The traffic sign is of Speed Limit 30”
  elseif(output==4)
    return “The traffic sign is of Speed Limit 50”
  elseif(output==5)
    return “The traffic sign is of Speed Limit 80”
  elseif(output==6)
    return “The traffic sign is of Compulsory Turn Left Ahead”

Table 1 Comparison of precision, recall, and $F1$-score

Classifier	Precision (%)	Recall (%)	$F1$-score (%)
Random forest	72	73	72
Decision tree	65	66	65
K-nearest neighbor	70	71	70

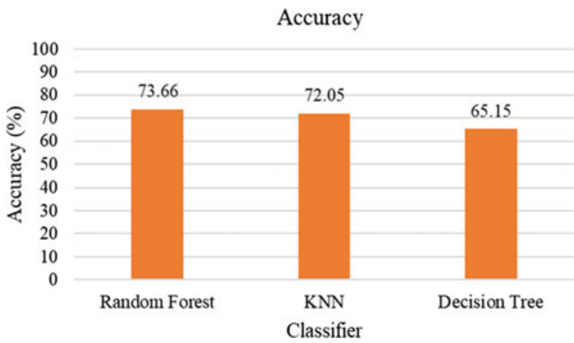


Fig. 6 Accuracy comparison of different classifiers

3 Result

The proposed system recognizes the traffic signs using three classifiers. The model was trained using random forest, decision tree, and KNN. We calculated the precision, recall, and $F1$-Score for all three classifiers. The precision value obtained for the random forest classifier is 0.72 which is higher than the precision score of decision tree and K-nearest neighbor. The random forest outperformed the other two classifier for the other two metrics of Recall and $F1$-Score. The performance parameters of different classifiers are shown in Table 1.

The random forest classifier gave an accuracy of 73.66%. The decision tree gave an accuracy of 65.15%, and the KNN gave 72.05% accuracy.

The bar graph of accuracy achieved by three classifiers is shown in Fig. 6. On the basis of the above graph, we can say that the random forest gives the highest accuracy as compared to others.

4 Conclusion

The proposed system has a SIFT descriptor, and 3 algorithms are used to recognize traffic signs. This system helps self-driven cars to detect and identify the traffic signs so that they can process accordingly. Safe movement of the car is ensured as the system performance is high. When the traffic signs on the road are not clear or not clearly readable, the accuracy of the system may reduce. Also, in a rainy or foggy atmosphere, it becomes difficult for the system to recognize the signs accurately. This system can widely be used in all-electric and autonomous cars. Also, the system can be modified accordingly so that we can use the system with adaptive cruise control in manual cars as well. The system can be improved to give more precise results and improve the accuracy by increasing the dataset. Adding to that, we can also work on making the system faster and improve the detection of signs in all climatic conditions.

References

1. Yang Y, Luo H, Xu H, Wu F (2015) Towards real-time traffic sign detection and classification. IEEE Trans Intell Transp Syst 17(7):2022–2031
2. Saadna Y, Behloul A (2017) An overview of traffic sign detection and classification methods. Int J Multimedia Inf Retrieval 6(3):193–210
3. Ardianto S, Chen C-J, Hang H-M (2017) Real-time traffic sign recognition using color segmentation and SVM. In: 2017 international conference on systems, signals and image processing (IWSSIP). IEEE, pp 1–5
4. Sugiharto A, Harjoko A (2016) Traffic sign detection based on HOG and PHOG using binary SVM and k-NN. In: 2016 3rd international conference on information technology, computer, and electrical engineering (ICITACEE). IEEE, pp 317–321
5. Ellahyani A, El Ansari M, El Jaafari I (2016) Traffic sign detection and recognition based on random forests. Appl Soft Comput 46:805–815
6. El Margae S, Sanae B, Mounir AK, Youssef F (2014) Traffic sign recognition based on multi-block LBP features using SVM with normalization. In: 2014 9th international conference on intelligent systems: theories and applications (SITA-14). IEEE, pp 1–7
7. Yao C, Wu F, Chen H, Hao X, Shen Y (2014) Traffic sign recognition using HOG-SVM and grid search. In: 2014 12th international conference on signal processing (ICSP). IEEE, pp 962–965
8. Abukhait J, Zyout I, Mansour AM (2013) Speed sign recognition using shape-based features. Int J Comput Appl 84(15)
9. Alghmgham DA, Latif G, Alghazo J, Alzubaidi L (2019) Autonomous traffic sign (ATSR) detection and recognition using deep CNN. Procedia Comput Sci 163:266–274
10. Pei S, Tang F, Ji Y, Fan J, Ning Z (2018) Localized traffic sign detection with multi-scale deconvolution networks. In: 2018 IEEE 42nd annual computer software and applications conference (COMPSAC), vol 1. IEEE, pp 355–360
11. Chen J, Jia K, Chen W, Lv Z, Zhang R (2021) A real-time and high-precision method for small traffic-signs recognition. Neural Comput Appl 1–13
12. Abedin MZ, Dhar P, Deb K (2016) Traffic sign recognition using hybrid features descriptor and artificial neural network classifier. In: 2016 19th international conference on computer and information technology (ICCIT). IEEE, pp 457–462
13. Eom T-J, Goswami K, Kim B-G, Lee J (2012) Hybrid color space based road sign detection technique. In: 2012 IEEE second international conference on consumer electronics-Berlin (ICCE-Berlin). IEEE, pp 81–83
14. Greenhalgh J, Mirmehdi M (2012) Traffic sign recognition using MSER and random forests. In: 2012 proceedings of the 20th European signal processing conference (EUSIPCO). IEEE, pp 1935–1939
15. Ertler C, Mislej J, Ollmann T, Porzi L, Kuang Y (2019) Traffic sign detection and classification around the world. arXiv preprint arXiv:1909.04422
16. Islam KT, Raj RG, Mujtaba G (2017) Recognition of traffic sign based on bag-of-words and artificial neural network. Symmetry 9(8):138
17. Larsson F (2011) Shape based recognition–cognitive vision systems in traffic safety applications. Ph.D. diss., Linköping University Electronic Press

18. Le TT, Tran ST, Mita S, Nguyen TD (2010) Real time traffic sign detection using color and shape-based features. In: Asian conference on intelligent information and database systems. Springer, Berlin, pp 268–278
19. Gim J, Hwang M, Ko BC, Nam J-Y (2015) Real-time speed-limit sign detection and recognition using spatial pyramid feature and boosted random forest. In: International conference image analysis and recognition. Springer, Cham, pp 437–445
20. Zaklouta F, Stanciulescu B (2012) Real-time traffic-sign recognition using tree classifiers. IEEE Trans Intell Transp Syst 13(4):1507–1514
21. Zhou Z (2018) Research on traffic sign recognition algorithm based on SVM of LBP. In: 2018 international conference on transportation & logistics, information & communication, smart city (TLICSC 2018). Atlantis Press, pp 10–16
22. Zaklouta F, Stanciulescu B, Hamdoun O (2011) Traffic sign classification using kd trees and random forests. In: The 2011 international joint conference on neural networks. IEEE, pp 2151–2155
23. Zaklouta F, Stanciulescu B (2014) Real-time traffic sign recognition in three stages. Robot Auton Syst 62(1):16–24
24. Meuter M, Nunn C, Gormer SM, Muller-Schneiders S, Kummert A (2011) A decision fusion and reasoning module for a traffic sign recognition system. IEEE Trans Intell Transp Syst 12(4):1126–1134
25. Stallkamp J, Schlipsing M, Salmen J, Igel C (2011) The German traffic sign recognition benchmark: a multi-class classification competition. In: The 2011 international joint conference on neural networks. IEEE, pp 1453–1460
26. Zaklouta F, Stanciulescu B (2011) Segmentation masks for real-time traffic sign recognition using weighted HOG-based trees. In: 2011 14th international IEEE conference on intelligent transportation systems (ITSC). IEEE, pp 1954–1959
27. Sharif S, Lilja J (2020) Traffic sign recognition using machine learning
28. Temel D, Kwon G, Prabhushankar M, AlRegib G (2017) CURE-TSR: challenging unreal and real environments for traffic sign recognition. arXiv preprint arXiv:1712.02463
29. Ruta A, Li Y, Liu X (2010) Real-time traffic sign recognition from video by class-specific discriminative features. Pattern Recogn 43(1):416–430
30. Shao F, Wang X, Meng F, Rui T, Wang D, Tang J (2018) Real-time traffic sign detection and recognition method based on simplified Gabor wavelets and CNNs. Sensors 18(10):3192
31. Yang X, Qu Y, Fang S (2012) Color fused multiple features for traffic sign recognition. In: Proceedings of the 4th international conference on internet multimedia computing and service, pp 84–87
32. Wali SB, Hannan MA, Hussain A, Samad SA (2015) An automatic traffic sign detection and recognition system based on colour segmentation, shape matching, and svm. Math Probl Eng 2015
33. Tian T, Sethi I, Patel N (2014) Traffic sign recognition using a novel permutation-based local image feature. In: 2014 international joint conference on neural networks (IJCNN). IEEE, pp 947–954
34. Boujemaa KS, Berrada I, Bouhoute A, Boubouh K (2017) Traffic sign recognition using convolutional neural networks. In: 2017 international conference on wireless networks and mobile communications (WINCOM). IEEE, pp 1–6
35. Online web resource: https://data.mendeley.com/datasets/4tznkn45mx/1, last accessed on 1-12-21
36. Online web resource: http://www.nlpr.ia.ac.cn/pal/trafficdata/recognition.html, last accessed on 1-12-21
37. Online web resource: https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign, last accessed on 3-12-21
38. Online web resource: https://btsd.ethz.ch/shareddata/, last accessed on 3-12-21
39. Online web resource: https://www.vicos.si/resources/dfg/, last accessed on 10-12-21

Electronic Travel Aid for Crosswalk Detection for Visually Challenged People

Shripad Bhatlawande, Neel Gokhale, Dewang V. Mehta, Parag Gaikwad, Swati Shilaskar, and Jyoti Madake

Abstract Vision is one of the most important sensory systems. It helps us to sense and gauge the surroundings around us. It also helps us to see colors as well as to assess any dangers in our path. Therefore, the proposed system is aimed to aid people who are visually impaired or blind. These people face the most difficulties when they are commuting alone. Independent movement in an unknown environment is strenuous for these people. They need to be always accompanied by someone, especially when crossing roads. The proposed system can be used when crossing the roads to detect a zebra crossing. Although there are a few solutions available today, those are not easy to use. Achieving ease of use with a cheaper cost is the author's aim in building this system. Computer vision and machine learning models have been used to find a solution to this problem. The authors have prepared a dataset of 5510 images. The system uses SIFT descriptors for feature extraction. K-means and PCA for dimensionality reduction are used. Five ML classifiers are trained, and the performance is assessed. The highest accuracy was achieved by the Random Forest Classifier, which was 89%.

Keywords Computer vision · Visually challenged · Crosswalk detection · Python · Machine learning · SIFT descriptors · PCA · Random forest classifier

---

S. Bhatlawande (✉) · N. Gokhale · D. V. Mehta · P. Gaikwad · S. Shilaskar · J. Madake
Electronics and Telecommunication Engineering Department, Vishwakarma Institute of Technology, Pune 411037, India
e-mail: shripad.bhatlawande@vit.edu

N. Gokhale
e-mail: neel.gokhale19@vit.edu

D. V. Mehta
e-mail: dewang.mehta20@vit.edu

P. Gaikwad
e-mail: parag.gaikwad20@vit.edu

S. Shilaskar
e-mail: swati.shilaskar@vit.edu

J. Madake
e-mail: jyoti.madake@vit.edu

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_20

1 Introduction

Visually impaired are the people who have no vision or have encountered a vision loss either from birth or at a later age. In the world, around 285 million people are visually impaired [1]. This equates to around 1 person being affected in every 27.71. Visual impairment mostly exists in people who are 50 and above. They face problems like walking outdoors, studying, and communicating with other people. This topic was chosen to provide aid for people suffering from visual impairment.

In modern times, Science and Technology have proved to be a boon for people who face physical disabilities. With the progress of new techniques in artificial intelligence and machine learning, many day-to-day problems faced by them are being solved. On the one hand, technologies like advanced prosthetics have provided aid to people who have disabilities, while on the other hand technologies like genetic modification have even eliminated some diseases. Advancements in mobile cameras with reduced size and increased clarity are being made. The processing capabilities of mobile phones and microprocessors are improving at an exponential rate. The reach of these technologies is increasing, and they are being available at cheaper prices. Hence, common people can use high-end technologies like AI-ML for their aid. Object detection is a versatile field and can be easily deployed on most generic smartphones these days. Machine learning and artificial intelligence have a lot of potential to improve the lives of people with sensory disabilities, especially visual impairment. In the initial stages of AI and ML algorithms, they were deployed on specially designed and expensive hardware. In [2], they have proposed an electronic stick with a buzzer that beeps whenever the person approaches any obstacle. In [3], they have devised a pair of smart glasses, wherein depth sensor and ultrasonic sensor are both utilized to solve the object detection problem. The above-mentioned devices needed exclusive hardware as well as produced beeps to indicate an obstacle. Mobile phones have eliminated the need for such specific hardware. App-based software's has also been developed which can detect objects and identify places. They also give all this data in audio format [4–6]. All these technologies help in day-to-day commuting for the visually impaired but fall short to cover unusual scenarios, such as walking on the street or reading out directional boards. Refs. [7] and [8] try to cover a lot of these mobile-assisted technologies developed for the visually impaired as well as the blind to traverse urban environments. They have also covered some technologies which were specifically developed for urban environments. Zebra crossing detection is one of the important needs of visually impaired people which is being currently tackled by many machine learning algorithms. In [9] and [10], the developers have proposed a real-time system for crosswalk detection using the monocular depth camera. The horizontal projection was used to separate the possible crosswalk area from the road lanes. The result of this experiment shows that the method proposed in this paper has an effective methodology to detect the crosswalk.

In [11], a block-based Hough transform was deployed to recognize the crosswalk in scenes, images, or videos. Edge detection was used as pre-processing. To minimize the effect of various shadows, adaptive thresholding was used. Parallel line detection was used to recognize the crosswalk along with Hough transforms. In [12], unique geometrical features are used for zebra crossing detection. Sorted order of zebra-crossing stripe's edges was used as the unique geometrical feature. Gabor filter was used to disarm the effect of the noise and shadow. Sobel edge detectors were used to detect the horizontal lines. Parallel horizontal edges were used to vanish points to verify zebra crossing. In [13], using a single camera-captured image a fast and stable algorithm was developed for detecting the crosswalk. This was achieved using bipolarity-based segmentation and projective invariant-based recognition. The feature points were extracted using the fisher criterion, and lastly, projective invariants were used to recognize the crossing. In [14], a methodology was developed to detect the zebra crossing and recognize it. It has employed image processing techniques such as the flood fill operation, Hough transform, adaptive histogram equalization, and support vector machine classifiers. Firstly, the image is converted to the binary or grey-scale, and then, the flood fill operations and morphological operations are applied. SVM was used to detect ROI for recognizing the zebra crossing. In [15], a convolutional neural network (CNN) was used to detect zebra crossing. Whereas in [16], an approach for crosswalk detection from aerial imagery was proposed. Detection on different datasets is done using the HOG as well as the LBPH features, combined with SVM. This system is used to detect crosswalks in aerial imagery using the data-driven approach. In [17, 18], a solution to detect a particular traffic rule violation, i.e., stopping on zebra crossing, is developed. Firstly, the pre-processing used includes the binarization of images, filtering, and morphological processing. In filtering, the median and Laplacian filters are used. In run time, the image captured is sent to the detector to detect the output. This approach is tested on 50 test data samples spread on Indian roads. In [19], detection and localization of crosswalks are proposed using the Maximally Stable Extremal Regions (MSER) and extended Random Sample Consensus (RANSAC). MSER is used for feature extraction.

2 Methodology

This paper presents a mobility aid for the detection of crosswalks. The system block diagram is shown in Fig. 1. It consists of a camera, a processor-based system, and an earphone. The camera acquires the information of the road and provides it as input to the processor-based system. An algorithm has been implemented to detect and interpret crosswalks. The system translates detected crosswalk scenes into audio feedback. The audio feedback is conveyed to the visually impaired user via earphone.

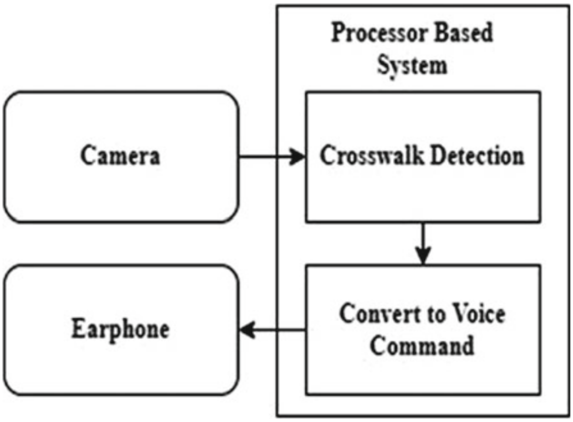


Fig. 1 Crosswalk detection system

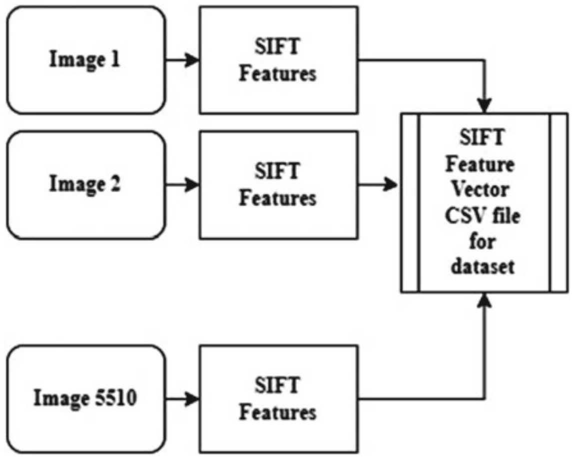
The dataset includes a total of 5510 images. Of the 5510 images, 60% images were obtained from the Internet [20] and 40% were collected by the authors. There were 3010 positive images and 2500 negative images. The sample images in the dataset are shown in Fig. 2.

All images from the dataset were resized to (200, 200) pixels. These resized images were converted to grayscale.

Scale Invariant Feature Transform (SIFT) was applied to 5510 images to extract features. The total size of the resulting feature vector was [1,455,359 × 128]. The process of feature vector compilation is shown in Fig. 3 and described in Algorithm 1.

Fig. 2 Sample images from the dataset

Fig. 3 Extraction of SIFT features



Algorithm 1 Compilation of Feature Vector

Input file: Images from the dataset
Output file: Feature Descriptor of size (1,455,359, 128)

```
Initializations of variables: i = 1, n = number of images
for (i to n):
        Read ith image
        Resize the image to (200, 200)
        Convert images to grayscale
        Normalize and detect SIFT descriptors
        Append normalized descriptors to (.csv) feature vector file
        i++
end for
end
```

The dimension of the feature vector was reduced using dimension reduction techniques. This large-size feature vector was optimized using K-means clustering. The feature vector of size (1,455,359, 128) was divided into 5 clusters. The cluster value 5 was chosen based on the Elbow method. Every individual image was predicted with a pre-trained K-means model. This formed a 5-bin histogram for each image. The Histogram bin values were normalized. Standardization was also performed on

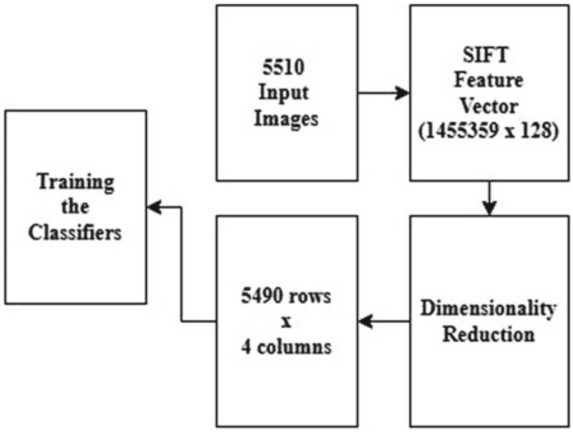


Fig. 4 Dimensionality reduction process

this data. This resulted in a reduced feature vector size of (5491, 6). Then, Principal Component Analysis (PCA) was applied to further reduce the dimensions. The size of the final feature vector was reduced to (5491, 4). The dimensionality reduction process is shown in Fig. 4.

The reduced feature vector file was provided to five classifiers (i) decision tree classifier, (ii) Random Forest classifier, (iii) Support vector classifier (SVC), (iv) K-Nearest Neighbor classifier, and (v) Logistic regression. The first classifier trained was the decision tree classifier. It uses a criterion called "entropy." This criterion helps in determining the splitting of data points. The decision tree classifier splits the data of all points which belong to different classes till they get separated. Entropy function and difference of entropies for classification are shown in Eqs. (1) and (2), respectively.

$$E(s) = \sum_{i=1}^{c} p_i \log_2(p_i) \tag{1}$$

$$H(s) = -\text{probability of}\,(a+) - (\text{probability of}\,(a-)) \tag{2}$$

In Eq. (1), p_i is the probability of class i. In Eq. (2), $(a+)$ = Percentage of positive class and $(a-)$ = Percentage of negative class in the decision tree. The second classifier trained was Random Forest. The Random Forest classifier uses multiple decision trees. The classifier used 100 decision trees.

$$\text{MSE} = \frac{1}{L}\sum_{k=1}^{L}(f_k - y_k)^2 \tag{3}$$

In Eq. (3), L is the number of data points, f_k is the value returned by the model and y_k is the actual value for data points k. The third classifier used was K-Nearest

Neighbors (KNN). It places new data points into a class that is similar to existing data points by comparing them. It measures the similarity of the test data and places it in the most relevant group to solve classification problems. The distances calculated in KNN are of three types: Manhattan distance, Euclidean Distance, and Minkowski distance. Minkowski distance is given in Eq. (4).

$$\text{Dist.}(x, y) = \sum_{i=1}^{n} (|x_i - y_i|^p)^{1/p} \tag{4}$$

The value of p decides the distance nature. The distance is Euclidian for $p = 2$, and Manhattan for $p = 1$.

The third classifier used was the Support Vector Classifier (SVC). It achieves better accuracy with a relatively minimal quantity of data. SVC creates a hyperplane that segregates feature into classes. The general equation of SVM with the linear kernel is shown in Eq. (5)

$$y = w^{\mathrm{T}} X + b \tag{5}$$

where w represents the vectors in the linear hyperplane, b is the biased term, and X represents variables in that term. SVC with three types of kernels, namely linear, polynomial, and RBF (radial bias function), have been trained. The last classifier trained was Logistic regression. It uses a sigmoid function to map predicted values to probabilities. The threshold value defined for this sigmoid function is 0.5. The complete process for classification and detection of the crosswalk is shown in Fig. 5.

The process of detection of crosswalks inside the processor is given in Algorithm 2.

Algorithm 2 Detection of crosswalks inside the processor

Input: Image from camera
Output: Voice command to the user

1: Extract SIFT descriptors from the Image
2: Append SIFT descriptors to Feature Vector File
3: Use K-means on the feature vector to reduce columns to 5 from 128
4: PCA with 3 components to reduce the size from 5 columns to 3 columns
5: Load the Saved Model (Classifier)
6: Predict if the image contains a crosswalk or not.
7: **If** (Prediction==0) Play Voice command “There is a crosswalk”
8: **If** (Prediction==1) Play Voice command “There is no crosswalk”
9: **END**

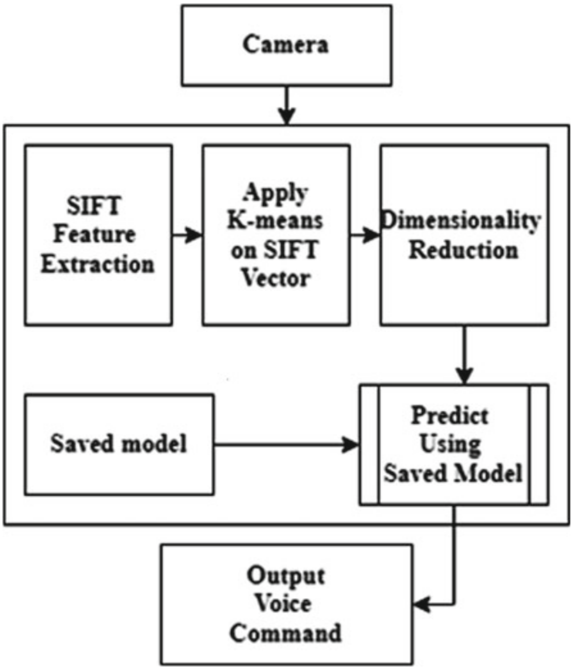


Fig. 5 Classification and detection of crosswalks

3 Performance Evaluation and Results

The model was trained on 4411 (80%) images and tested on 1099 (20%) images. An array of five classifiers was used for the classification and detection of crosswalks. Performance measures such as recall, *F*-measure, and precision were compared for all the classifiers used in the system. The accuracy analysis is presented in Table 1 and Fig. 6.

The confusion matrix for the classifier with the highest accuracy is shown in Table 2.

Table 1 Analysis of classifier performance

Classifiers	Accuracy (%)	Precision (%)	Recall (%)	*F*1 (%)
Random forest	89.62	88.68	88.32	88.50
Decision tree	85.53	84.34	83.50	83.92
KNN	88.08	89.10	83.90	86.42
Logistic regression	86.62	87.55	82.09	84.73
SVC (Linear kernel)	86.80	87.44	82.69	85
SVC (Polynomial)	86.26	82.39	88.53	85.35
SVC (RBF)	87.53	88.29	83.50	85.83

Fig. 6 Accuracy analysis

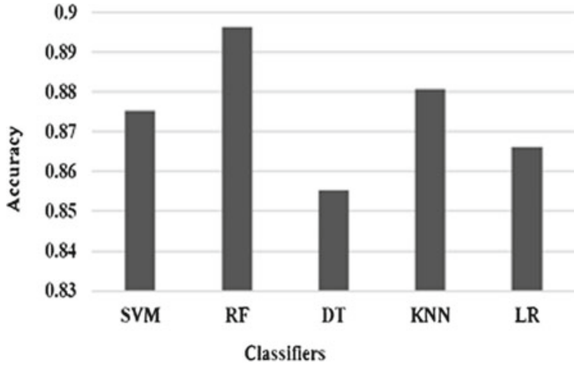


Table 2 Confusion matrix for the random forest classifier

Total (1099)	Predicted positive	Predicted negative
Actual positive	546	56
Actual negative	58	439

Among all the classifiers, the Random Forest classifier provided the highest accuracy (82.62%), whereas the KNN classifier provided 88.08%. “Recall is a metric that measures the number of true positives predicted from the total data,” i.e., true positives and false negatives. Precision is also similar to recall. The difference is that it measures the true positives from the true positives and false negatives. The $F1$ score or measure is given by Eq. (6).

$$F1 \text{ Score} = 2 * (\text{Precision} + \text{Recall}) / (\text{Precision} * \text{Recall}) \tag{6}$$

The highest precision was achieved with the KNN algorithm (89.10%) followed by Random Forest (88.68%). SVC polynomial kernel has the highest recall at (88.53%), whereas the highest $F1$ measure was attained by the Random Forest.

Random Forest algorithm constructs multiple decision trees during training time. This allows the algorithm a high sense of pattern recognition in a very complex dataset. It was trained with one hundred such decision trees. SVC with linear kernel gave lower accuracy than radial bias function because of the non-linearity of the training data. KNN had a better accuracy than logistic regression as the input data was highly non-linear. Random Forest was chosen as it had the highest overall metric. The Random Forest model took 6.16 MB (64, 63, 488 bytes) of disk space. The complete system can be ported on a smartphone. The computational time for the Random Forest model to make predictions on a single frame was 0.2399 s on a machine with AMD RYZEN 7 4800H Processor. The accuracy can be increased by a margin by increasing the dataset. Increasing the number of images may allow the classifier to detect irregular crosswalks. Crosswalks exist on busy roads. Hence, a lot of crosswalks contain rough patches and smudges. Increasing the dataset might help the classifier to detect crosswalk with these smudges.

4 Conclusion

This paper presented a system that can accurately detect crosswalks for assisting visually impaired people. This system utilized computer vision techniques and consistently detected crosswalks. The system implementation compared five different classifiers wherein the Random Forest classifier achieved the highest classification and detection accuracy (89%). A novel blend of K-Means and PCA has been used in the implementation for compilation of an optimized feature vector. This approach ensured a fast convergence of the algorithm and enabled its porting on a portable embedded system (Raspberry-pi board). The system needs uniform illumination for accurate detections. The shadows, dirt, and non-uniform illumination are some of concerns. The accuracy of this model can be improved by training it on a larger dataset. This system can be implemented in the form of a blind stick, cap, spectacles, and smart clothing. The complete system can be comfortable ported on smartphone for assistance of visually impaired people.

Acknowledgements We express our sincere gratitude to the visually impaired participants in this study, orientation, and mobility (O&M) experts and authorities at The Poona Blind Men's Association, Pune. The authors thank the Rajiv Gandhi Science and Technology Commission, Government of Maharashtra, Mumbai, and Vishwakarma Institute of Technology Pune for providing financial support (RGSTC/File-2016/DPP-158/CR-19) to carry out this research work.

References

1. Bourne RRA, Flaxman SR, Braithwaite T, Cicinelli MV, Das A, Jonas JB, Keeffe J et al (2017) Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis. Lancet Glob Health 5(9):e888–e897
2. Kumar DS, Anand MP, Raj KD, Raj PT, Yaswanth R, Yogesh S (2019) Electronic stick for visually impaired people with buzzer alert. Int J Recent Technol Eng (IJRTE). Randall D (2003) Living inside a smart home: a case study. In: Harper R (ed) Inside the smart home. Springer, Heidelberg, pp 227–246
3. Bai J, Lian S, Liu Z, Wang K, Liu D (2017) Smart guiding glasses for visually impaired people in an indoor environment. IEEE Trans Consum Electron 63(3):258–266
4. Bagwan SMR, Sankpal LJ (2015) VisualPal: a mobile app for object recognition for the visually impaired. In: 2015 international conference on computer, communication and control (IC4). IEEE, pp 1–6
5. Murali VN, Coughlan JM (2013) Smartphone-based crosswalk detection and localization for visually impaired pedestrians. In: 2013 IEEE international conference on multimedia and expo workshops (ICMEW). IEEE, pp 1–7

6. Ivanchenko V, Coughlan J, Shen H (2008) Detecting and locating crosswalks using a camera phone. In: 2008 IEEE computer society conference on computer vision and pattern recognition workshops. IEEE, pp 1–8
7. Hakobyan L, Lumsden Jo, O'Sullivan D, Bartlett H (2013) Mobile assistive technologies for the visually impaired. Surv Ophthalmol 58(6):513–528
8. Basori AH (2020) HapAR: handy intelligent multimodal haptic and audio-based mobile AR navigation for the visually impaired. In: Technological trends in improved mobility of the visually impaired. Springer, Cham, pp 319–334
9. Zhang DF (2017) An effective detection algorithm of zebra-crossing. In: International conference on electrical and information technologies for rail transportation. Springer, Singapore, pp 809–816
10. Wang S, Tian Y (2012) Detecting stairs and pedestrian crosswalks for the blind by RGBD camera. In: 2012 IEEE international conference on bioinformatics and biomedicine workshops. IEEE, pp 732–739
11. Wu X-H, Renjie Hu, Bao Y-Q (2019) Block-based Hough transform for recognition of zebra crossing in natural scene images. IEEE Access 7:59895–59902
12. Khaliluzzaman M, Deb K (2016) Zebra-crossing detection based on geometric features and vertical vanishing point. In: 2016 3rd international conference on electrical engineering and information communication technology (ICEEICT). IEEE, pp 1–6
13. Uddin MS, Shioyama T (2005) Bipolarity and projective invariant-based zebra-crossing detection for the visually impaired. In: 2005 IEEE Computer Society conference on computer vision and pattern recognition (CVPR'05)-workshops. IEEE, p 22
14. Meem MI, Dhar PK, Khaliluzzaman M, Shimamura T (2019) Zebra-crossing detection and recognition based on flood fill operation and uniform local binary pattern. In: 2019 international conference on electrical, computer and communication engineering (ECCE). IEEE, pp 1–6
15. Haider MM, Hoque MR, Khaliluzzaman M, Hassan MM (2019) Zebra crosswalk region detection and localization based on deep convolutional neural network. In: 2019 IEEE international conference on robotics, automation, artificial-intelligence and internet-of-things (RAAICON). IEEE, pp 93–97
16. Koester D, Lunt B, Stiefelhagen R (2016) Zebra crossing detection from aerial imagery across countries. In: International conference on computers helping people with special needs. Springer, Cham, pp 27–34
17. Sampathkumar J, Rajamani K (2013) Automatic detection of zebra crossing violation. In: Proceedings of the fourth international conference on signal and image processing 2012 (ICSIP 2012). Springer, India, pp 499–509
18. Alvarez S, Llorca DF, Sotelo MA (2013) Camera auto-calibration using zooming and zebra-crossing for traffic monitoring applications. In: 16th international IEEE conference on intelligent transportation systems (ITSC 2013). IEEE, pp 608–613
19. Zhai Y, Cui G, Gu Q, Kong L (2015) Crosswalk detection based on MSER and ERANSAC. In: 2015 IEEE 18th international conference on intelligent transportation systems. IEEE, pp 2770–2775
20. Kaggle, Crosswalk Dataset, from Davidsilvam (2021). https://www.kaggle.com/davidsilvam/crosswalkdataset

Intelligent Systems for Social Media

Twitter Sentiment Analysis Using Enhanced BERT

Suman Mann, Jyoti Arora, Mudita Bhatia, Ritika Sharma, and Rewangi Taragi

Abstract With the fast improvement of Internet social platforms, consumer evaluations have emerged as a critical foundation for clients to recognize merchandise and make decisions. New means of communication, such as microblogging, have evolved in the last decade, and a lot of information is provided by tweets and short messages, that are used to predict feelings of the users and their perception of what is going on around the globe. The most prominent sentiment in the tweet can be recognized by sentiment analysis techniques. Sentiment analysis is the method of extracting and recognizing the user's evaluations of products and models and has various approaches using machine learning algorithms to classify the emotion behind that text. This paper investigates the usage of Enhanced BERT models to recognize the sentiment behind the tweet. BERT or bi-directional encoder representations from transformers was designed to help computers understand the meaning of ambiguous language in text, by using the surrounding text to understand the context in which that text could have been written. For a successful evaluation using Enhanced BERT, the Kaggle SMILE dataset is considered and will be tested for emotions such as happiness, sadness, etc., and categorized as such. Experiments display that this version of the model achieves an accuracy of 0.96.

Keywords Sentiment analysis · BERT · Tweet sentiment analysis · Precision · Recall

S. Mann · J. Arora · M. Bhatia (✉) · R. Sharma · R. Taragi
Department of Information Technology, Maharaja Surajmal Institute of Technology, Affiliated to GGSIPU, New Delhi, India
e-mail: muditabhatia25@gmail.com

S. Mann
e-mail: sumanmann@msit.in

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_21

1 Introduction

In recent times, the exceptional boom of social networks, private blogs and assessment apps has made the availability of user-generated content material very freely available. Taking this into consideration, it can be said that human beings are now confident enough to express their opinions publicly on such platforms. So, the analysis of this content material presents essential statistics regarding the thoughts of people about different topics.

Twitter is a well-known Web site where users can express themselves through their posts, which are short tweets, with a limit of 140 characters, and express their opinion on important topics and share their day-to-day activities. As a result, people use slang, acronyms, emoticons and abbreviated forms to condense their statements. People also use sarcasm and polysemy to express their viewpoints. With over 100 million users and over 500 million tweets sent every day, Twitter has established itself as a prominent microblogging platform [1].

It is also very useful for categorizing reviews of various markets, which is why Twitter is considered in the proposed model to predict people's doubts and feelings about the different issues going on in the world. Covering almost the entire urban population, Twitter has become an indispensable part of various sentiment checking techniques, and the proposed version takes advantage of the large viewership and usership of the site, to recognize and classify the emotion of its users regarding the daily news, be it the Coronavirus, lockdown, closure of institutions and organizations or just the daily political news. The proposed Enhanced BERT model works to analyze and categorize sentiments of users using their tweets, and the performance of this model is evaluated against the emotion parameters and compared with other existing models, to check whether the desired improved results have been achieved or not.

For this evaluation, the dataset used for the proposed Enhanced BERT model is the Kaggle SMILE dataset, which covers tweets from May 2013 to June 2015, from Twitter handles affiliated to British Museums, with emotions such as happy, sad, disgust, surprise and irrelevant. Thus, this version of the Enhanced BERT model gives better results than its predecessor approaches, because it follows comparative analysis on performance metrics. Unlike the pre-vailing NLP models, which examine the textual content in only one direction, from left to right or vice-versa, Enhanced BERT scans the total series of words all at once; therefore, it is a faster and much more feasible method. The Enhanced BERT model is a steppingstone toward a newer and better approach, to analyze and categorize sentiments of the users even on short datasets, and thus can be deployed for larger twitter datasets, to analyze the emotion with which the user is blogging his or her opinion on the Web site. On this idea of Enhanced BERT, this paper proposes a powerful sentiment analysis approach.

The following section explains the previous works that have been done in these field, which have laid the foundation for successfully carrying out the research done for the proposed model, and the later sections discuss the architecture and techniques of Enhanced BERT, explaining the various preprocessing tactics that are used in the two-step pipeline to convert the text into machine-readable text, by removing the emojis, symbol, hashtags and other twitter jargon. Also, the experimental analysis and results performed using the Enhanced BERT model based on various classifiers such as precision, recall and F1 score, which are performance metrics are explained in detail. These metrics help us to calculate the accuracy with which that emotion was recognized. The results achieved by experimental analysis of this model were comparatively analyzed based on their F1 scores. The final section concludes that the Enhanced BERT model which provides a new powerful approach in the field of sentiment analysis.

2 Literature Survey

Many ML and NLP approaches are used today to help us understand the text and take our valuable information from that. Dhruv Rathee and Suman Mann have used ML and deep learning technologies to detect phishing attacks which requires the ML model to identify the language of such mails and mark it as suspected attack [1].

Few sentiment analyses are done by characterizing posts about electric products like mobile phones using ML approach. Many ML approaches are used in such cases to recognize the sentiment with which the text has been written [2]. Newer research tried to preprocess the dataset, then extracted an adj from the dataset that had important meaning (feature vector), used this list and subjected ML algorithms such as Naïve Bayes, Maximum Entropy and SVM. At last, they measured the performance in terms of recall, precision and accuracy [3].

Another approach was applied where the tweet was preprocessed and classified based on its content as happy, sad and no change; and compared the performance of the algorithm based on precision and recall [4]. But recognizing emotion with which tweet has been written and separating the categories is also an important part, which these approaches failed to do.

The technique applied could only recognize various possibilities of emotions with which the tweet could have been writ-ten. Andrea Chiorrini and Alex Mircoli investigated a similar model and deployed it with different classifiers, achieving an accuracy of 92% [5]. They tested their model for emotion recognition and sentiment classification. This model achieved a high accuracy and formed the basis for many further research grounds on this aspect.

Duyu Tang, Nan Yang, Furu Wei proposed a model to check the sentiment-specific embedding. Their work highlighted the use of NLP for learning task word embedding and was highly useful to predict the sentiment of particular words in the tweet, whether they were expressed with anger, grief, sorrow, etc. [6]. Walaa Medhat, Ahmed Hassan, Hoda Korashy gave a sophisticated update on previous approaches to sentiment analysis. They tried to group together many datasets and then distinguish the different emotions and tabulate them so [7].

Marco Pota, along-with his associates, tested state-of-the-art techniques to check existing BERT models for newer datasets, including emojis and the Italian language.

They gave a methodological approach to build newer models for languages other than English. The research also tried opinion mining to recognize people's opinions. The research also tried opinion mining to recognize people's opinions and attitudes for day-to-day situations. It includes extracting the sentiment from the user's tweet and then classifying its polarity [8].

Vasu Negi, Suman Mann and Vivek Chauhan proposed an approach to recognize the Devanagari character recognition using artificial neural network which can be further be improved if a similar BERT model is developed with all the resources and books written in that language [9]. The approach followed by Suman Mann, Anish Batra and Guneet Singh Sethi for their research for personalized automation using artificial intelligence was based on the communication of data from one end to radio via wireless media and was implemented for the micro-processing stage for this research [10].

Suman Mann and Sakshi Hooda observed that recent trends of incorporating computing technology into the medical fields, keeping track of patients, diseases, and their treatments, has led to accumulation of large amounts of data which can essentially be successfully replaced as classifiers for predicting the severity of the patient's condition, by application of data analysis and data mining techniques [11]. This also formed the basis for the research proposed, as it proved to be an important step in realizing that data analysis could be done for unique and peculiar datasets as well. Similar approaches proposed by Deepa Gupta and Suman Mann introduced intelligent devices for communicating data with other environments using sensor networks [12], while others proposed an effective and improved method for securing the preprocessing techniques for enhanced results [13].

While many researchers have worked toward enhancing the performance of their techniques in this field, to test out the sentiment behind the text and to classify that correctly, the area where the research faced a pit stop was the accuracy of their model. The proposed Enhanced BERT model proves to be a steppingstone toward a newer and better approach to analyze and categorize sentiments of the users, based on the accuracy achieved for various emotions amidst the large dataset they were tested for.

3 Proposed Enhanced BERT Model

In the proposed version of sentiment analysis, the model ex-tracts tweets from the Twitter site, and the accumulated tweets are then be subjected to preprocessing under a supervised set of rules at the saved data as is shown in Fig. 1. This approach follows the previously proposed models for categorizing into six groups of varying ages, using certain facial recognition features, involving the same procedural pipeline, namely preprocessing, feature extraction and classification, as is used in the proposed model [14]. The steps for analyzing the tweet make use of preprocessing tactics that transform the Twitter text into plain text, such as emoji, and classifying this text on the pre-trained model [15].

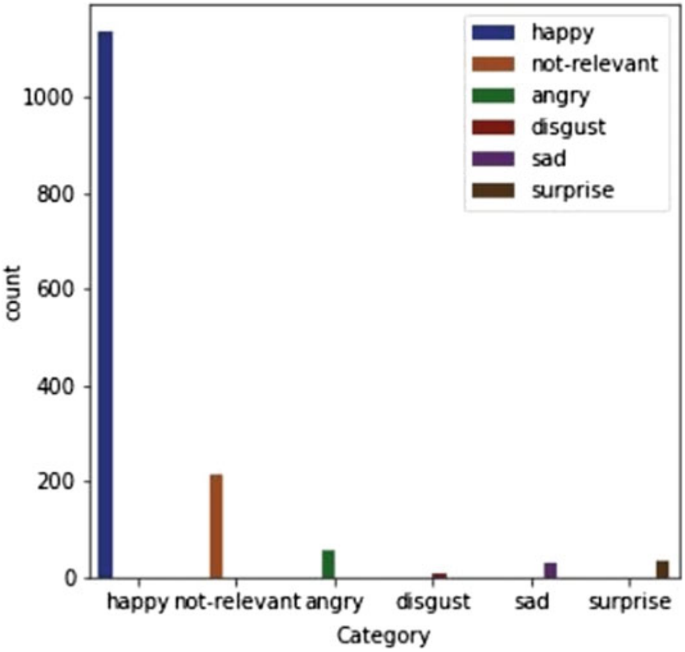


Fig. 1 Emotions on tweet intensity dataset: distribution of class of the test set. Categories: anger, disgust, happiness, surprise and sadness

Preprocessing strategies are used to transform the textual content so that it may be easily subjected to required set of rules under the proposed model. Tweets have several unique features, i.e. emoji, user mentions, hashtags as well as regular Internet constructs and different noisy sources, random numbers, etc., which must be preprocessed so that the machine can understand the underlying sentiment behind these tweets [16].

BERT preprocessing modules includes several preprocessing techniques that helps to save a lot of time. This technique is called BERT to Analyze Twitter Data [17].

The BERT architecture is recognized as one of the most popular models among the modern language modeling architectures. Its generalization functionality is such that it could be tailored to specific tasks in line with specific needs, be it relation extraction or Named Entity Recognition (NER), query answering or sentiment analysis [17].

Following are some insights into the BERT architecture:

There are two inputs, the first from word tokens and the second from segment layer.

After adding these two, they result in a position embedding, followed a layer normalization and dropout.

The next step is Multi-head Self-Attention layers which are 12 in number, and each is having 9 steps each.

After this, the two outputs are—one for Next Sentence Prediction (NSP) and one for Masked Language Modeling (MLM) [18].

In emotion recognition and classification, precision and recall are performance metrics that are applied to the data retrieved from a collection of text, and precision and recall are the two building precision and recall metrics into a single metric. At the same time, the F1 score is a measure of the test's accuracy [7]. The precision and recall are calculated and the F1 score is evaluated. Based on blocks of the F1 score. The goal of the F1 score is to combine the F1 score of the emotions, the comparison table is formulated, and the performance of the model can be easily evaluated. A good improvement in the F1 score shows good achieving accuracy for that emotion, whereas a decline or a slow improvement in the score shows that that emotion was not very well evaluated or that emotion could not be recognized as well as the others in the dataset.

Unlike the prevailing NLP models, which examine the textual content in only one direction, from left to right or vice-versa, Enhanced BERT scans the total series of words all at once. Enhanced BERT employs a transformer that is a device for recognizing relationship among words in a dataset. An Enhanced BERT system basically consists of various encoders and a decoder. The encoders serve the function of taking the input while the decoder generates predictions. Since the essential motive of BERT is to generate a pre-trained model, encode takes priority over the decoder.

The SMILE Twitter Emotion dataset, which is used in this study, is a collection of tweets mentioning 13 Twitter handles affiliated with British museums that was collected between May 2013 and June 2015. It has 3085 tweets, each expressing one of five emotions: rage, disgust, delight, surprise, and sadness [19]. This system is more powerful than the prevailing ones, as it may be feasible to realize how the records decided from the illustration of the result may have an effect in a specific field.

4 Experimental Results and Analysis

The SMILE Twitter Emotion dataset, which contains 3085 tweets labeled regarding the five emotions as shown in Fig. 2, was used for evaluating the performance of the proposed architecture on the job of sentiment. The model was assessed using following metrics: classification of accuracy and the F1 score. Each occurrence in the dataset is linked to a label emotion as well as a metric called intensity, which measures the emotion's intensity [20].

Let x_{ij} be the data number belonging to jth class, classified as ith class [21].

Let C be the classes number and N be the total amount of data. The accuracy achieved by a classifier is computed as [21]:

$$\text{accuracy} = \left(\frac{1}{N}\right)\sum_{i=1}^{C} x_{ii} \tag{1}$$

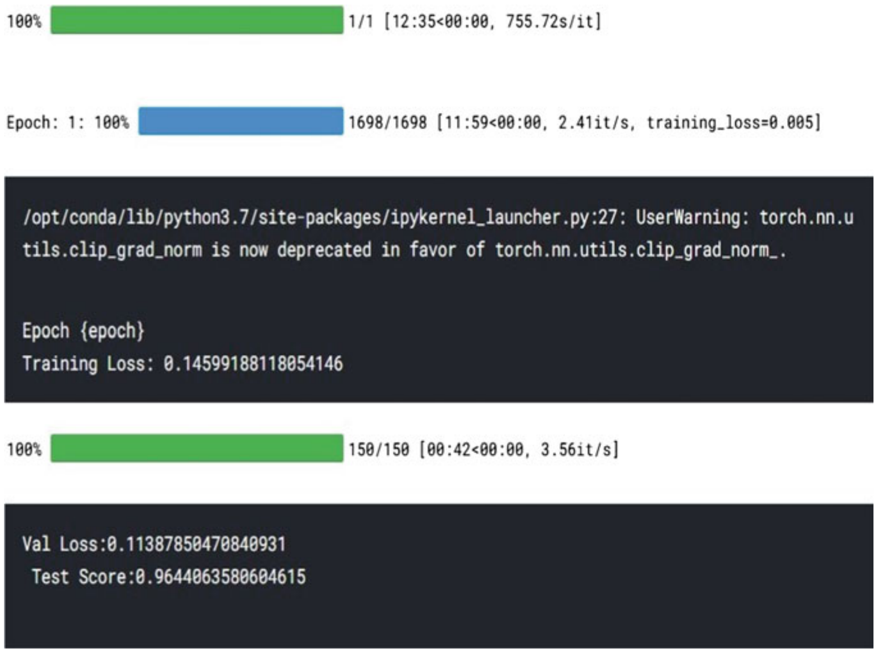


Fig. 2 Accuracy of the model ≈ 96.44%

Precision and recall of ith class are calculated as follows [21]:

$$\text{precision}_i = \frac{x_{ii}}{\sum_{j=1}^{C} x_{ij}} \tag{2}$$

$$\text{recall}_i = \frac{x_{ii}}{\sum_{j=1}^{C} x_{ji}} \tag{3}$$

F1 score of ith class is equal to [20]:

$$F_{1i} = 2 \cdot \frac{\text{precision}_i \cdot \text{recall}_i}{\text{precision}_i + \text{recall}_i} \tag{4}$$

So, the F1 score achieved by the model is defined as the average of F_{1i} [21]:

$$F_{1i} = \frac{1}{C} \sum_{i=1}^{C} F_{1i} \tag{5}$$

The comparative F1 results for the models on the sentiment emotion categories of happy, sad and anger are presented in Table 1.

Table 1 Comparison between different techniques based on their F1 score

Technique used	F1 score		
	Anger	Happy	Sad
Baseline-STLM	0.670	0.558	0.599
SS-STLM	0.712	0.597	0.793
BERT	0.736	0.930	0.808

The model achieved best results on the happy emotion and worst results on anger emotion, as can be seen from table below. Thus, it can be proved that the model has made substantial improvements in recognizing the various emotions behind the tweets, than the previous models could.

The results indicated in Fig. 2 show that the accuracy for the BERT model is 96.44% which means the model performed well even on small datasets. Thus, within a short period of time, a BERT model that works on test data with a good score can be built.

5 Conclusion

The proposed model was able to construct a practical sentiment analysis approach of Twitter system based on the Enhanced BERT model. The proposed technique is based on a two-step procedure, with the first step involving a series of preprocessing techniques to convert Twitter jargon, such as emoji into plain text, and the second step involving an updated version of BERT that had been pre-trained on plain text to fine-tune and classify tweets according to their polarity, such as happy, sad, angry, surprised and not relevant or irrelevant tweets, and the second step utilizing a version. Experiments display that this version of the model achieves an accuracy of 0.96. Thus, this version of the Enhanced BERT model gives good results even on short datasets, and thus can be deployed for larger twitter datasets, to analyze the emotion with which the user is blogging his or her opinion on the Web site. On this idea of Enhanced BERT, this paper proposes a powerful sentiment analysis approach.

References

1. Rathee D, Mann S, Detection of e-mail phishing attacks—using machine learning and deep learning. Int J Comput Appl 183(47). https://doi.org/10.5120/ijca2018918026
2. Joselson N, Hellen R (2019) Emotion classification with natural language processing
3. Gautam G, Yadav D (2014) Sentiment analysis of Twitter data using machine learning approaches and semantic analysis. IEEE, pp 103–110
4. Radford A, Narasimhan K (2019) Improving language understanding by generative pre-training, pp 11–35
5. Chiorrini A, Mircoli A, Diamantini C, Potena D (2021) Emotion and sentiment analysis of tweets using BERT, pp 15–32

6. Tang D, Wei F, Qin B, Yang N, Liu T, Zhou M (2015) Sentiment embeddings with applications to sentiment analysis. IEEE Trans Knowl Data Eng 28(2):496–509
7. Medhat W, Hassan A, Korashy H (2014) Sentiment analysis algorithms and applications: a survey. Ain Shams Eng J 5:1093–1113
8. M. Pota, M. Ventura, R. Catelli, and M. Esposito (2020) An effective BERT-based pipeline for Twitter sentiment analysis: a case study in Italian, pp 112–117
9. Negi V, Mann S, Chauhan V (2009) Devanagari character recognition using artificial neural network. Int J Eng Technol 2161–2167
10. Batra A, Sethi GS, Mann S (2019) Personalized automation of electrical and electronic devices using sensors and artificial intelligence—"The Intelligizer System". Computational intelligence: theories, applications and future directions—volume I. AISC, vol 798
11. Hooda S, Mann S (2020) Sepsis-diagnosed patients' in-hospital mortality prediction using machine learning: the use of local big data-driven technique in the emergency department. Int J Grid Distrib Comput 13(1)
12. Mann S, Gupta D, Arora Y, Chugh SP, Gupta A (2021) Smart hospitals using artificial intelligence and internet of things for COVID-19 pandemic. Smart healthcare monitoring using IoT with 5G
13. Gupta D, Jha SK, Mann S (2021) Internet crimes—it's analysis and prevention approaches. In: 2021 9th international conference on reliability, Infocom technologies and optimization (trends and future directions) (ICRITO), 2021, pp 1–4. https://doi.org/10.1109/ICRITO51393.2021.9596396
14. S. Mann et al (2018) Estimation of age groups using facial recognition features. Int J Eng Comput Sci 23945–23951
15. Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. ArXiv:1810.04805, no 2, pp 18–40
16. Mancini M, Mircoli A, Potena D, Diamantini C, Duca D, Toscano G (2020) Prediction of pellet quality through machine learning techniques and near-infrared spectroscopy. Comput Ind Eng 15–31
17. Mishne G (2005) Experiments with mood classification in blog posts. Live J 1104–1153
18. Tang D, Wei F, Yang N, Zhou M, Liu T, Qin B (2014) Learning sentiment-specific word embedding for twitter sentiment classification. In: Proceedings of the 52nd annual meeting of the Association for Computational Linguistics, vol 1, pp 1555–1565
19. Kaggle. SMILE Twitter emotion dataset—useful for learning sentiment analysis
20. Chang L-H, Pyysalo S, Kanerva J, Ginter F (2019) Towards fully bilingual deep language modeling. ArXiv:2010.11639, pp 5–16
21. Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. ArXiv:1810.04805

A Study on Sentiment Analysis of Twitter Data in Marathi Language for Measuring Depression

Mrunmayee Jakate, Snehal Lavangare, Nirmiti Bhoir, Aarushi Das, and Deepali Kadam

Abstract Sentiment analysis is nothing but an information retrieval system based on natural language processing (NLP). It is a machine learning algorithm. The rudimentary prototype of sentiment analysis comes under the classifier problem where the negative, positive, and neutral are the classes we would be expecting as an output, given text (tweets in our case) as an input. Feature selection methods (information gain and chi square) and feature extraction will be performed to get the output sentiment. After studying three classifiers, support vector machine (SVM), K-nearest neighbour (KNN), and Naive Bayes, the latter algorithm, i.e. Naive Bayes being most accurate among them would be considered for implementation. As mentioned, we would be focusing on major three output sentiment classes: positive, negative, and neutral. Further, we are getting into a detailed analysis of tweets reflecting a depressive state of mind due to sadness. Though our ultimate goal is to predict the mental health of a user by observing the consistency and frequency of tweets along with its sentiment analysis, for now, we are just dealing with one tweet per user at a time.

Keywords Sentiment analysis · Marathi text · Depression · Mental health · Devanagari · Information gain · Chi square

1 Introduction

Just one click and anyone can express support or resistance towards anything happening in this world! Everyone is in a rush to put not only their opinion but also their intense feelings, whether happy or sad on a table. Social media! Twitter! One of the powerful tools to express our emotions, let it be happiness or be it grief. Gone are those days, when people used to write letters or articles to convey their thoughts. Now the survey shows, even though the character limit is 280 (earlier it was 140), most of the tweets are just 34 chars long. This is because of the abbreviations people tend to

M. Jakate · S. Lavangare · N. Bhoir (✉) · A. Das · D. Kadam
Information Technology, Datta Meghe College of Engineering Airoli, Navi Mumbai, India
e-mail: nirmitibhoir.mumbai@ieee.org

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_22

use while tweeting (Great: Gr8, भा.पो. : भावना पोहोचल्या) on Twitter. Thus, delivering more information about what they are going through, suffering from, or enjoying, in one sentence itself. Life has not only become fast but quite self-centred too. Hence, people count on social media to seek attention and responses for their thoughts, opinions, and emotions. Unfortunately or fortunately, people find it more comfortable and safe to showcase their loneliness, disappointment, sufferings, and disagreement on a public platform. In our paper, we are conducting fine-grained analysis by operating on low-level signals such as sentiments, opinions, emotions, areas of interest of an individual expressed through posts on Twitter to address the mental health issues of society by making high-level coarse-grained predictions regarding it. Depression and suicidality are major concerns nowadays, and we opted to take this as our topic of study. So we have mainly categorized negative emotions into sadness and anger. Sadness is further branching into grief, depression, and anger into abuse, aggression. In this paper, after obtaining the negative sentiment output, we will be mainly focusing on treating tweets reflecting sadness. The English language has already seen an adequate amount of work, while the research in the Marathi language is comparatively less. Marathi, being the most prominent language in India and the official language of Maharashtra, we have decided to go with this language.

Application of our sentiment analysis incorporates social media monitoring where after asserting their emotional illnesses we would be providing an option to get connected with counsellor and mental health specialists. The track of intensity, consistency, and frequency of the tweets posted by the user will be monitored to ensure accurate analysis.

2 Related Work

In 2021, Nithin P, Owais Ahmed, Aditya Jason Hans, Amaan Faraaz and Prof. Madhusudhan M. V. surveyed a Depression Detection Model Based on Sentiment Analysis on Twitter API [1]. In this research paper, the authors have presented their work by categorizing the tweets from Twitter into three types of sentiments: which are positive, negative, and neutral. If the tweet is classified as positive, it shows that the person is not likely to have anxiety or depression. Whereas if the tweet is neutral, it means that the user may or may not be depressed but can be susceptible to depression. Lastly, negative is the most basic level where depression and anxiety symptoms are identified through tweets of users. The database for this model was collected online straight from twitter using In house Twitter API. Naive Bayes algorithm was found to be the most accurate one.

In April 2021, Atharva Kulkarni, Meet Mandhane, Manali Likhitkar, Gayatri Kshirsagar, and Raviraj Joshi published a paper on L3CubeMahaSent: A Marathi

Tweetbased Sentiment Analysis Dataset [2]. L3CubeMahaSent is the first available dataset of tweets in the Marathi language. The dataset consists of tweets that have been collected from various Marathi personalities' Twitter accounts. There are approximately 16,000 annotated tweets categorized into positive, negative, and neutral. There is an equal amount of positive, negative, and neutral tweets thereby making the dataset unbiased. Two (positive and negative only) and three class sentiment analysis is performed on the dataset using deep learning models like CNN and BiLSTM. The accuracies of these methods are also compared, the maximum three class accuracy obtained is 83.14 and 93.13% for two class sentiment analysis.

In February 2020, K. Senthamilselvan, D. Aneri, C. Adithya, P. Kani Kumar presented a paper on Twitter Sentiment Analysis using Machine Learning Techniques [3]. The method proposed by the authors comprised of four steps—data preprocessing, feature extraction, balancing and scoring, and sentiment classification. The algorithm for each step of this process is also presented by the authors. Data preprocessing aimed to remove all the irrelevant data inside the tweets like punctuation marks, @ characters, stopwords, URL, etc. Feature extraction is performed using TF-IDF, Count_Vectorizer, and Word_to_Vec algorithms. Balancing and scoring involved assigning polarity to each tweet and tagging it as positive, negative, or neutral. Machine learning algorithms like multinomial Naive Bayes and logistic regression are used in sentiment classification which involves categorization of tweets. The TF-IDF vectorizer when used with logistic regression provided an accuracy of 88.1 and 83% with multinomial Naive Bayes. Similarly, the count vectorizer gave an accuracy of 88.4% with logistic regression and 87% with the multinomial Naive Bayes algorithm. Airline sentiment and IMDB datasets are used in this model. In conclusion, the top performance is achieved by the logistic regression algorithm with the count vectorizer. In order to accomplish finer outcomes, deep learning methods can also be implemented.

In March 2020, Sonali Rajesh Shah, Abhishek Kaushik, Shubham Sharma and Janice Shah presented a paper on Opinion-Mining on Marglish and Devanagari Comments of YouTube Cookery Channels Using Parametric and Non-Parametric Learning Models [4]. Their methodology involves six steps, that is starting from (i) extracting data using YouTube API, (ii) labelling and categorization of the data, (iii) preprocessing data, (iv) vectorization, (v) applying machine learning models, and lastly (vi) testing the data. In order to extract data, they analysed Marathi cookery channels on YouTube using YouTube API. They achieved 55.54% of accuracy using BNB along with TF-IDF Vectorizer, 55.54% using BNB along with TF-IDF vectorizer and 54.10% using BNB along with hashing vectorizer.

In 2019, Priyanka Arora and Parul Arora surveyed mining of Twitter data for depression tweets [5]. They classified the tweets based on keywords such as depression, anxiety, and mental illness followed by preprocessing the data. For classifying the tweets, multinomial Naïve Bayes and support vector regression (SVR) algorithms are executed. The SentiWordNet dictionary was used to calculate the positivity and negativity of the sentence. These sentiment scores are calculated using TextBlob and SentiWordNet dictionary, POS tags, POS vector, POS score, and n-grams. The database for this model was extracted from Twitter streaming.

In 2019, Jini Jojo Stephen and Prabu P surveyed the twitter data by detecting the magnitude of depression in twitter users using sentiment analysis [6]. The motive of this paper is to present a methodical procedure that can detect the level of depression in Twitter. In this research paper, the authors have proposed their work by mentioning how different emotions will be identified and a weighted score for each tweet will be calculated, with the average score for a single day determining the final magnitude. The prime focus of this paper is to come up with a system which can check whether the twitter users are experiencing depression based on their former tweets from months ago. The main objective of this paper is to develop a system that can identify depression in twitter users based on tweets posted by them over a period of time. The depression magnitude was calculated using these steps, first by base emotion calculation, and checking the sentiment score evaluation and then finally. For each emotion that was assessed, weights were obtained which were then used to calculate the final depression magnitude.

In June 2017, Sujata Deshmukh, Nileema Patil, Surbhi Rotiwar, Jason Nunes published a paper on Sentiment Analysis of Marathi Language [7]. In this paper, the authors have presented a corpus-based approach which involves two modules, i.e. (i) feasible corpus for Marathi language from English SentiWordNet and (ii) second module presents mapping of keywords. They achieved 60–70% of accuracy using Yandex translator. Thus for Asian, vernacular languages, the translation of bigger sentences are only partially accurate. They also included separate exception classes to eliminate special characters. This method concentrated on the development of a huge and diversified latest corpus along with efficient mapping of data and formation of accurate sentiments.

In April 2016, Vishal A. Kharde and S. S. Sonawane published a paper on Sentiment Analysis of Twitter Data: A Survey of Techniques [8]. This paper provided a deep insight into the approaches of performing sentiment analysis of Twitter data. The various steps in sentiment analysis which include preprocessing of data, feature extraction, training, and classification are explained in detail. They used the datasets of Twitter publicly available on the Stanford website. The machine learning approaches—supervised and unsupervised learning techniques are described. In this approach, the tweets are categorized into classes using classifiers like Naive Bayes, maximum entropy, and support vector machine. The authors provided a detailed comparison between different machine learning, lexicon based, cross-lingual, and cross-domain techniques with their accuracy percentages. The Naive Bayes algorithm provided an accuracy of 76.44% with the bigram dataset.

In October 2015, Lai Po Hung, Rayner Alfred, Mohd Hanafi Ahmad Hijazi, and Joe Henry Obit published a paper on "A Review on Feature Selection Methods for Sentiment Analysis" [9]. This paper presented the various feature selection techniques like information gain (IG), mutual information, chi square, document frequency, and mutual information that can be implemented to increase the accuracy of the classifier algorithm. The presence of multiple features in a dataset can lead to poor execution of the machine learning algorithm, which might lead to incorrect output.

In November 2014, Aliza Sarlan, Chayanit Nadam, and Shuib Basri published a paper on Twitter Sentiment Analysis [10]. Twitter is a very popular microblogging site in which users communicate via tweets. The sentiment analysis of these tweets can be used for various objectives. Sentiment analysis is the procedure of extracting the polarity of a sentence. The two main approaches given are machine learning and lexicon approach. There are four techniques to assign polarity—natural language processing (NLP), case-based reasoning (CBR), artificial neural network, and support vector machine (SVM). SVM is a machine learning algorithm that gives an accuracy of 70–81.3% on the tested dataset. Using the Twitter API, the tweets are collected and categorized as positive or negative. In conclusion, the use of machine learning algorithms along with NLP techniques will aid in achieving precise results.

S. No.	Name: research paper	Authors	Language	Datasets	Findings	Gaps
1	Depression detection model based on sentiment analysis on Twitter API	Nithin P, Owais Ahmed, Aditya Jason Hans, Amaan Faraaz, Prof. Madhusudhan MV	English	Collected tweets: Twitter API	This paper provides us a seed list of initial 60 words to represent anxious depression. Explained the classification of tweets accordingly into 3 categories—positive, negative and neutral. Compared the accuracy and runtime and gave the most accurate algorithm according to the tests	This paper did not mention any formulas regarding the algorithm that they have mentioned
2	L3CubeMahaSenI: a Marathi Tweet-based sentiment analysis dataset	Atharva Kulkarni, Meet Mandhane, Manali Likhitkar, Gayatri Kshirsagar and Raviraj Joshi	Marathi	D3CubeMarathiSent	This paper presented the first universally available Marathi twitter dataset consisting of 16,000 tweets categorized as positive, negative and neutral. Deep learning models like CNN and BERT are used to calculate the accuracy of the model	–
3	Twitter sentiment analysis using machine learning techniques	K. Sentamilselvan, D. Aneri, A. C. Athithiya, P. Kani Kumar	English	–	Logistic regression using Word_2_Vec	More features topics such as POS tagging, negation handling could have been explained
4	Opinion-mining on Marglish and Devanagari comments of YouTube Cookery Channels using parametric and non-parametric learning models	Sonali Rajesh Shah, Abhishek Kaushik, Shubham Sharma and Janice Shah	Marathi	YouTube API	MLP and BNB works finest with the count vectorizer with the accuracy of 60.68 and 60.60% accusingly. For testing dataset they have carried out tenfold-cross-validation and using test statistics	Unstable list of stopwords, removed emotions and neglected sentiment

(continued)

(continued)

S. No.	Name: research paper	Authors	Language	Datasets	Findings	Gaps
5	Mining Twitter data for depression detection	Priyanka Arora, Paul Arora	English	Prepared dataset from Github	The research conducted by the authors revealed that SVM, K clustering and multinomial Naive Bayes are the machine learning algorithms used to achieve good accuracy. With their own dataset of 3754 tweets they achieved an accuracy of 79.7% with SVM and 78% with Naive Bayes	–
6	Detecting the magnitude of depression in Twitter users using sentiment analysis	Jini Jojo Stephen, Prabu P	English	Tweets from Twitter	Negative magnitude is measured till a particular period and then the mental health of the user is stated	–
7	Sentiment analysis of Marathi language	Sujata Deshmukh, Nileema Patil, Surbhi Rotiwar, Jason Nunes	Marathi	English sentiword	POS tagging, negation handling, identifying metaphors	Limited for sentence level sentiment analysis, haven't mentioned any specific ML algorithm, limited size of corpus
8	Sentiment analysis of Twitter data: a survey of techniques	Vishal A Kharde, S. S. Sonawane	English	Publically available dataset of Stanford University	Various feature vectors like POS, negation, ngram, position of terms, words and their frequencies are considered while calculating the sentiment which increases the accuracy. Various machine learning algorithms such as SVM, Naïve Bayes, maximum entropy are used and their performance is compared using the accuracy metric. SVM algorithm gave the maximum accuracy of 77.73% when used with a large dataset with unigram + bigram + trigram feature enabled	The paper mentions about multiclass sentiment classification: less positive, strong positive, less negative, etc., but the approach to this procedure is not given

(continued)

(continued)

S. No.	Name: research paper	Authors	Language	Datasets	Findings	Gaps
9	A review on feature selection methods for sentiment analysis	Lai Po Hung, Rayner Alfred, Mohd Hanafi Ahmad Hijazi and Joe Henry Obit	English	–	This paper aims on the feature selection methods of analysing sentiments. The three selection methods are as follows: filter methods, wrapper methods, and embedded methods. Under filter methods document frequency, information gain, chi-square, mutual information, odds ratio, categorical proportional difference are explained	The accuracy of the various feature selection methods are not compared
10	Twitter sentiment dialysis	Aliza Sarlan, Chayanit Nadam, Shuib Basri	English	Twitter API	There are 4 techniques to assign polarity—natural language processing (NLP), cash-based reasoning (CBR), artificial neural network, support vector machine (KVM). The use of machine learning algorithms along with NLP techniques will help in achieve accurate results	–

3 Proposed Methodology

In our proposed system, input to the classifier is Marathi tweets, and output is from three basic classes: positive, neutral, and negative (Fig. 1).

3.1 Collection of Marathi Tweet Data

We have gathered some of our dataset from Twitter, but due to shortage of Marathi tweets some of them have been framed by our team members (Fig. 2).

3.2 Preprocessing of Data

While posting tweets the users generally are not concerned about the grammar or spelling mistakes in their sentences. Hence, a lot of redundant data is present which need to be cleaned, else it may pose difficulty while analysing the sentiment. This preprocessed data will be passed to the processing step as an input. There are four main steps for preprocessing of data.

3.2.1 Sentence Segmentation

A paragraph of string of given language will be divided into separate sentences. We will separate the set of sentences based on the delimiter which is full stop (.) (पूर्णविराम).

3.2.2 Tokenization

In this step, we will remove all the punctuation marks (विरामचिन्हे) except exclamation and question mark. In tokenization, we are not eliminating exclamations marks and question marks. This is because, exclamation mark is an inevitable punctuation while showing emotions and expressions while writing a sentence. Also, sometimes question mark may act as an important factor in rhetorical question like: "“कधी कधी वाटतं, आयुष्य खरंच सुंदर आहे का?", "माझी खरंच कोणाला गरज आहे का?", "मी जगण्याच्या लायकीचा आहे का?". Here, the person is not asking questions, but indicating self-doubt.

Also extra characters and hyperlinks are removed with correction of spelling mistakes and replacing emoticons with its sentiment (Fig. 3).

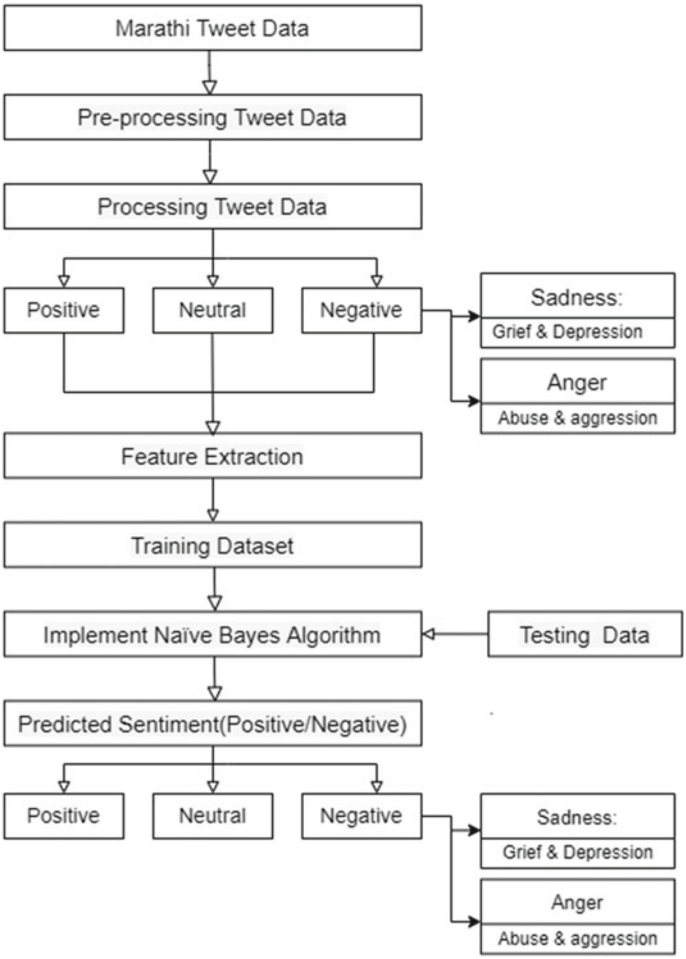


Fig. 1 Flowchart of proposed method

3.2.3 Stopword Removal

The words which are not relevant while calculating the sentiment of a sentence are stopwords.

In Marathi language usually the conjunctions (उभयान्वयीअव्यय : आणि, पण, किंवा, मात्र, त्यामुळे) and pronouns (सर्वनाम : मी, आम्ही, ते, त्या, हा, ही, जे) are treated as stop-words (Fig. 4).

Tweet	Sentiment
सगळे रस्ते बंद झाले आहेत.	negative
हा एकांत मला खायला उठत आहे.	negative
हा एकांत खूप त्रासदायक आहे.	negative
हा एकांत एक दिवस माझा जीव घेईल.	negative
हा एकांत मला नकोसा झाला आहे.	negative
मला आता हा एकांत सहन होत नाही आहे.	negative
हा एकटेपणा मला खायला उठत आहे.	negative
हा एकटेपणा खूप त्रासदायक आहे.	negative
हा एकटेपणा एक दिवस माझा जीव घेईल.	negative
हा एकटेपणा मला नकोसा झाला आहे.	negative
मला आता हा एकटेपणा सहन होत नाही आहे.	negative
कोणती आशाच उरली नाही आहे.	negative
जगण्यातली मजा हरवली आहे.	negative
जगणे नकोसे झाले आहे.	negative
देव अस्तित्वातच नाही आहे.	negative
आपल्या लोकांवर विश्वासच राहिला नाही आहे.	negative
आपण फक्त ऐकून घ्यायचं, बोलणाऱ्यांच्या जगात ऐकण्याला खूप मागणी आहे.	negative
सगळं उध्वस्त करून जातात वादळं मग ते बाहेरचे असू द्या किंवा मनातलं!	negative

Fig. 2 Dataset

	Before Punctuation Removal	After Punctuation Removal
0	सगळे रस्ते बंद झाले आहेत.	सगळे रस्ते बंद झाले आहेत
1	हा एकांत मला खायला उठत आहे.	हा एकांत मला खायला उठत आहे
2	हा एकांत खूप त्रासदायक आहे.	हा एकांत खूप त्रासदायक आहे
3	प्रत्येक गावात अस एक स्मशान असायला हवं, जिथे आठवणी जाळता येतील....!	प्रत्येक गावात अस एक स्मशान असायला हवं जिथे आठवणी जाळता येतील
4	फक्त आठवणी मनात उरलेल्या असतात, काही चांगल्या काही वाईट......	फक्त आठवणी मनात उरलेल्या असतात काही चांगल्या काही वाईट
5	सगळंच कसं टोचत आठवणी पण काटे पण...	सगळंच कसं टोचत आठवणी पण काटे पण
6	साऱ्या फितूर होऊनि आठवणी , आज ओवीवळ पाझरती ..!	साऱ्या फितूर होऊनि आठवणी आज ओवीवळ पाझरती
7	राहील्या त्या आठवणी. :'(	राहील्या त्या आठवणी
8	गेले ते दिवस आणि राहिल्या त्या फक्त आणि फक्त आठवणी...!!!	गेले ते दिवस आणि राहिल्या त्या फक्त आणि फक्त आठवणी
9	दानोळी ज़िल्हा परिषद - स्वाभिमानी विजयी	दानोळी ज़िल्हा परिषद स्वाभिमानी विजयी
10	भारतरत्न डॉ. बाबासाहेब आंबेडकर यांच्या ६५ व्या महापरिनिर्वाण दिनानिमित्त अभिवादन!	भारतरत्न डॉ बाबासाहेब आंबेडकर यांच्या व्या महापरिनिर्वाण दिनानिमित्त अभिवादन

Fig. 3 Preprocessing of data—punctuation removal

	Before Stopword Removal	After Stopword Removal
0	सगळे रस्ते बंद झाले आहेत	सगळे रस्ते बंद
1	हा एकांत मला खायला उठत आहे	एकांत मला खायला उठत
2	हा एकांत खूप त्रासदायक आहे	एकांत खूप त्रासदायक
3	प्रत्येक गावात अस एक स्मशान असायला हवं जिथे आठवणी जाळता येतील	प्रत्येक गावात स्मशान हवं जिथे आठवणी जाळता येतील
4	फक्त आठवणी मनात उरलेल्या असतात काही चांगल्या काही वाईट	फक्त आठवणी मनात उरलेल्या चांगल्या वाईट
5	सगळंच कसं टोचत आठवणी पण काटे पण	सगळंच कसं टोचत आठवणी काटे
6	साऱ्या फितूर होऊनि आठवणी आज ओवीवळ पाझरती	साऱ्या फितूर होऊनि आठवणी ओवीवळ पाझरती
7	राहील्या त्या आठवणी	राहील्या आठवणी
8	गेले ते दिवस आणि राहिल्या त्या फक्त आणि फक्त आठवणी	गेले दिवस राहिल्या फक्त फक्त आठवणी
9	दानोळी ज़िल्हा परिषद स्वाभिमानी विजयी	दानोळी ज़िल्हा परिषद स्वाभिमानी विजयी
10	भारतरत्न डॉ बाबासाहेब आंबेडकर यांच्या व्या महापरिनिर्वाण दिनानिमित्त अभिवादन	भारतरत्न डॉ बाबासाहेब आंबेडकर महापरिनिर्वाण दिनानिमित्त अभिवादन

Fig. 4 Preprocessing of data—stopword removal

3.2.4 Stemming

It is a process of breaking down a word which is in its inflected form to the root word. In Marathi language, for stemming प्रत्यय: suffixes (prefixes and postfixes) are removed along with शब्दयोगी अव्यय: prepositions to obtain the stem word which is nothing but the samanya roop (सामान्य रुप) as output. Consider the example of stemming explained in Fig. 5 for the following examples:

मांजरीने दूध प्यायले· त्याच्या आठवणीत मी रात्रंदिवस रडायचे. माझ्या आनंदाला माझे यश · हे कारण आहे

3.3 *Processing of Data*

Now, we are ready with clean preprocessed data for further implementation. In this stage, the dataset would be split into two categories, i.e. training and testing dataset. The training dataset comprises 80% of the original data and remaining 20% is the testing dataset. The dataset is majorly divided into positive and negative tweets depending on the words used in the sentence. Positive emotions are further categorized into love, success, and happiness. Happiness is further divided into joy,

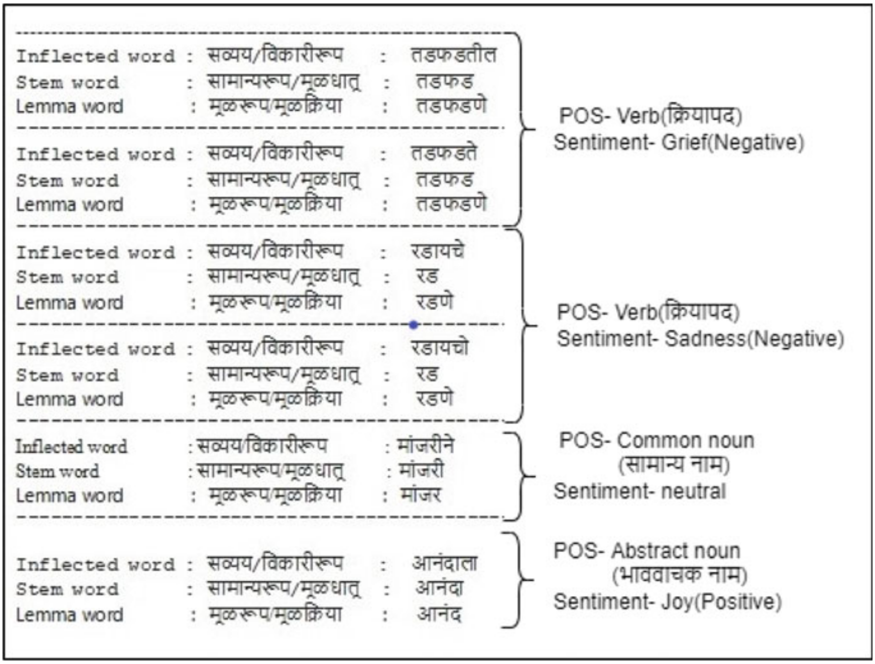


Fig. 5 Example of stemming

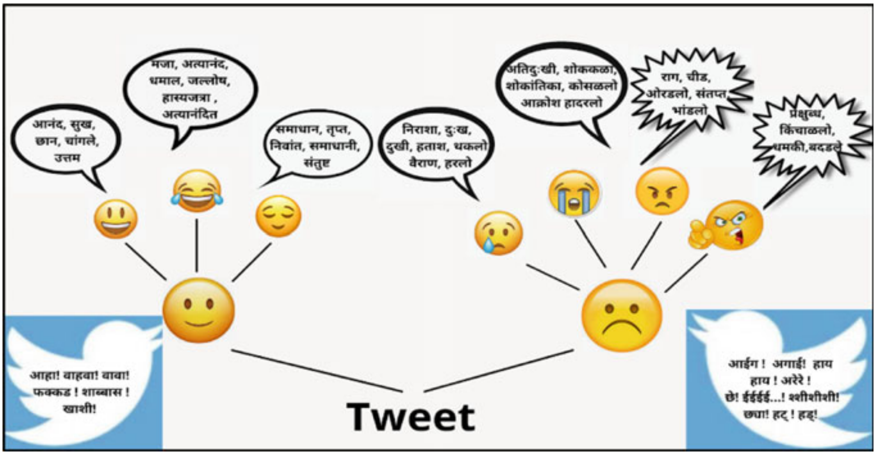


Fig. 6 Classification of sentiments

amusement, and satiety/relief. Negative emotions are categorized as anger, depression, embarrassment, fear, and disgust. Anger is categorized into resentment and outrage. On the basis of intensity of reason causing depression, later is classified into sadness and grief. Taking causes of sadness into consideration, it gets divided into guilt, anguish and loss of any form. So we would be dealing with depression sentiment in detail by studying its intensity and causes. There can be multiple reasons for depression like—loss of any form, suffering, or guilt. Figure 6 shows the detailed classification of sentiments.

3.4 Feature Selection Methods

Sentiment analysis is performed using two approaches: lexicon and machine learning. The machine learning approach involves four major steps—data preprocessing, feature extraction, feature selection, and classification. The presence of many features in a dataset might lead to poor output from the classifier algorithm. Hence to obtain accurate result and avoid overfitting, we should select the features which are most dominant in the training dataset and discard the remaining. There are three kinds of feature selection techniques-embedded, filter, and wrapper. Information gain and chi square are filter methods.

3.4.1 Information Gain (IG)

The algorithm of information gain calculates the importance of a feature based on the gain value. If this score surpasses the threshold value, then it is selected. The more the value of gain, more is its importance/influence of the particular feature on the dataset.

$$\text{Entropy } (S) = -\sum_{i=1}^{k} P(i|s)\log_2 P \tag{1}$$

$$\text{Information Gain } (S,t) = \text{Entropy } (S) - \sum_{v \in \text{values}(t)} \frac{\text{Sv}}{S}\text{Entropy (Sv)} \tag{2}$$

S—a set of all aspects,

$p(i|S)$—the fraction of the aspects in S belonging to class i. The classes are +ve, −ve, or neutral.

values(t)—refers to the set of all possible values for feature t.

3.4.2 Chi Square (χ^2)

Chi square is one of the filter methods which is used to discard the insignificant features in the dataset. It is considered as one of the most important and reliable methods in statistics. During analysing the sentiments, chi square will calculate the lack of independence between the feature (token in each tweet) and the class (positive, negative, or neutral)

$$X^2(t,c) = \frac{N^*(AD - CB)^2}{(A+C)^*(B+D)^*(A+B)^*(C+D)} \tag{3}$$

where N = total number of samples,

A = number of times ‘t’ occurs and output class ‘c’ is positive,

B = number of times ‘t’ occurs and output ‘c’ is negative,

C = number of times ‘t’ does not occur and output class ‘c’ is positive,

D = number of times ‘t’ does not occur and ‘c’ is negative.

3.5 Feature Extraction

3.5.1 TF-ISF

Term-frequency-inverse sentence frequency (TF-ISF) is a feature which is used to compute the score of each word for retrieval of data to represent how vital a certain word or phrase is to a given document.

$$\text{TF score of word } (x) = \frac{\text{Number of times } x \text{ appeared } (n)}{\text{Total words in the document } (N)} \tag{4}$$

$$\text{ISF score of word } (x) = \log\left(\frac{N}{D}\right) \tag{5}$$

3.5.2 N-gram

A N-gram consists of a continuous sequence of N elements within a set of texts. Sole keyword-based crawling, i.e. unigram approach, suffers from low precision as well as low recall. As mentioned in [10] accuracy obtained by unigram, bigram, trigram, and their combinations is showing a better result in all the four algorithms which is Naive Bayes, maximum entropy, support vector machine, and stochastic gradient descent. Hence, we would consider the combinations of the 3 n-grams for our system. One example of how ngram approach improves the accuracy of system is as follows:

Unigram: तळपायाची (noun): neutral

Bigram: तळपायाची <u>आग</u> (morpheme): anger

Trigram: <u>तळपायाची आग मस्तकात</u> (phrase): outrage.

3.5.3 POS

POS, i.e. part of speech is a programme that reads text in any language and for each word, tokens are allocated such as adjectives, adverbs, verbs, and nouns. In sentiment analysis, POS plays an important role, especially when they are adjectives (विशेषण), adverbs (क्रियाविशेषण), and interjections (केवलप्रयोगी अव्यय).

विशेषण:

Positive: नावाजलेले कौतुकास्पद हसरा, हसतमुख, Negative: उपेक्षित दुर्लक्षित हतबल, रागीट

क्रियाविशेषण:

Positive: मनःपूर्वक, मनापासून, Negative: रडत , भेकत, ओरबाडत, सुस्कारत , राबवून

केवलप्रयोगी अव्यय:

Positive: हर्षदर्शक : आहा! वाहवा! वावा!, प्रशंसादर्शक : फक्कड ! शाब्बास ! खाशी!

Negative: शोकदर्शक : आईग ! अगाई! हाय हाय ! अरेरे !, तिरस्कारदर्शक : छे! ईईईई...! श्शीशीशी! छ्या! हट् !

The graph below shows the influence of POS tagging on the output polarity of a sentence. The presence of a noun, pronoun, conjunction or preposition will not affect the sentence polarity as it does not contribute towards the overall sentiment of a sentence. Whereas a verb, adjective, adverb, or interjection will each carry a specific weightage and can increase/decrease the value of output polarity. The example given below will explain this concept.

Example: त्या घटने मुळे मी **अतिशय दुखावलो**..

The sentiment words in this example are **'अतिशय', 'दुखावलो'**.

SR NO	SENTENCE	CALCULATION	OUTPUT SCORE	OUTPUT POLARITY
1	त्या घटनेनी मी अतिशय सुखावलो.	0+0+0+(2* +1)	+2	+VE
2	त्या घटने मुळे मी अतिशय दुखावलो.	0+0+0+0+(2*-1)	-2	-VE
3	रितेश देशमुख खूप नावाजलेले अभिनेते आहेत.	2*(+1)	2	+VE
4	काही लोक खूप रागीट असतात	2*(-1)	-2	-VE
5	आहा! चहा फार छान झाला आहे.	2 +(2*+1)	4	+VE
6	शाब्बास! अशीच प्रगती करत रहा.	2 +1	3	+VE
7	मला खूप झोप येत आहे.	2*0	0	Neutral
8	त्या घटनेनी मी सुखावलो.	1*+1	+1	+VE
9	त्या घटने मुळे मी दुखावलो.	0+0+0+0+(1*-1)	-1	-VE

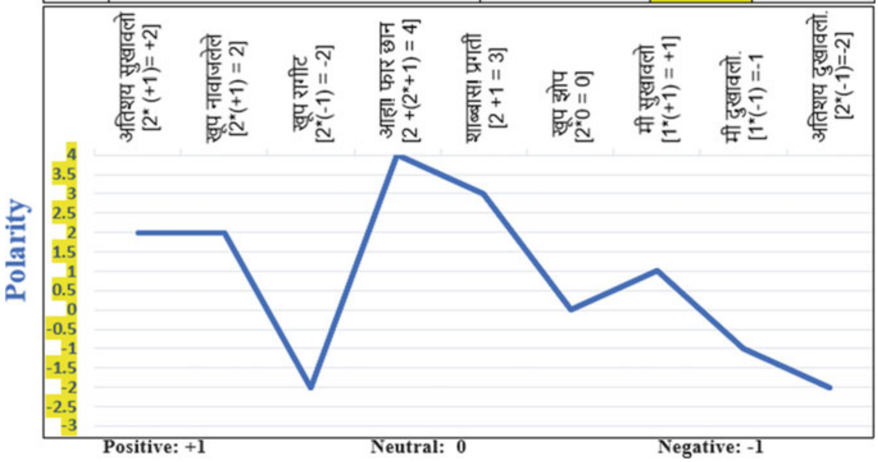


Fig. 7 POS

The words **'त्या'** , **'घटने'**, **'मुळे'**, **' मी'** are considered to be neutral with a value of **0** per word. The word '**अतिशय**' is an adverb (क्रियाविशेषण) with a assigned polarity of **2** and '**दुखावलो**' is a negative verb with assigned value of **(−1)**. Hence, the sentiment score of the sentence is found to be $\mathbf{0 + 0 + 0 + 0 + (2 * -1) = -2}$. We have considered eight sentences consisting of POS with their scores and have plotted a graph to give an overview (Fig. 7).

3.5.4 Negation Handling

Negations are words that affect the opinion orientation of other words in a sentence. Negation handling is a means of determining the scope of negation and reversing the opinion polarity of words to which negations are applied. There are three types of negations namely syntactic, diminisher, and morphological. The graphs below will indicate the change in resultant polarity of the sentence due to presence of the particular feature.

(i) **Syntactic**—नाही, उलट, नव्हते, नको, शकले नाही, केले नाही, करू नये.

Example: "मी हताश आहे.".

His sentence is classified as negative because of the term हताश, whereas "मी हताश नाही." will be classified as a positive sentence due to syntactic negation "नाही". On the other hand "मी हताश नाही असे नाही". gets classified as most negative since there are consecutive occurrences of syntactic negation twice (−ve * −ve = +ve) to the word हताश tagged as negative.

Example: माझे जीवन **त्रासदायक नाही** आहे.

In this sentence, the words "माझे जीवन" are neutral with score 0, "**त्रासदायक**" is a negative word with sentiment value -1, but due to the presence of negation term "नाही" the sentiment of the sentence is reversed and it becomes 0 + 0 + (−1)*(−1) + 0 = +1 which is positive.

We have considered eight sentences consisting of syntactic negation terms with their scores and have plotted a graph to give an overview (Fig. 8).

SR NO	SENTENCE	CALCULATION	OUTPUT SCORE	OUTPUT POLARITY
1	मी हताश नाही.	-1*-1	+1	+VE
2	कोणती आशाच उरली नाही आहे.	-1*(+1)	-1	-VE
3	मी जेवलो नाही .	-1*0	0	Neutral
4	तिकडे खुर्ची नव्हती	-1*0	0	Neutral
5	मी दुःखी नाही असे नाही.	(-1*-1)*-1	-1	-VE
6	माझे जीवन त्रासदायक नाही आहे.	-1*-1	+1	+VE
7	कोणालाही उगाचच डिवचू नये.	-1 * -1	+1	+VE
8	हा एकटेपणा मला नकोसा झाला आहे.	-1 -1	-2	-VE
9	देव अस्तित्वातच नाही आहे.	1*(-1)	-1	-VE

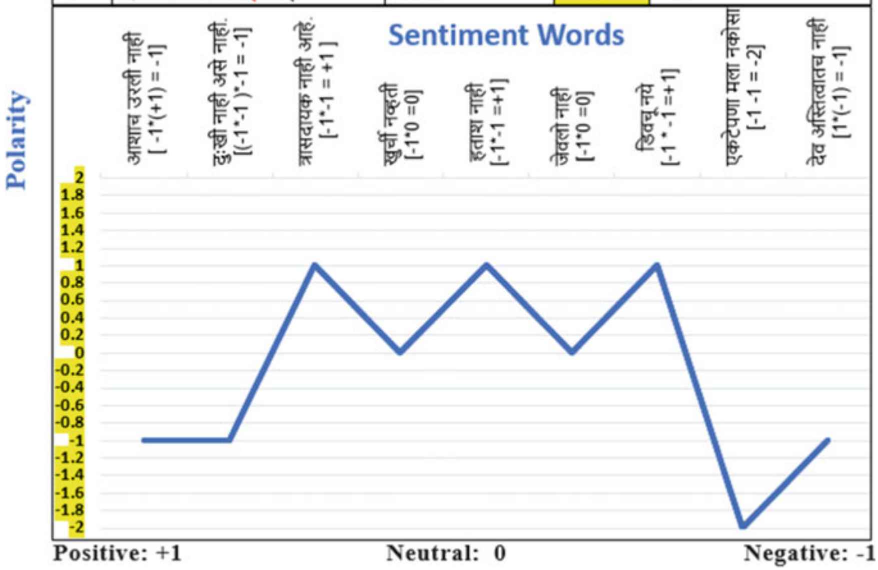


Fig. 8 Negation handling—syntactic

(ii) **Diminisher**—कमी, थोडे, थोडेसे, कधीकधी, क्वचित, कधीतरी.·

Example: "मी क्वचित आनंदी असतो."

In this sentence, "क्वचित" decreases the strength of "आनंदी" (positive) substantially. As a result, it diminishes the polarity of the sentence (positive) and rather than completely classifying it as negative, it classifies it as *least positive*.

Example: आजकाल मला **कधीकधी औदासिन्य** वाटते.·

The sentiment words which has been considered are **कधीकधी' 'औदासिन्य**. The words **आजकाल' 'मला' 'वाटते** are considered to be neutral with a value of **0** per word. The '**औदासिन्य**' is a negative word with an assigned value of **−1**. '**कधीकधी**' is the diminishing word used with an assigned value of **0.5** which reduces the polarity of the negative word. The sentiment score of the sentence is found to be **0 + 0 + 0.5*(−1) + 0 = −0.5**, and hence, output sentiment is less negative.

We have considered eight sentences consisting of Diminisher with their scores and have plotted a graph to give an overview (Fig. 9).

(iii) **Morphological**—अ, अन, अव , अप , ना, नि:, दु:, परा, कु, बे, ना.

Example: "त्याने वापरलेल्या **अप**शब्दांमुळे मला मी जगण्यास **ना**लायक वाटू लागलो".

In the above example, "शब्द" and "लायक" are root words (मूळरूप) which are *neutral and positive*, respectively. Since we combined morphological negations (prefixes) **अप** and **ना** with the root words they are now *negative ones*. And thus, the output for this tweet would be negative.

Example: मी आज **निराश** आहे.

Here, the words 'मी' 'आज' 'आहे' are neutral words with a value of 0. '**निराश**' is the sentiment word with assigned value of 1.

The sentiment score of the sentence is found to be **0 + 0 + −1** * **(+1) = −1**, and hence, output sentiment is less negative.

We have considered eight sentences with their scores and have plotted a graph to give an overview (Fig. 10).

3.5.5 Dependency Relation

When we examine the sentence "माणूस भूतकाळातल्या सुखापेक्षा दुःखामध्ये जास्त रमतो.". Here, रमतो (रमणे) represents the fondness, i.e. positive sentiment towards दुःख (negative factor). This does not mean सुख (positive factor) is not at all pleasant, but less likeable. Thus, while calculating the polarity of this sentence with respect to सुख the sentiment about दुःख has to be considered, as they are not independent (but comparative to each other: पेक्षा).

SR NO	SENTENCE	CALCULATION	OUTPUT SCORE	OUTPUT POLARITY
1	त्या घटनेनंतर मी क्वचित आनंदी असतो.	0.2*(+1)	0.2	+VE
2	आजकाल मला कधीकधी औदासिन्य वाटते	0.5*(-1)	-0.5	-VE
3	कधीकधी आठवणी डोळे झाकताच जिवंत होतात..	0.5(+1)	0.5	+VE
4	क्वचित आठवणी मनात उरलेल्या असतात, काही चांगल्या काही वाईट......	0.2*[0.4*(+1) + 0.4*(-1)]	0	Neutral
5	ती नक्कीच रडत होती.	1*(-1)	-1	-VE
6	थोडेच दिवस झालेत पण वर्ष संपल्यासारखे वाटत आहे.	0.4*0	0	Neutral
7	कोरोनमुळे मी माझ्या मित्रांना क्वचितच भेटते.	0.2*(0)	0	Neutral
8	कधीकधी हा एकांत नकोसा वाटतो.	0.5*[(1* (-1)]	-0.5	-VE

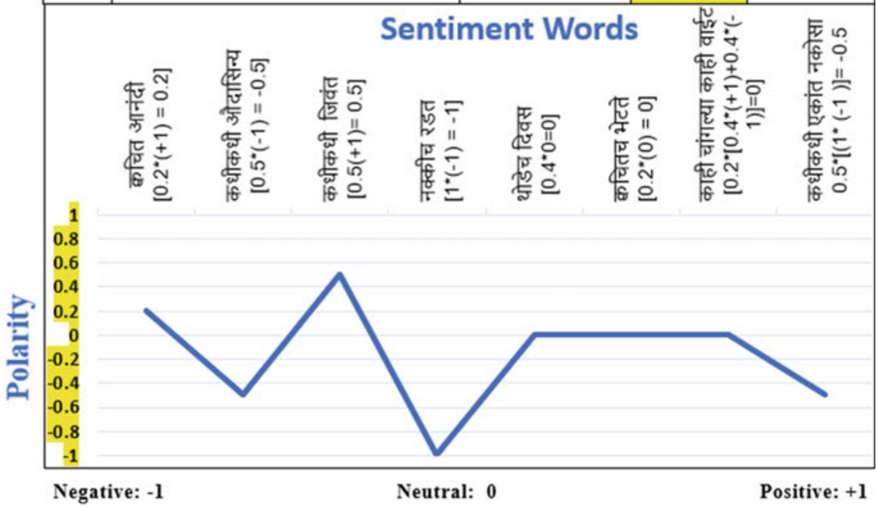


Fig. 9 Negation handling—Diminisher

3.5.6 Implicit Function

It allows the extraction of multiple features per sentence. Adjectives and adverbs are the most ordinary implicit feature indicators. A feature extraction process consists of two steps: first, identifying the part of the sentence in which the feature is implied, followed by determining the sentiment of that part of the sentence. The second step is to find the sentiment of that part of the sentence.

For example, “मी आज हवेत तरंगत आहे”

In this sentence, user is indicating joy (positive) indirectly without mentioning joy, the user is not physically flying its only conveying joy by using phrase (वाक्यप्रचार).

SR NO	SENTENCE	CALCULATION	OUTPUT SCORE	OUTPUT POLARITY
1	काही निराश होऊन विस्कळीत होतात.	0.4*(-1) -1	-1.4	-VE
2	त्या नालायकाच्या कुठे नादी लागतो.	-1*(+1)	-1	-VE
3	आता मला आरोग्याकडे दुर्लक्ष करून चालणार नाही.	[-1*(+1)]*-1	+1	+VE
4	त्याने वापरलेल्या अपशब्दांमुळे मी दुःखावले.	-1*(+1) -1	-2	-VE
5	मी आज निराश आहे.	-1*(+1)	-1	-VE
6	एवढे क्रूर दुष्कृत्य मी कुठेही पहिले नाही.	-1 -1 -1	-3	-VE
7	काही लोक बेशरम असतात.	-1	-1	-VE
8	हे लोक बेअक्कल आहेत .	-1	-1	-VE

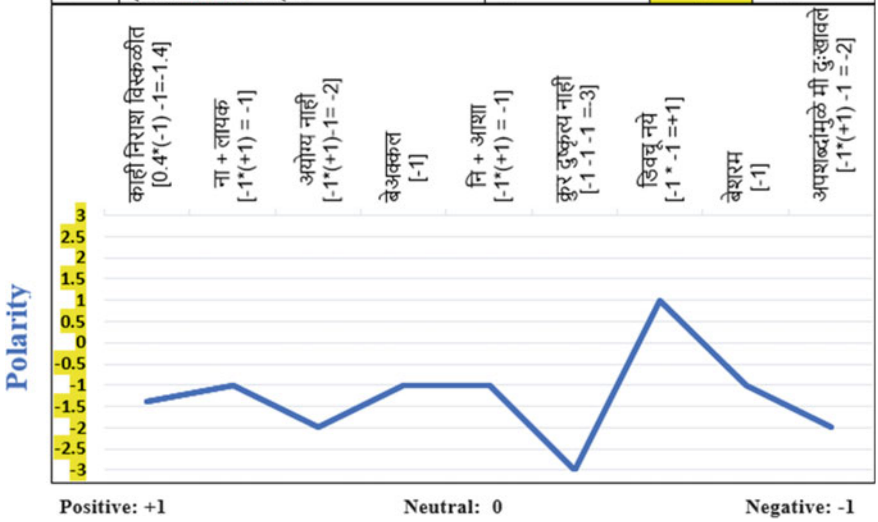


Fig. 10 Negation handling—morphological

3.6 *Methods*

3.6.1 SVM

A support vector machine (SVM) is a significant yet flexible machine learning algorithm that scrutinizes data in order to perform classification and regression. With the help of this algorithm we can achieve accuracy of only 50%.

3.6.2 K-Nearest Neighbours

The KNN algorithm assumes that things with similar characteristics are located close together.

In other words, things which are alike are adjacent to each other. To select the K that's correct for your data, we run the KNN algorithm various times with incompatible values of K and choose the K based on the number of inaccuracies we come across while carrying on the algorithm's capability to specifically make predictions when it is given new data. We can achieve accuracy of approximately 81% with the help of this algorithm.

3.6.3 Naive Bayes

Naive Bayes classifier is a classification technique that presumes independence among other mediums. It is a technique based on the assumption that a specific feature's presence in a class cannot be correlated with its presence in any other class. It provides a way of calculating following probability $P(c|x)$ from $P(c)$, $P(x)$, and $P(x|c)$.

$$P(x) = \frac{P(c)P(c)}{P(x)} \tag{6}$$

With the help of this algorithm, we can achieve maximum accuracy of 90–93%. As this is the highest accuracy achieving algorithm, we will be using it for our implementation.

4 Conclusion

Sentiment analysis is quite a popular research field yet less work has been done in the Marathi language. The addiction to social media platforms has increased due to the growing usage of the Internet in the Indian population. Studies have proven that depression among teenagers, and young adults has become more common over the past decade. Therefore, we are mainly focusing on depression in this paper. In this study, we will be using the machine learning approach to compute the opinion of the Marathi tweet displayed by the user on twitter. Till now, we have collected and generated a database for Marathi tweets and have tagged them with -1, 0, and 1 as negative, neutral, and positive. We have also manually created dictionaries for positive and negative Marathi words. Simultaneously, work is being done towards the completion of the machine learning model. This project is still under implementation. This preliminary study will help us in building a model which will not only analyse the positive, negative, and neutral sentences but will also dig deep into the negative emotion and categorize it as depressing or sad. Depression analysis of text data on social media can help us in saving many lives or at least assist people by detecting depression at an early stage.

5 Future Work

To make the outcome of this work efficient, we will have to keep making the database as rich as possible. As our inclination is more towards the depressing emotion the Marathi sentences depicting the same will be added to the dataset. The scope of this project can be further extended by detecting different emotions like joy, abuse, outrage, etc., expressed via text. Depending on the emotion extracted from text, various applications can be implemented on social media platforms.

References

1. Nithin P, Ahmed O, Hans AJ, Faraaz A, Madhusudhan MV (2021) Depression detection model based on sentiment analysis on Twitter API. Int J New Technol Res (IJNTR) 7(5):16–20. ISSN: 2454-4116
2. Kulkarni A, Mandhane M, Likhitkar M, Kshirsagar G, Joshi R (2021) L3CubeMahaSent: a Marathi tweet-based sentiment analysis dataset. In: Proceedings of the 11th workshop on computational approaches to subjectivity, sentiment and social media analysis, 19 April 2021. Association for Computational Linguistics, pp 213–220
3. Senthamilselvan K, Aneri D, Adithya C, Kani Kumar P (2020) Twitter sentiment analysis using machine learning techniques. Int J Eng Adv Technol 9(3). ISSN: 2249-8958 (Online)
4. Shah SR, Kaushik A, Sharma S, Shah J (2020) Opinion-mining on Marglish and Devanagari comments of YouTube cookery channels using parametric and non-parametric learning models
5. Arora P, Arora P (1989) Mining Twitter data for depression detection. IEEEM. Young, The technical writer's handbook. University Science, Mill Valley, CA. 978-1-5386-9436-7/19/$31.00
6. Stephen JJ, Prabu P (2019) Detecting the magnitude of depression in Twitter users using sentiment analysis. Int J Electr Comput Eng 9(4):3247
7. Deshmukh S, Patil N, Rotiwar S, Nunes J (2017) Sentiment analysis of Marathi language. JournalNX 3(6)
8. Kharde VA, Sonawane SS (2016) Sentiment analysis of Twitter data: a survey of techniques
9. Hung LP, Alfred R, Hijazi MHA, Obit JH (2015) A review on feature selection methods for sentiment analysis. Adv Sci Lett 21:2957–2962
10. Sarlan A, Nadam C, Basri S (2014) Twitter sentiment analysis. In: 2014 international conference on information technology and multimedia (ICIMU), 18–20 Nov 2014, Putrajaya, Malaysia

GMM-EM-ACO Model for Congestion Free Routing in Social Internet of Things

D. Bhavya, D. S. Vinod, S. P. Shiva Prakash, and Kiril Krinkin

Abstract The emergence of social Internet of Things (SIoT) is accelerating the growth of smart objects in recent times. The SIoT refers to the interconnection of every smart object in a given environment having some relationship. As every technology has its own challenges, SIoT poses a challenge in congestion of routing paths. The routing as many SIoT applications, namely smart city and smart home. The proposed model has two phases, namely Gaussian mixture model (GMM) and ant colony optimization (ACO) that helps to identify congestion free paths in a network. The GMM model uses expectation maximization (EM) to select the path by identifying maximum likelihood. The clusters are converged obtaining the value of +7.4427 for 10 clusters. ACO resulted in searching the path between the source and destination nodes while navigating within the network.

Keywords Gaussian distribution · Gaussian mixture model (GMM) · Social IoT · Machine learning (ML)

1 Introduction

The emergence of the social Internet of Things (SIoT) for Internet of Things (IoT) is accelerating the growth of smart objects in the current trends [1]. The inter connections of every smart object in an environment with having a relationship between each object is known as SIoT. The SIoT has 10 types of relationships, namely parental object relationship (P-OR), co-location object relationship (CLOR), owner-

D. Bhavya · D. S. Vinod · S. P. Shiva Prakash (✉)
Department of Information Science and Engineering, JSS Science and Technology University Mysuru, Mysuru, Karnataka, India
e-mail: shivasp@jssstuniv.in

D. S. Vinod
e-mail: dsvinod@jssstuniv.in

K. Krinkin
Department of Software Engineering and Computer Applications, Saint Petersburg Electrotechnical University "LETI", Saint Petersburg, Russia
e-mail: kirill@krinkin.com

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_23

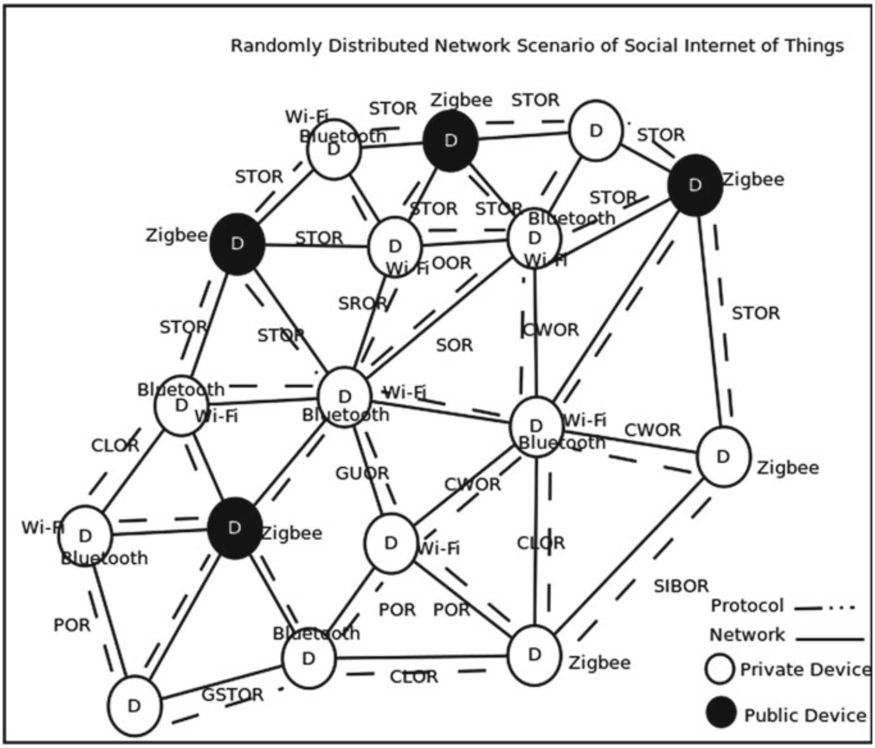


Fig. 1 Randomly distributed network scenario of social Internet of Things

ship object relationship (OOR), guardian object relationship (Gr-OR), social object relationship (SOR), guest object relationship (Gu-OR), sibling object relationship (Si-OR), stranger object relation (St-OR), service object relationship (Se-OR), and co-work object relationship (CWOR) [2–5]. These relationships are established in the SIoT network based on the various connected protocols, namely Wi-Fi, zigbee, bluetooth, Wi-Fi-Direct and BLE [6, 7]. All the relationship and protocols scenario of SIoT is shown in Fig. 1. In SIoT, the network has both static nature and the mobility nature of nodes. This mobility nature of nodes has various significant factors like identification of neighbor nodes and congestion of path in a network which effect the energy consumption of a node. In the network of SIoT, increase in the nodes is one of the key challenges. Unlike traditional networking theories, such problems can be dealt with in machine learning techniques. The machine learning techniques like analysis of distribution, clustering and predictions modeling techniques using Gaussian mixture models(GMM) and expectation and maximization (EM). The ant colony optimization (ACO) that helps to search paths in the network. The main objective is to identify congestion free paths in a network based on clustering various protocol and relationship-based path searching in a SIoT. The SIoT network consists of various protocol connections between objects. These connected objects are sharing the information in a relationship.

The congestion of the routing path is the main challenge because many pairs of objects between iterations for sharing information can retain excellence in optimizing this assignment to and better variable-value pairs and some time to run the automatic updating mechanism [8–10]. The contribution of this study is to reduce the randomness of the search. The organization of the work is as follows: Sect. 2 presents the related work, Sect. 3 describes problem statement, Sect. 4 provides system model and problem formulations, Sect. 5 on proposed model and explains algorithm, Sect. 7 on dataset, results and comparative study and Sect. 8 presents the conclusion and future of the work.

2 Related Work

The section explains the related works carried out in SIoT addressing the issues of routing congestion. Atzori et al. provided the concept called social Internet of Things (SIoT) paradigm for IoT through social relationship [1]. Lv et al. worked on 6G-enabled objects using low energy [11]. This concept increases in smart devices and creates a complex network for trending concepts that provides the information sharing or recommendation to smart devices which helps users in a service. Moni et al. worked on supporting domain-specific model-driven software engineering (MDSE) for Internet of Things (IoT) using various clustering techniques [12]. Maheshwari et al. worked on ACO to analyze alive nodes, dead nodes, energy consumption and data packets [13]. Guan et al. worked on automatic updating mechanism in ACO [14]. Farag et al. [15] worked on protocol IPv6 for heterogeneous dynamic low-power and lossy networks (LLNs). The work is carried out under Internet of Things (IoT) application using reinforcement-learning Q-learning. Natarajan et al. [16] worked on machine learning (ML)-based path construction for neighborhood sensing in telecommunication-based cognitive radio (CR) IoT applications. The work is carried out under artificial neural network (ANN) for analysis of residual energy and resource scalability in CR-IoT.

3 Problem Statement

The nodes in a SIoT network having various protocols will be connected between two nodes and the path for every communication during routing. The choice of path selection for routing is based on protocols between the nodes. These path selection criteria sometimes affects the network failure or network partition like failure of network because failure of protocol connections due to various protocols connections makes congestion in path for object which plays an important role in connection. Therefore, it requires an analysis of routing protocol based on presence of connection between the nodes. Hence, this work focuses on solving the connections using machine learning techniques.

4 System Model

The SIoT S_m is defined has a set of objects O where O=$(1, 2, \ldots, n)$ are randomly distributed in a space and which are connected in a network N_x. These objects O are connected through relationship N_R using objects profiles. There are 10 types of relationships N_R, where $R = R1, R2, \ldots, R10$. The SIoT objects establish networks for nearby objects identified based on the distance in a network using the artificial intelligence in an object. Consider the system model S_m having the set of objects in a network. The path between the objects, it will interact through the relationship as defined in the Eq. 1. In general, SIoT network 'nx' from nodes required to find the path in an environment. Hence the system model S_m can be defined as in Eq. 1.

$$S_m = \sum O\left\{N_{nx}|_R : \forall N_{(R,nx)} = f(N_{nx}, N_R)\right\} \tag{1}$$

4.1 Problem Formulation

Consider a network graph $N_{nx}(N_n$,$N_R)$ where N_n represents the number of objects, N_R represents the set of relational links. The object network N_{nx} has $(s, d, r) \in N$ where, $N = 1, 2, \ldots, n$ objects.

The objective function, let $P_{s,d}$ be the path from source s to destination d in the network using searching technique v. The objective function can be defined is shown in Eq. 2.

$$\forall N = N_{nx} \in (s, d)$$
$$\Longrightarrow \sum_{1}^{n}(f(N_{nx}.N_R)) \tag{2}$$

$\forall$ N, subjected to $N_{nx} \in G(V, E)$ and $N_R \in (s, d, v)$ where, $f(N_R)$ is connected using edges E and vertices V in routing algorithm to discover the shortest distance from object source s to object destination d using the searching technique v has constraint to establish the relationship link between the objects. Therefore, using the routing algorithm to find the N_R is to discover the congestion free path between the object in the network N_{nx}.

5 Proposed Model

In this work proposed, a model using GMM-EM-ACO, this model has two phases for suggesting the path in routing. The first phase using analyze density in the network using the GMM is shown in Eq. 3. The GMM clusters are formed using parameters

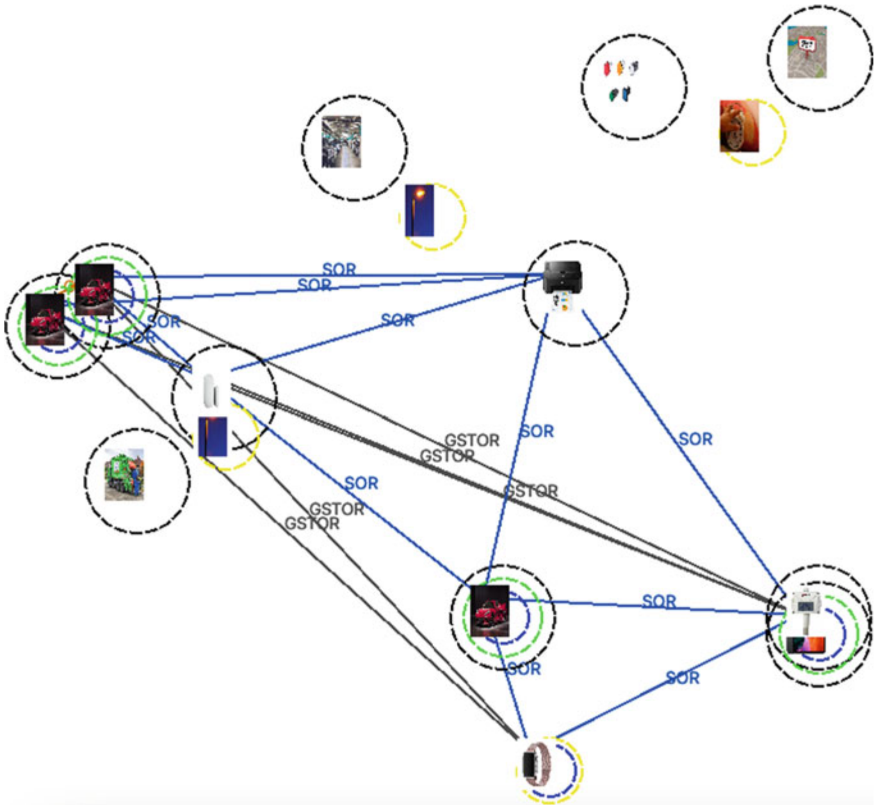


Fig. 2 Sample on congestion free path selection using relationships between the objects

such as mean and variance of each node. The probability density function identifies the covariance of the distributions. Hence the mixture of Gaussian distribution identifies the cluster using components of variance, mean of nodes and standard deviations to find the log likelihood while testing the fitness of the model. These components accesses the latent variables using Baysian inference-based EM algorithm that optimizes the clusters. In second phase, ACO technique is used for routing in the SIoT. ACO is used to search the congestion free path in the clusters. The solutions are generated by a probabilistic constructive mechanism in ACO. The ACO is associated with components edges E and vertices V in graph G(V,E). A pheromone trail value T_{ij} is associated with each component $(E, V)_{ij}$. Pheromone values allow the probability distribution of different components of the solution to be modeled during the search. The components deposit a certain amount of pheromone, and it depends on the quality of the solution found. The sample on congestion free path selection using relationships between the objects is shown in Fig. 2.

The Gaussian distribution:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{3}$$

where $f(x)$ = probability density function, σ = standard deviation, μ = mean.

Let us consider the SIoT environment of smart city the Gaussian mixture model (GMM) that generates the set of relationship in nodes N_R and set of missing relationship nodes $N_n R$ and vector of a network Nx which is having the unidirectional along with likelihood function $L(Nx; N_R, N_n R) = P(N_R, N_n R|Nx)$, the maximum likelihood estimate of the unknown nodes in the network by maximizing the marginal likelihood of the observed relationship in nodes. The expectation maximization (em) algorithm chooses the network Nx to maximize the likelihood in path which means expecting more nodes in the relationship N_R is represented as in Eqs. 4 and 5.

$$N^x = \text{argmax}_n N_R(n, N_R^x) \tag{4}$$

and maximizing

$$Nx^{x+1} = \text{argmax}_{Nx} N_R(n^x, N_R). \tag{5}$$

The likelihood calculations are based on the mean and variance of the Gaussian distribution for given observations N_R that produces the cluster K with mixture components to the marginal probability distributions represented in Eq. 6.

$$P(N_R = n) = \sum_{k=1}^{K} \Pi_k P(N_R = n|Z_i = k) \tag{6}$$

where $Z_i \epsilon (1, \ldots, K)$ is the latent variable representing the mixture component for N_R, $P(X_i|Z_i)$, and π_k is the mixture proportion representing the probability that N_R belongs to the k^{th} mixture component. The EM algorithm update the cluster using Pi, μ, and $\sum$ values is represented in Eqs. 7–9.

$$\Pi = \frac{N_k}{T_k} \tag{7}$$

$$\mu = \frac{1}{N_k} \sum_i r_{ik} N_R \tag{8}$$

$$\sum = \frac{1}{N_k} \sum_i r_{ik} N_R - {\mu_k}^T (N_R - \mu_k) \tag{9}$$

where N_k is the number of nodes in a cluster and T_k is the total number of nodes in a network.

6 Algorithm

In proposed GMM-EM-ACO algorithm initially, identify the relationship-based clusters based on parameter. It updates the cluster using components of parameters. Next in ACO select, the shortest edge with a large amount of pheromone to build a solution in probability using a local pheromone and update rules at end of each iteration update the trials on applying a modified global pheromone to updating rule.

Algorithm: GMM-EM-ACO

```
While Objects in Network do
        identifyRelations()
        clusterConnections()
        Parameterupdate()

    while clusters do
        selectRelations()
        generateSolutions()
        daemonActions()
        pheromoneUpdate()
    repeat

 repeat
```

7 Results

This section explains the results obtained through the GMM. The GMM gives the clusters based on the relationships between the objects, namely STOR, CLOR, SIBOR, GUOR, CWOR, GSTOR, SOR, OOR, SROR and GUOR. These cluster relationships are based on private objects and public devices profiles such as device-ID, ID-user, device-type, device-brand and device-models. The GMM-EM obtains the log likelihood of positive values +7.4427 for 10 clusters based on relationships which is shown in Fig. 3. The obtained cluster are identified the congestion free path between the objects, using ACO search the path for a fixed amount of time. The ACO redouble effort after the new best path, the distance affects pheromones, all relative pheromone reward based on finding a new best path is shown in Fig. 4.

This comparative study of various routing techniques using machine learning algorithms is given in Table 1. Moussa et al. proposed ANN-ACO MPPT (Maximum Power point tracking) based on an ant colony optimization (ACO) to develop ANN for optimal values of power quality improvement and distortion free signals [17]. Rathor et al. worked on predicting location in Geographical Information System

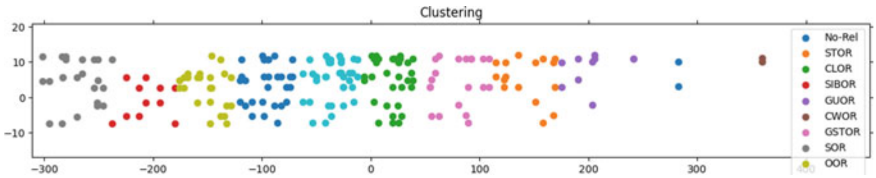


Fig. 3 GMM-based clusters

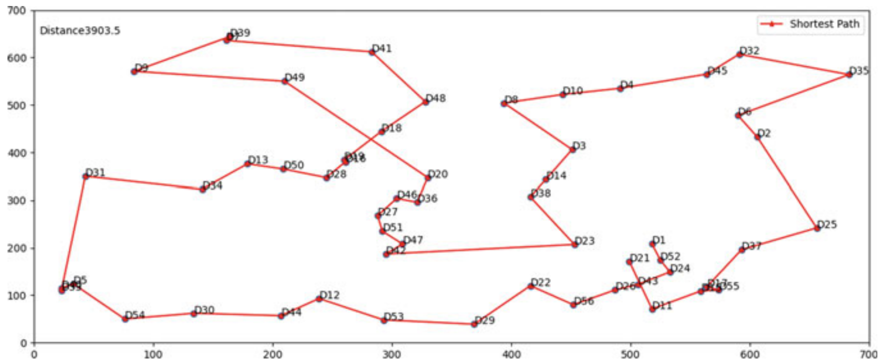


Fig. 4 ACO-based path selection

Table 1 Comparative study

Dataset	Technology	Scope	Relationship
Power grid data [17]	Maximum power point tracking using ACO	Control system	No relationship
GIS data [18]	Predict using ACO-ANN	IoT	No relationship
Smart city data (proposed work)	Prediction using GMM-ACO	IoT	Relationship

(GIS) data to identify quality of service constraints using artificial neural network (ANN) [18].

8 Conclusion and Future Work

This section explain the work is based on Gaussian mixture model (GMM) EM obtain the log likelihood of positive values +7.4427 for 10 clusters and ACO ant colony optimization used to solve congestion free paths in a network. The limitation of the work is based on the relationship it fix the path. In the future, work on the real-time big data to analyze the SIoT environment using deep learning models.

Acknowledgements This work was carried out under the "Development program of ETU "LETI" within the framework of the program of strategic academic leadership" Priority-2030" No. 075-15-2021-1318 on 29 Sept 2021.

References

1. Atzori L, Iera A, Morabito G (2011) SIoT: giving a social structure to the internet of things. IEEE Commun Lett 15(11):1193–1195
2. Khelloufi A et al (2020) A social relationships based service recommendation system for SIoT devices. IEEE Internet Things J
3. Marche C et al (2020) How to exploit the social internet of things: query generation model and device profiles' dataset. Comput Netw 174:107248
4. Ali DH (2015) A social Internet of Things application architecture: applying semantic web technologies for achieving interoperability and automation between the cyber, physical and social worlds. Diss., Institut National des Télécommunications
5. Nitti M, Atzori L, Cvijikj IP (2014) Friendship selection in the social internet of things: challenges and possible strategies. IEEE Internet Things J 2(3):240–247
6. Marche C, Atzori L, Nitti M (2018) A dataset for performance analysis of the social internet of things. In: 2018 IEEE 29th annual international symposium on personal, indoor and mobile radio communications (PIMRC). IEEE, p 2018
7. Phuttharak J, Loke SW (2018) A review of mobile crowdsourcing architectures and challenges: toward crowd-empowered internet-of-things. IEEE Access 7:304–324
8. Bhavya D, Vinod DS, Shiva PS A comparative analysis of machine learning algorithms for healthcare device data of social IoT. In: Conference of open innovations association, FRUCT No. 25. FRUCT Oy
9. Varsha V, Shiva PS, Krinkin K (2019) Energy aware handover in LBS interference for5G dense heterogeneous network. In: Conference of open innovations association, FRUCT No. 25. FRUCT Oy
10. Shiva Prakash SP. Power aware routing in dynamic IEEE 802 11s wireless mesh networks
11. Lv Z, Lou R, Li J, Singh AK, Song H (2021) Big data analytics for 6G-enabled massive internet of things. IEEE Internet Things J 8(7):5350–5359
12. Moin A, Badii A, Günnemann S (2021) Enabling un-/semi-supervised machine learning for MDSE of the real-world CPS/IoT applications. arXiv preprint arXiv:2107.02690
13. Maheshwari P, Sharma AK, Verma K (2021) Energy efficient cluster based routing protocol for WSN using butterfly optimization algorithm and ant colony optimization. Ad Hoc Netw 110:102317
14. Guan B, Zhao Y, Li Y (2021) An improved ant colony optimization with an automatic updating mechanism for constraint satisfaction problems. Expert Syst Appl 164:114021
15. Farag H, Stefanovic C (2021) Congestion-aware routing in dynamic IoT networks: a reinforcement learning approach. arXiv preprint arXiv:2105.09678
16. Natarajan Y, Srihari K, Dhiman G, Chandragandhi S, Gheisari M, Liu Y, Lee CC, Singh KK, Yadav K, Alharbi HF (2021) An IoT and machine learning-based routing protocol for reconfigurable engineering application. IET Commun
17. Moussa N, Nurellari E, El Belrhiti El Alaoui A (2021) A novel energy-efficient and reliable ACO-based routing protocol for WSN-enabled forest fires detection
18. Rathod N, Sundaresan R (2022) Relay placement algorithms for IoT connectivity and coverage in an outdoor heterogeneous propagation environment. IEEE Access 10:13270–13289
19. Yao L et al (2016) Things of interest recommendation by leveraging heterogeneous relations in the internet of things. ACM Trans Internet Technol (TOIT) 16(2):1–25

Natural Language Processing

Gesture Recognition for American Sign Language Using Pytorch and Convolutional Neural Network

Devashsih Sethia, Pallavi Singh, and B. Mohapatra

Abstract Human–computer interaction (HCI) is the most prevalent topic of active research due to the demand for machine learning and computer vision. American Sign Language (ASL) is one of the most popular languages used by deaf and dumb people in the world. The deaf and dumb people use hand gestures to communicate. Hand gestures vary from person to person in shape, size, scale, and image quality. Hence, nonlinearity exists in this problem. In the area of image processing, there has been tremendous progress made recently, and it's proven that neural networks have numerous applications in interpreting sign language. The recognition of ASL in real-time motion is employed using an efficient artificial intelligence tool, and Convolutional Neural Network (CNN) has been proposed in this work. The dataset of 27,455 images of 25 English alphabets has been used to train and validate our model. The model is tested on 7172 images which were divided into many classes. The maximum validation accuracy of the model with enhanced data was found to be 99.8% which is better than many existing methods in real-time motion.

Keywords Sign language · CNN · Gesture recognition · Pytorch

1 Introduction

ASL (American Sign Language) is the most widely used sign language in the USA. Since deaf and dumb people's main limitation is communication, and they are unable to communicate via spoken languages. Sign language is their only means of communication. Deaf and dumb persons use their hands to express various gestures in order to communicate with others. Gestures are non-verbally communicated messages that are recognized by eyesight. Sign language is the nonverbal communication of the deaf and dumb. There are a variety of sign languages, including British, Indian, and American. Users of ASL may struggle to understand British sign language (BSL), and vice versa.

D. Sethia · P. Singh · B. Mohapatra (✉)
Department of Electrical, Electronics and Communication Engineering, Galgotias University, Gr. Noida 201310, India
e-mail: writetobm@gmail.com

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_24

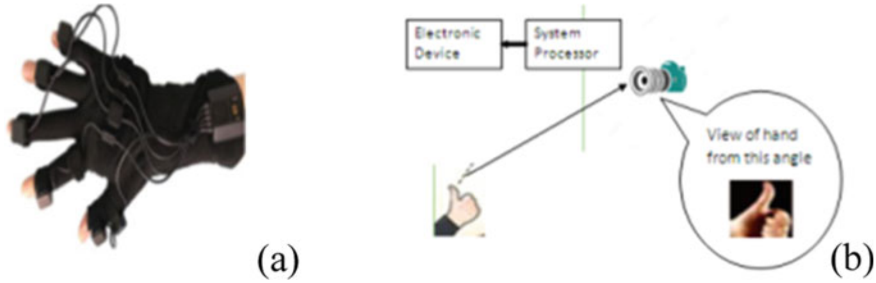


Fig. 1 **a** Sensor-based technique and **b** vision-based techniques

Numerous methods for evaluating gestures/sign language have been developed. First of all, there is a technique that requires users to wear sensors (input devices) such as gloves and hand straps shown in Fig. 1a. These strategies offer the advantage of not being distracted by various backgrounds when identifying motions. But they come at the cost of lack of natural resources, size, and expense. Secondly, camera-vision strategies use imaging tools such as cameras or kinetic sensors to capture data based on the perception of how a person perceives in their environment shown in Fig. 1b. The placement of cameras, hand visibility and the way it is segregated in the image, the effectiveness of the extricate characteristics, and the classification methods all contribute to the efficiency of these systems [1].

The biggest problem is finding a way to train a computer to detect the motion of hands. The adjustment of the fingers and the form of the hands vary according to the motion of the hands. As a result, uncertainty is one of the aspects of hand gestures that should be considered. The information of the data conveyed by the images can be used to accomplish the uncertainty in hand gestures. This method involves combining two tasks: feature extraction and classification. Traditional image processing algorithms-based hand gesture recognition was not generally used in HCI due to its limited low accuracy and complexity of the algorithm. An account of development in gesture detection established on machine learning has recently progressed quickly in HCI [2, 3].

Traditional pattern recognition techniques are unable to analyze intrinsic information of the data in its original form [4]. Subsequently, it results in extracting features from original configuration which consumes time, and it is not automated. For classification, CNN's are used which are a type of deep neural network. Deep neural networks extricate features on the go [5] and employ completely linked layers for classification by using CNN. Furthermore, CNN comprehends the complicacy of the visual images and uncertainty between the images. Therefore, using a CNN-based feed forward network approach is used to solve the problem. This is the most well-known method of detection using several neural networks.

Despite the fact that the hand motions are static, they are recognized in real time in this proposed model. However, the extraction procedure is time-consuming, and it is unlikely that all available features will be extracted. Since, the neural network is highly recognized, the most infamous method which is CNN is useful to extract features out of organized information. Thus, there is a retraction toward automated feature extraction and neural network, leading to deep neural network or CNN, emerging as a solution.

The following sections of the paper are described as follows: Sect. 2 presents literature survey and related work, proposed methodology has been presented in Sect. 3. Section 4 describes results and discussion, whereas conclusion and future works have been discussed in Sect. 5.

2 Literature Review

Several significant studies in the domain of hand gesture recognition are discussed as it has been a burning topic of numerous studies. An ANN-based gesture detection system has been developed [6]. In this research work, the images were classified in this manner depending on skin tones. Changes in pixel through cross-sections, border, and scalar descriptions like aspect ratio and edge ratio were chosen as ANN features. After those feature vectors were established, they were sent to the ANN for training. The author claimed accuracy of the system estimated to be approximately 98%. A statistical technique for detecting gestures based on Haar-like traits has been proposed [7]. The AdaBoost technique was utilized to boost the classifier in this system. A model for real-time hand gesture identification was presented [8]. These gestures are collected using a commercial surface sensor called the Myo wristband, which sends data to the computer through Bluetooth. Signal acquisition, preprocessing, feature extraction, classification, and postprocessing are the five stages of the proposed model. They employed the KNN rule in conjunction with the dynamic time warping technique for the classification stage. The claimed model's accuracy was 86%. A novel approach to identify hand gestures in a complex environment based on Single-Shot Multibox Detector (SSD) which is a deep learning algorithm which uses a neural network with 19 layers [9]. The image pyramid method is used to recognize gestures in this system. To recognize both large and small hand gestures, the algorithm crops the image into blocks. The model is tested using the hand gestures of four characters in three different complex backdrops. The system's response time is below 20 ms, indicating excellent real-time performance. The least accuracy is over 93.8%, with a high accuracy of 99.2%. The mechanism, however, is only regulated for four characters: "w", "o", "r" and "k".

A 3D CNN-based method for recognizing hand gestures has been proposed [10]. The identification in this method was challenging as capturing the gestures images with different depth and a light intensity skewed the results. They acquired 77.5% accuracy on the VIVA challenge dataset by utilizing a data augmentation approach. An approach to recognize gestures using CNN that is resilient under five invariants has been developed by Flores et al. [11]. These invariants include illumination, noise, scale, rotation, background, and translation. The dataset used was Sign Language of Peru (LSP). Architecture of two CNN models was constructed which is related to static signs and achieved 95.37% accuracy for the first CNN and 96.20% accuracy for the second CNN. Two unique CNN models capable of recognizing 24 static ASL signs were developed. They separate the ASL signs into two datasets. One into color images and the other into a combination of color and depth images. The author claimed an accuracy for the two datasets which is 86.52% and 85.88%, respectively. The results of these two sets of datasets were compared utilizing transfer learning with pre-trained models like VGG-19, VGG 16, and others [12]. To recognize sign alphabets, a system was suggested that used both deep learning models as well as image processing techniques. Using YCbCr, the images are segmented. The characteristics from handcrafted methods and a deep learning network termed VGG-19 are combined. They are delivered to the Support Vector Machine classifier to classify the signs using the serial fusion method. The accuracy achieved by the proposed system was roughly claimed to be 98.44% [13].

3 Proposed System and Methodology

In this approach, the majority of this assignment's usage has been handled through transfer learning; however, our network has been created from scratch. To analyze the image and train the model, our model uses CNN architecture. There are many CNN layers which are accompanied further by a pooling layer, ReLU layer, batch normalization layer, and dropout layer in the architecture. The architecture is composed of the following layers: Input, two convolution 2D layer, two batch normalization layer, two activation function (ReLU), two pooling 2D layer, two dropout layer which is further followed by two fully connected layers, an activation function (ReLU) and a LogSoftmax layer as output.

The collection of dataset and CNN which were employed with the feed forward network are described in this section. Figure 2 depicts the conventional method of the process flowchart. The method entails gathering data, initializing the preprocessing method, the CNN with feed forward network is composed, construction of model is done by training and tuning the model, and later by testing it.

Step-1: Data Collection

The gesture recognition technique contains three main components. According to the previous [7] research, the gesture data collection module, the feature extraction module, and the gesture classification module are used. The gesture classification module employs an appropriate classifier to classify distinct movements. For additional step recognition, researchers propose definite characteristics extrication methodology, for instance, range and inclination from the hand's terminus [14] and

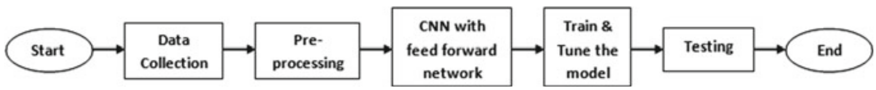


Fig. 2 Conventional flowchart process

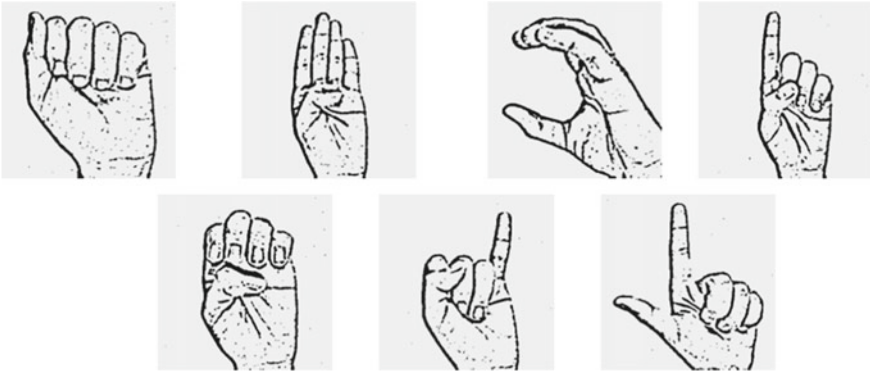


Fig. 3 Sample dataset

orientation histogram [14]. Furthermore, we've chosen 25 static gestures as the recognizable data. The dataset format is based on the famous MNIST dataset, which is a collection of English letters (A–Z). For each alphabet A–Z, each training and test case represents a label (0–25) as a one-to-one map. For instance, there are no test cases for the 9th letter which is J and the 25th letter which is Z. This is because of hand movements made while doing this gesture. There are 24 alphabet signs that have been taken for the process. As the alphabet J and Z requires movement, it becomes hard for the system to recognize that in real time. Therefore, we have 24 alphabets as data and one empty set sample of which is shown in Fig. 3. Hence, there are 27,455 images utilized for training the model and 7172 images utilized for testing.

To gather the images of the hand gestures, a computer camera was employed. The sign language gestures were made in front of the computer camera in an effort to train and validate the model. The input images are believed to contain exactly one hand, with the palm facing the camera. However, the identification procedure of the images would be more effective and less complex, if the surrounding of the image is not complicated and the contrast is high. Therefore, simple backgrounds of the images and more uniformity were assumed to be considered.

Step-2: Preprocessing

A basic preprocessing method was conducted with the collected data. This was done aiming to achieve better efficiency and less computational complexity. Subsequently, the frame size is initially set to 20 × 20 mm. The backdrop of the images is then eliminated using OpenCV's background removal method [15, 16]. Background subtraction aids in obtaining relatively quick and preliminary identifications of the items in the video stream, which can then be handled more delicately. After removal of the background, as the original image is preprocessed only the remnant of the hand is left. Furthermore, the original image is converted into the monochromatic image as it is gray scaled. It is easier for CNN to learn and it takes less time as well [17–20]. The image is also modified to fit within the desired image area. The photographs were then downsized to fit into a 20 × 20 frame and submitted to CNN. The image is thresholded after it has been reshaped and resized. The assignment of pixel values in respect to the threshold value is shown in the image. Furthermore, each pixel value is compared to the threshold value while thresholding shown in Fig. 4.

Step-3: CNN with feed forward network

CNN is particularly effective in the image identification process. As a result, it's also suitable for real-time processing. The CNN has three dimensions: width, height, and depth. A CNN is a multi-layer neural network in which each layer transfers activation amounts to another layer via a function. Convolution2D Layer, Rectified Linear Unit (ReLU) Layer, and MaxPooling2DLayer are used to extract features from data before it is transformed to the fully connected layer, LogSoftmax layer, and classification output layer. It is classified in Fig. 5.

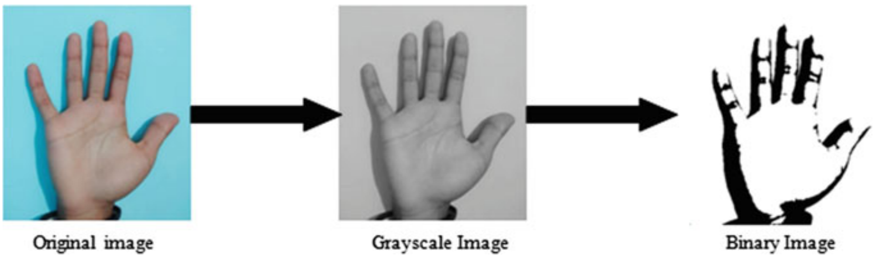


Fig. 4 Steps of preprocessing

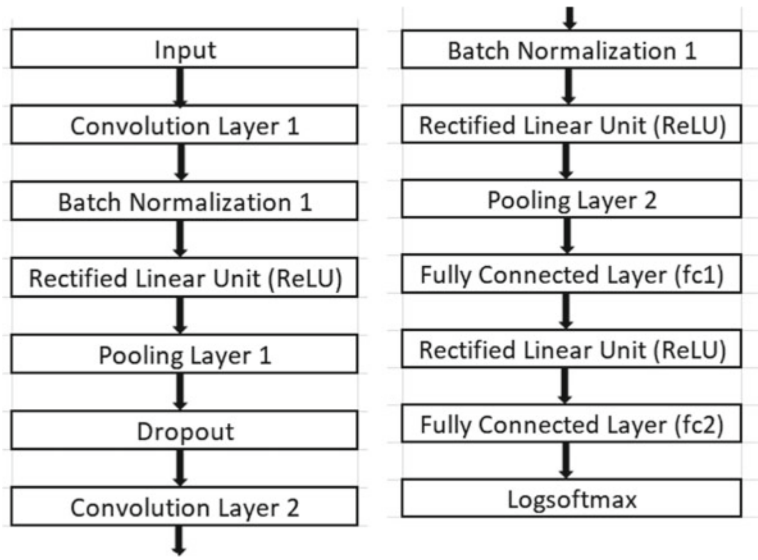


Fig. 5 Block diagram of the system

The two-dimensional Convolution2D Layer is used as the activation function. The kernel size of this layer is 5×5. It is necessary to give the input size and the stride has been set to default value. The input shape of the layer is $80 \times 80 \times 1$. Thus, hinting that a monochromatic image of 80×80 pixels will be sent to the CNN network. The next layer is batch normalization 2D layer which is used for accelerating deep network training by reducing internal covariate shift. It's a technique for standardizing network inputs that can be applied to the activations of a previous layer or directly to the inputs. For the uncertainty issue, ReLU was introduced [15]. This is an activation function which surpasses all other functions. Thereafter, a pooling two-dimensional layer (MaxPooling2DLayer) with a 2×2 pool size that takes the maximum value was added. It just selects the utmost significant information; this layer assists the network's understanding of the images which is followed by dropout layer as it prevents overfitting.

The next layer is another convolution 2D layer followed by batch normalization 2D layer. ReLU which is an activation function layer comes next. Finally, it goes to the max pooling two-dimensional layer with the pool size of 2×2 and so as to stop from overfitting, a dropout layer is employed. Furthermore, the next layer comprises a feed forward network which is fully connected having several nodes. ReLU is used as the activation function in this layer. Moreover, the linear module included creates a single layer feed forward network with 1280 input features and 250 output. This linear layer is capable of learning an average rate of correlation between the output and the input. Another feedforward linear layer with 250 input and 25 output is used.

The output layer uses one dimensional LogSoftmax function. The output of this layer is a logarithmic probability. In addition, on utilizing Pytorch open source machine learning library, this system is composed. Eventually, to retain the trail of the analysis procedure using the proposed CNN configured model the criterion of deprivation and precision were described.

Step-4: Train and Tune the model

A properly constructed CNN with correct training code should be able to memorize the answers to a set of images data. So, in order to train the neural network model, OpenCV is used for compilation and Pytorch as an open source machine learning library is employed. Also, one dimension LogSoftmax is used which is a software based activation function.

To train the model, some hyper-parameters were included in order to tune the model. Therefore, some transformations were done like re-sizing, scaling up, and down the image, snipping, breadth, and peak shifting on the dataset. The listed hyper-parameters were tuned. After training and tuning the model, testing of the system is required. Our model was trained to achieve good performance, mentioned in Sect. 4. The model is trained using the expanded dataset and tested as well. Furthermore, to increase the number of data, data enhancement is done by applying flip, zoom, shear, etc. Figure 6 shows the effect of enhanced data, as it depicts the flipping of a sign language. Data enhancement is mainly a technique to increase the number of variables where those variables are the data.

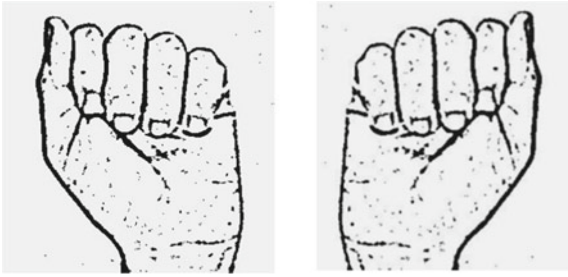


Fig. 6 Flipped images of same gestures

4 Results

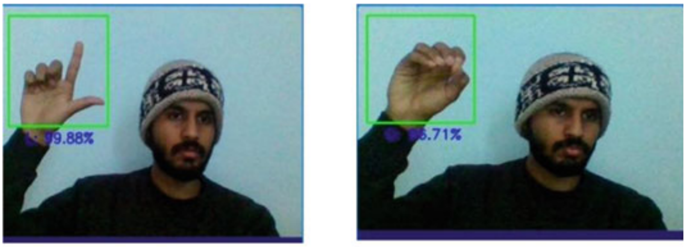
Python was utilized as the programming language to implement the system. Customarily, code was written and run by utilizing OpenCV which is a computer vision library. CNN classifier has been created using Pytorch, an open-source machine learning library. For array operations, NumPy was employed. Imutils is used for doing simple image processing tasks including translation, rotation, and scaling. We used a CNN with a feed forward network to experiment with recognition of ASL in this study. We acquired, labeled, and trained the dataset used. Furthermore, the intended work was to implement hand motions in real time on low-power computing systems.

The American Sign Language detection system was tested on characters "A", "B", "C", "D", "E", "F", "G", "H", "I", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "X" and "Y" which showed good performance. The results are given in Table 1. The model achieved a minimum accuracy of 96.71% for letter "O" and maximum accuracy of 99.88% for letter "L" in real-time motion. The results of letters "l" and "O" are shown in Fig. 7. In both phases, when we trained and tested the model for ASL recognition indicated that the second phase of the model provides higher accuracy then all other well-known algorithms. The enhancement of the expanded dataset of training samples is large which is 27,455 total samples. The Sign languages that show markable results achieved accuracy of above 99%. The letters achieving the high level predictions were characters "Y", "P", "W", "U", "D", "R", "V", "X", "G", "B" and "C". Meanwhile, there are letters with accuracy of above 98% and below 99% and some letters that have accuracy of below 98% and above 97% as given in Table 1.

The training method of the model has shown that the time taken in the training phase of the algorithm was average compared to other sturdy algorithms. The shortest recorded time for the observation was 200 ms in real-time motion which is a remarkable result. The mean classification time was about, although, in real-time the accuracy did differ but when the hand is stable the output is more accurate and precise. The bounding box is 20×20 mm which is small but it eliminates all the disturbances and focuses on the hand gestures. Sample of results has been shown in Fig. 8.

Table 1 Representing accuracy of all characters

Letter	Accuracy (%)	Letter	Accuracy (%)	Letter	Accuracy (%)
A	98.45	I	98.96	R	99.30
B	99.01	K	97.63	S	98.51
C	99.00	L	99.88	T	97.41
D	99.34	M	98.01	U	99.37
E	99.36	N	98.47	V	99.16
F	98.77	O	96.71	W	99.57
G	99.07	P	99.60	X	99.13
H	98.21	Q	97.09	Y	99.62



(a) (b)

Fig. 7 **a** Maximum accuracy of letter "L" and **b** minimum accuracy of letter "O"

5 Conclusions and Future Work

We used both traditional image processing techniques and a deep learning model to recognize sign alphabets in this suggested system. We employed both handcrafted features like ASL and a deep learning network called CNN in this case. We conclude that CNN is a well-known component based procedure, and real-time gesture recognition has a substantial impact on deep learning as a result of our research. The proposed approach achieved an accuracy of 99.88%. As a result, more precise gesture recognition is possible. This system is limited to detect sign languages and the 24 alphabets. To enhance this, gestures of numerous signs shall be included. Sign language identification using both hands is also unachievable with this system. Hence, future work could be recognition of gestures with both hands. Moreover, the recognition should not just be limited to alphabets and numbers. Rather, recognition of sentences, speech, and much more should be encouraged.

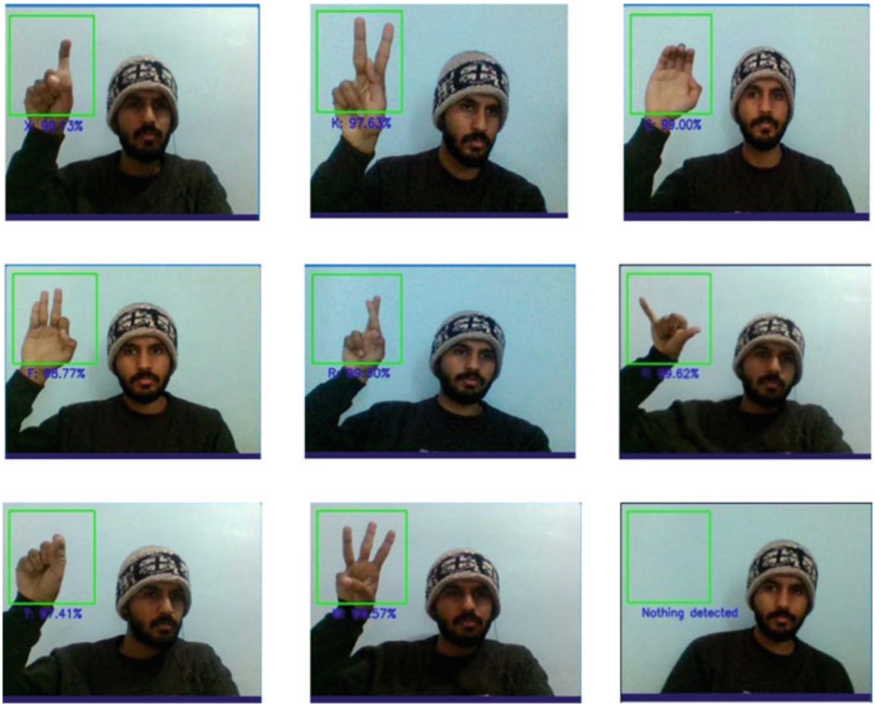


Fig. 8 Sample results

References

1. Parvathy P, Subramaniam K, Venkatesan GKDP, Karthikaikumar P, Varghese J, Jayasankar T (2020) Development of hand gesture recognition system using machine learning. J Ambient Intell Hum Comput 12:6793–6800. https://doi.org/10.1007/s12652-020-02314-2
2. Dongarra J, Gates M, Kurzak J, Luszczek P, Tsai YM (2018) Autotuning numerical dense linear algebra for batched computation with GPU hardware accelerators. Proc IEEE 106:2040–2055. https://doi.org/10.1109/JPROC.2018.2868961
3. Morchid M (2018) Parsimonious memory unit for recurrent neural networks with application to natural language processing. Neurocomputing 314:48–64. https://doi.org/10.1016/j.neucom.2018.05.081
4. LeCun Y, Bengio Y, H Ginton (2015) Deep learning. Nature 521:436. https://doi.org/10.1038/nature14539
5. LeCun Y, Haffner P, Bottou L, Bengio Y (1999) Object recognition with gradient-based learning. In: Shape, contour and grouping in computer vision, pp 319–345. https://doi.org/10.1007/3-540-46805-6_19
6. Nguyen T-N, Huynh H-H, Meunier J (2013) Static hand gesture recognition using artificial neural network. J Image Graph 1:34–38. https://doi.org/10.12720/joig.1.1.34-38
7. Chen Q, Georganas ND, Petriu EM (2008) Hand gesture recognition using Haar-like features and a stochastic context-free grammar. IEEE Trans Instrum Meas 57:1562–1571. https://doi.org/10.1109/TIM.2008.922070
8. Benalcázar ME, Jaramillo AG, Zea JA, Páez A, Andaluz VH (2017) Hand gesture recognition using machine learning and the Myo armband. In: Proceedings of the 25th European signal

processing conference (EUSIPCO), 28 Aug–2 Sept 2017, pp 1040–1044. https://doi.org/10.23919/EUSIPCO.2017.8081366
9. Liu P, Li X, Cui H, Li S, Yuan Y (2019) Hand gesture recognition based on single-shot multibox detector deep learning, 30 Dec 2019, pp 1–7. https://doi.org/10.1155/2019/3410348
10. Molchanov P, Gupta S, Kim K, Kautz J (2015) Hand gesture recognition with 3d convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 7–12 June 2015, pp 1–7. https://doi.org/10.1109/CVPRW.2015.7301342
11. Flores CJL, Cutipa AG, Enciso RL Application of convolutional neural networks for static hand gestures recognition under different invariant features. In: Proceedings of the 2017 IEEE XXIV international conference on electronics, electrical engineering and computing (INTERCON), Cusco, Peru, 15–18 Aug 2017, pp 1–4. https://doi.org/10.1109/INTERCON.2017.8079727
12. Paul P, Bhuiya M, Ullah M, Saqib MN, Mohammed N, Momen S (2019) A modern approach for sign language interpretation using convolutional neural network. In: Proceedings of the 16th Pacific rim international conference on artificial intelligence, Cuvu, Yanuca Island, Fiji, 26–30 Aug 2019, Part III, pp 431–444. https://doi.org/10.1007/978-3-030-29894-4_35
13. Rajan RG, Judith Leo M (2020) American sign language alphabets recognition using hand crafted and deep learning features. In: Proceedings of the 2020 international conference on inventive computation technologies (ICICT), 26–28 Feb 2020, pp 430–434. https://doi.org/10.1109/ICICT48043.2020.9112481
14. Mendoza-Garcia R, Landa-Hurtado L, Mamani-Macaya F, Valenzuela-Coloma H, Fuentes-Maya M (2014) Kinect-based trajectory teaching for industrial robots. In: Proceedings of the Pan-American congress of applied mechanics, Santiago, Chile, March 2014
15. Zivkovic Z (2004) Improved adaptive Gaussian mixture model for background subtraction. In: Proceedings of the 17th international conference on pattern recognition, 26 Aug 2004, pp 28–31. https://doi.org/10.1109/ICPR.2004.1333992
16. Zivkovic Z, Van Der Heijden F (2006) Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recogn Lett 773–780. https://doi.org/10.1016/j.patrec.2005.11.005
17. Grundland M, Dodgson NA (2007) Decolorize: fast, contrast enhancing, color to grayscale conversion. Pattern Recogn 2891–2896. https://doi.org/10.1016/j.patcog.2006.11.003
18. Singha J, Laskar RH (2015) Self co-articulation detection and trajectory guided recognition for dynamic hand gestures. IET Comput Vis 10:143–152. https://doi.org/10.1049/iet-cvi.2014.0432
19. Singha J, Laskar RH (2015) ANN-based hand gesture recognition using self co-articulated set of features. IETE J Res 61:597–608. https://doi.org/10.1080/03772063.2015.1054900
20. Singha J, Laskar RH (2016) Recognition of global hand gestures using self co-articulation information and classifier fusion. J Multimodal User Interfaces 10:77–93. https://doi.org/10.1007/s12193-016-0212-0

Enhanced Preprocessing Technique for Degraded Printed Marathi Characters

M. S. Sonawane, C. A. Dhawale, and C. H. Patil

Abstract Designing a dynamic and proficient OCR entity is an interesting and captivating region in image processing. The OCR organism purposes to decipher text in images to a machine comprehensible text. A successful degraded document has numerous applications like preservation of history, persistent documents, and many more. For the same, copious policies occur for numerous scripts, languages, and so far for virtuous class papers. Contrariwise, unique restricted versions have been explored for degraded printed Marathi characters. The OCR structure comprises stages like data collection, preprocessing, feature extraction, segmentation, classification, and recognition. For this work, we have developed our dataset size of 4900 images for 49 isolated Marathi characters. Among these, preprocessing phase plays a significant role, especially in the case of degraded characters. This paper focuses work on preprocessing of degraded Marathi characters. Mean Square Error, Mutual information, and Peak signal to noise ratio assessment factors are used for the evaluation of the enriched image. The proposed approach's effects are attained in MATLAB R2015a. Many researchers have made efforts for developing OCR for degraded printed documents for various languages, whereas very less attempts made for degraded Marathi character recognition. In the case of the degraded Marathi character recognition problem, the recognition accuracy is largely dependent on the preprocessing techniques and how the character information and features are retained. To overcome the limitations of earlier reported systems of loss of information and features retention, a novel and enhanced preprocessing technique is developed specifically for degraded Marathi character recognition. The proposed methodology is giving promising results as compared to other methods which are 35.14. Similarly, the average mutual information value of the proposed technique is 2.71 which is greater than others. The projected practices offered less value of average mean square error and response time. The proposed method for preprocessing will improve the recognition accuracy for degraded isolated Marathi characters.

M. S. Sonawane (✉) · C. A. Dhawale
SGBAU, Amravati (M.S.), India
e-mail: manojkumar.sonawane@rediffmail.com

C. H. Patil
School of Computer Science, Dr. Vishwanath Karad MIT World Peace University, Pune, India

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_25

Keywords OCR · Preprocessing · Degraded · Marathi · Mean square error · Mutual information · Peak signal to noise ratio

1 Introduction

India is a country having multiple languages, multiple scripts. India is devouring twenty-two languages nearby, precisely, Assamese, Bangla, Bodo, Gujarati, Dogri, Hindi, Kashmiri, Konkani, Kannada, Marathi, Manipuri, Maithili, Malayalam, Punjabi, Nepali, Oriya, Santali, Sindhi, Sanskrit, Telugu, Urdu, and Tamil. Merely twelve scripts have been utilized to transcribe such languages. Bodo, Dogri, Hindi, Konkani, Marathi, Maithili, Nepali, Sanskrit, etc., languages are written in Devanagari script [1–3]. The Sindhi language is written in Urdu script in Pakistan whereas in India it is written in Devanagari script. Bengali script is used to write the languages like Assamese, Bangla, and Manipuri. The Gurumukhi script is employed to carve the Punjabi language. Residual languages have fixed scripts. As in the case of English, lower- and upper-case letters variations are not present in Indian languages. Many languages in an India are occasioning from primordial Brahmi. Such languages are phonetic, and therefore, characters have defined contours. Apart from Urdu, all exceeding languages are written from the left to the right way [1]. Devanagari script is utilized in several Indian languages as stated above; around the world, more than 300 million population follow this script. Devanagari script is the establishment of languages in an India. Therefore, this script very important and plays a dominant role in the growth of literature, manuscripts, and writings. This script is usually recognized by a *shirorekha*, which ties the character's top in the word. However, in some words, characters may be detached. The letters covering vowels, consonants, conjuncts, etc., in this script are presently enumerated [2].

Marathi is a language of Maharashtra which is widely used, and this language is written in Devanagari [2–9]. The Marathi language is highly spoken language in the world. The Marathi language consists of 48 characters classified as 12 vowels and 36 consonants [3–5]. Handwritten Marathi character recognition is significant problem because of its application in various areas and domains like postal automation, historical document preservation, bank cheque automation, and form processing [1–5]. Handwritten Marathi character recognition is an interesting task due to inter-class and intra-class similarities.

In the present high-tech world, it is obligatory to have whole existing material in a digital form recognized through equipment. In India, wherever there is an abundance of data existing in the various formats like ancient texts, papers, books, and copies those are typically offered in printed or handwritten forms; such data are incompatible when it derives to scrutinizing information from large amount of pages. It must be digitized and rehabilitated to a written form to analyze by machineries performing examinations of a number of pages per second. It is also necessary to recuperate and dig the deprecated text. Immediately, the appropriate data about culture, history, tradition, etc., would be available to the masses. OCR has become interesting, largely focused, and rewarding uses in the field of image processing, pattern recognition, machine learning, and artificial intelligence. To differentiate printed or handwritten texts in frequently used languages like Chinese, Japanese, and English, prudently proficient also realistic OCR software are commercially offered [2–4].

OCR is the greatest obligatory portion of a document examination and study system that transforms the scanned books, text, and magazines into machine-understandable format. Document study and recognition could be divided into two parts: handwritten and printed. Handwritten documents are divided into two parts: online and offline [3, 4]. The printed documents can be furthermore distributed into two portions: good quality printed document and degraded printed document. Deprivation of the printed text can have smear inside, border smear, bleed-through, touching characters, low illuminations, unclearness, broken characters, etc. This discussion is shown in Fig. 1. The dotted portion focus on degraded printed characters. There are various errors or noise added which reduces the possibility of correct recognition.

It is stimulating to recognize printed Marathi characters in degraded documents because of noise, compound characters, high inter-variation, intra-variation between document background and text, etc. Similarly, because of the presence of the modifiers at varying positions, OCR expansion for printed Marathi degraded characters is hard. Many authors have taken exertions on degraded printed other language characters; however, no effort is observed in the Marathi language. It is feasible to inspire methodologies for degraded Marathi character recognition.

Many researchers have made efforts for developing OCR for degraded printed documents, whereas very less attempts for degraded Marathi character recognition. In the case of the degraded Marathi character recognition problem, the recognition

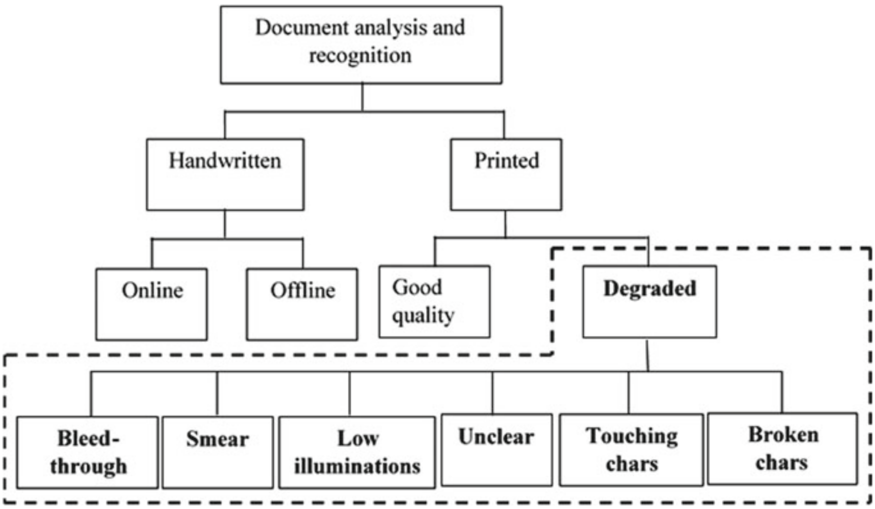


Fig. 1 Document analysis and recognition

accuracy is largely dependent on the preprocessing techniques and how the character information and features are retained. To overcome the limitations of earlier reported systems of loss of information and features retention, a novel and enhanced preprocessing technique is developed specifically for degraded Marathi character recognition.

The organization of the remaining manuscript is as follows: related work is described in Sect. 2 in short. Data collection is explained in Sect. 3. Different preprocessing practices and segmentation techniques utilized are written in Sect. 4. Three performance measures for an enriched image are explained in Sect. 5. In Sect. 6, the conversation of investigational effects is present. The concluding fragment is stated in Sect. 7.

2 Related Work

The literature is reviewed to related work, that is related to technological effect in the learning of varied text analysis and recognition methods applied to numerous languages for printed as well as handwritten scripts. The study of Indian languages offline character recognition, printed character recognition, regardless of any of the script is done and at that moment prolonged to the handwritten Marathi characters and numeral recognition in various Indian languages and scripts involving Devanagari script. The feature based on Rotation invariant texture and their trial in automated script recognition significances for the Chinese papers bare that a substantial portion of the miscalculations was because of papers mistakenly classified as Persian. The reason for it is not perfect [2]. The definite examination is also committed to touching character segmentation, handwritten compound characters, and word recognition in numerous Indian scripts. In the early 1970s, OCR's effort on the printed Devanagari script was started. Veena Bansal has prepared a comprehensive study of printed Devanagari text [10]. An organism for offline handwritten Devanagari character recognition by using direction information for feature extraction is presented by Pal et al. [11]. A summary of the various script identification tactics under each of these types is pronounced by Ghosh et al. [12]. Dhandra et al. [8] proposed an automatic technique for word-level script recognition system developed which is based on morphology and the method proposed for two printed bilingual documents of Devanagari and Kannada consisting of English numerals. The method established comprises a classifier and feature extractor. To mine separate text lines from Indian printed documents consuming curve text lines or multi-oriented text lines, the Water reservoir analogy is proficient [13]. An unexpected noise in an order might disrupt the normal state transmission for this direction, making it curious to train models. The plan used is compensating for definite negative effects of this noise. The scheme achieves a 98.88% recognition rate on handwritten numerals [14]. In the printed Gurmukhi script, different types of degradation agreed such as broken characters, heavily printed characters, touching characters, problems associated with every degradation category [15]. M. K. Mohahmed Althaf and M. Baritha Begum recommended a skillful and calm offline handwritten character recognition mechanism using radon feature extraction devising recognition 90% rate for 270 features [16]. An adaptive image contrast-dependent document image binarization system called novel document image binarization is offered by Su et al. [17]. The two-stage parameter free windowbased binarization method is proven by Chiu et al. [9]. Pixel-based evaluation procedure with an extraordinary focus on historical manuscripts, counting degradation, and complex types of font is recommended by Gatos et al. [18]. Hybrid binarization practice gives benefits of global and local thresholding. Such a procedure is designed by Sokratis et al. [19]. Patvardhan et al. [20] deliver an innovative system reliant on Wavelet Transform for document image binarization with tedious background, un-uniform enlightenment. In [21], different algorithm is described by Carlos Alexandre Barros Mello et al. that treats diversely placed shadows, front-back intervention, brightness deviation, enlightenment, aspect variants, etc. In this algorithm edge detection, morphological operations are utilized to recognize a text onto an image. Other than these, several methods for different scripts and languages are present.

3 Data Collection

Standard and benchmark dataset for degraded Marathi characters is not available till today [3–5]. For this work, the typical database was not presented. Therefore, for examination purposes, degraded printed Marathi documents are gathered from numerous places such as The Vagdevata Mandir, Dhule, Deccan College Postgraduate and Research Institute, Pune, The Bhandarkar Oriental Research Institute, Pune, Jaykar library of Savitribai Phule Pune University, and The Bharat Itihas Sanshodhak Mandal, Pune, etc. The experiment we executed included approximately 1000 diversely printed degraded Marathi documents which are degraded have been utilized. By assuming the isolated Marathi characters, work gathered 100 trials of every isolated character with diverse degradations. Total 49 Marathi printed degraded isolated characters with all having the 100 trials collected in this investigation effort. So the size of the research dataset becomes $49 \times 100 = 4900$ images. Figure 2 displays 25 sample images of isolated character 'AA'.

For every character illustration, work distinctively allocates the English names and numbers in this study.

4 Preprocessing and Segmentation

Preprocessing is demarcated as scrubbing the image document and raising it appropriately for greater feature extraction. Several practices and methods are considered under it. Normalization is the well-known method of restoring the arbitrary sized image into the usual sized image. An inter-class disparity among characters is

Fig. 2 Degraded character 'AA' dataset with 25 trials

removed using normalization. Here, an input image is normalized to 40 × 30 dimension and then transformed into a grayscale image for supplementary processing as the original image is colored one [3]. Digital images are swayed by many types of noise that stand up through data attainment exercises because of the camera's or scanner's sensor, document oldness, etc. Noise is the result of a fault in the image acquisition course that affects pixel values that do not mimic the true intensities of the actual one. Salt-pepper noise and background noise are two types of noise that are very common. The mean and standard median filters [22] are sorts of spatial digital filtering techniques, frequently engaged to sanitary salt-pepper and Gaussian noises inside the input image. In this study, adaptive Gaussian filtering is functional to the image.

Binarization is the technique by which grayscale image is rehabilitated into binary images that relies on the pixel value. This is operated to mine text from low prominence background. The pixel that produces the character wants 1 bit of data each. The rest on the pixel value black or white image is substituted by 0 or 1, respectively. The threshold-based approach is beneficial here. Thresholding is a vigorous technique in image segmentation requests. The straightforward notion of thresholding is to pick a best gray level threshold value for splitting interesting substances in an image from the background reliant on their gray level distribution [23]. In this study, all the pixels with a threshold value larger than a threshold value (128) are set to 1 else to 0 (refer Fig. 3). Sample images are shown in Figs. 4 and 5.

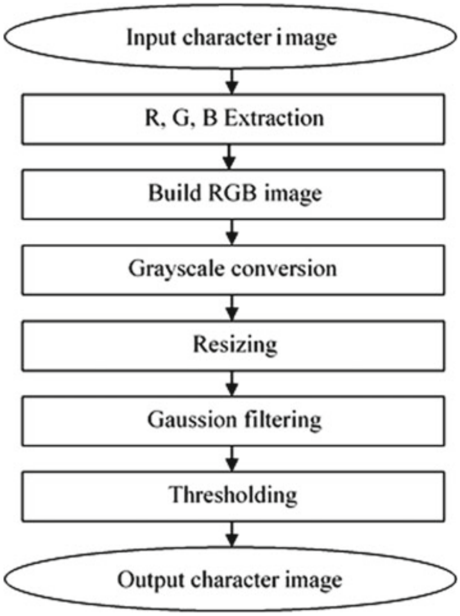


Fig. 3 Preprocessing of input character image

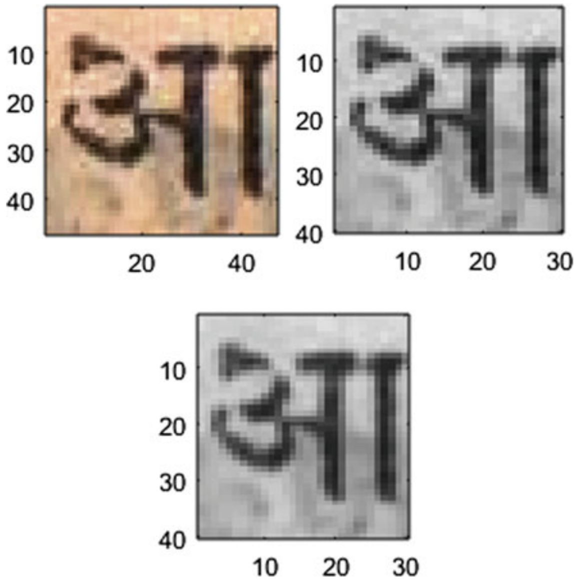


Fig. 4 Original, grayscale-resized, and Gaussian filtered images

Fig. 5 Gaussian filtered image and threshold image

5 Assessment Factors for Enhanced Image

There exist numerous assessment factors. In this work for evaluation of the enhanced image, Mean Square Error, Mutual information, and Peak signal to noise ratio constraints are exploited.

5.1 Mutual Information

Mutual information has its place in the information or probability model, which computes the extent of data that one variable contains about one more. Mutual information is the closeness degree among images. The aforementioned succeeds the volume of information in units such as Shannon's or bits acquired around one variable over another [24].

5.2 Mean Square Error

It is the cumulative squared error between 2 assumed images. A small MSE value offers reduced error. There is an antithesis relation between PSNR and MSE.

$$\mathrm{MSE} = \frac{\sum_i \sum_i (x_i - R_i)^2}{(M * N)}$$

Here, MSE—Mean Square Error, $M \times N$—Image dimension, X—Original Image, R—Renovated Image.

5.3 *Peak Signal to Noise Ratio*

This is the ratio between the maximum possible signal power and corrupted image power. Dimension and class both input and production image should be the same. Peak signal to noise ratio is employed as superiority deepness among images. The bigger peak signal noise ratio value in the modernized image gives a larger class. Peak signal to noise ratio is demarcated by Decibel unit [25]. For grayscale images, peak signal to noise ratio is termed as,

$$\mathrm{PSNR} = 10\log_{10} x - \frac{255 \times 255}{\mathrm{MSE}}$$

6 Experimental Results and Discussion

For investigational dedications, the dataset is organized by using 100 trials. This character dataset is made up of 49 characters with 100 samples of each. Hence, the total dataset size is about 4900 characters. After that, two phases of image processing, preprocessing and segmentation, are applied. Table 1 exhibits the quality investigation of the pre-processed and threshold image using the above-mentioned assessment parameters. The presentation indicates the condensed error rate and progresses the PSNR and mutual information recitals for a few Marathi character images.

Table 1 Effects of MSE, PSNR, MI

Character	MSE	PSNR	MI
A	8.6	36.13	2.72
AA	16.34	33.84	3.017
AHA	19.66	32.90	2.47
AI	5.36	36.48	2.4
AM	11.31	34.94	2.7
ANGA	12.66	34.73	2.76
AU	5.07	37.24	2.22
B	10.84	35.66	3.13
BH	8.7	36.27	3.11
CH	5.3	38.46	3.02
CHHA	22.45	32.5	2.99
D	13.69	34.86	2.78
DD	13.2	34.77	2.76
DDH	13.25	34.66	2.91
DH	10.25	35.69	2.9

Table 2 Average performance analysis with similar methods

Methods	PSNR	MSE	MI	Time (s)
Median filtering	29.88	15.73	1.98	0.98
Wiener filtering	32.39	13.77	2.31	0.93
Adaptive histogram	33.72	13.13	2.43	1.04
Proposed	35.14	12.57	2.71	0.68

Likewise, the study calculated for all characters' recitals and measured their average fallouts. Table 2 demonstrates the reasonable study among the diverse preprocessing practices in terms of mediocre PSNR, MSE, MI, and time.

7 Conclusion

This segment of study chiefly concentrated on degraded printed Marathi characters. This segment aims to offer the research impact in which the study initiated with the dares of degraded Marathi characters dataset collection. The usual dataset was not offered so arranged on its own. The course of dataset assortment and trials of all the datasets is revealed in this work. The dataset holds 4900 characters, comprising 100 trials for all 49 Marathi characters. The preprocessing approach and character segmentation using the thresholding tactic described, designed, and estimated. This exploration assumed the thresholding as the portion of preprocessing only, consequently, the pre-processed conclusion was further utilized for the feature extraction and classification phases. As observed in Table 2, the average PSNR value of the proposed methodology is 35.14, which is superior to other stated methods. Similarly, the average MI value of the proposed technique is 2.71 which is greater than others. The projected practices offered less value of average MSE and response time. The study practices the prepared dataset, so preprocessing and segmentation methods are functional only on the dataset prepared. In imminent, there can be the approach that achieves well on varied datasets. Numerous languages are standing in the world. The systems in this study are appropriate only for the Marathi language. For other languages, other methods may be employed.

References

1. Khobragade RN, Koli NA, Makesar MS (2013) A survey on recognition of Devnagari script. Int J Comput Appl Inf Technol II(I). ISSN: 2278-7720
2. Patil CH, Mali SM (2019) Handwritten Marathi consonants recognition using multilevel classification. Int J Comput Appl 975:8887
3. Mali SM, Patil CH (2015) Marathi handwritten numeral recognition using Zernike moments and Fourier descriptors. Int J Comput Appl 975:8887

4. Patil CH, Mali SM (2015) Segmentation of isolated handwritten Marathi words. Int J Comput Appl 975:8887
5. Nagane AS, Mali SM (2020) Segmentation of characters from degraded Brahmi script images. Applied computer vision and image processing. Springer, Singapore, pp 326–338
6. Nagane A, Patil CH, Mali SM (2018) Binarization of degraded Brahmi script estampage images. In: International conference on changing perspective of industries with Industry 4.0
7. Ramana Murthy OV, Roy S, Narang V, Hanmandlu M (2012) Devanagari character recognition in the wild. Int J Comput Appl (0975-8887) 38(4)
8. Dhandra BV et al (2006) Word-wise script identification from bilingual documents based on morphological reconstruction. IEEE Trans Pattern Anal Mach Intell 32(12)
9. Chiu Y-H, Chung K-L, Yang W-N, Huang Y-H, Liao C-H (2012) Parameter-free based two-stage method for binarizing degraded document images. Pattern Recogn 45:4250–4262
10. Prathima G, Rao GKS (2011) A survey of Nandinagari manuscript recognition system. Int J Sci Technology 1(1). ISSN (online): 2250-141X. www.ijst.co.in
11. Pal U, Sharma N, Wakabayashi T, Kimura F (2007) Off-line handwritten character recognition of Devanagari script. In: Proceedings of the 9th international conference on document analysis and recognition, Parana, 23–26 Sept 2007, pp 496–500
12. Ghosh D, Dube T, Shivaprasad AP (2010) Script recognition—a review. IEEE Trans Pattern Anal Mach Intell 32(12)
13. Ko AH-R et al (2009) Leave-one-out-training and leave-one-out-testing hidden Markov models for a handwritten numeral recognizer: the implications of a single classifier and multiple classifications. IEEE Trans Pattern Anal Mach Intell 31(12)
14. Pal U, Roy PP (2004) Multioriented and curved text lines extraction from Indian documents. IEEE Trans Syst Man Cybern Part B: Cybern 34(4)
15. Jindal MK, Sharma RK, Lehal GS (2007) A study of different kinds of degradation in printed Gurmukhi script. In: Proceedings of the international conference on computing: theory and applications, 2007
16. Mohahmed Althaf MK, Baritha Begum M (2012) Handwritten characters pattern recognition using neural networks. In: International conference on computing and control engineering (ICCCE 2012), 12–13 April 2012
17. Su B, Lu S, Tan CL (2013) Robust document image binarization technique for degraded document images. IEEE Trans Image Process 22(4)
18. Ntirogiannis K, Gatos B, Pratikakis I (2011) A performance evaluation methodology for historical document image binarization. IEEE
19. Sokratis V, Kavallieratou E, Paredes R, Sotiropoulos K (2011) A hybrid binarization technique for document images. Springer
20. Patvardhan C, Verma AK, Vasantha Lakshmi C (2012) Document image binarization using wavelets for OCR applications. In: ICVGIP '12 proceedings of the eighth Indian conference on computer vision, graphics and image processing, Article No. 60. ACM, New York, NY, USA
21. Neves RFdP, Zanchettin C, Mello CA (2013) An adaptive thresholding algorithm based on edge detection and morphological operations for document images. In: DocEng '13 proceedings of the 2013 ACM symposium on document engineering. ACM, New York, NY, USA
22. Gonzales CR (2013) Digital image processing, 3rd edn. Pearson
23. Khan MW (2014) A survey: image segmentation techniques. Int J Future Comput Commun 3(2)
24. Liu X, Duan Z, Xu W (2016) Improved computing method of mutual information in medical image registration. Int J Signal Process Image Process Pattern Recogn 9(4):415–424
25. Sara U, Akter M, Uddin MS (2019) Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study. J Comput Commun 7:8–18

Intelligent Systems in Medical and Healthcare Management

Cough Audio Signal-Based Clinical Emergency Classification of Corona Variant Infected Patients Using Multiclass SVM

N. M. Jyothi and S. Madhusudhanan

Abstract The variants of coronavirus both delta and omicron are much more contagious and affecting greater percentage of human population. In this research, an attempt is made to predict classification of clinical emergency treatment of corona variant infected patients using their recorded cough sound file. Cough audio signal features such as zero crossing and mel-frequency cepstral coefficients (MFCC), chromo gram (chroma_stft), spectral centroid, spectral roll off, spectral-bandwidth are to be extracted and stored along with patient ID, date, and timings. Digital signal processing of recorded cough audio file obtained needs to be cleaned and pre-processed and normalized to get a training dataset in order to build intelligent ML model using multiclass classifier SVM for predicting the class labels with maximum accuracy. The model proposed in this research paper helps to systematically plan and handle emergency treatment of the patients by classifying their severity based on the cough audio signal using SVM. The built model predicts and classifies the emergency treatment level as low, medium, and high with 96% accuracy.

Keywords Audio features · Multiclass SVM · Spectrum

1 Introduction

Pandemic corona and its variants are prevalent all over the world. The growing pandemic has put more pressure on the medical system thus paralyzing its functioning across the world. More number of infected patients is hospitalized due to severe breathing problem and coughs. They need immediate and special treatment with emergency medicines and oxygen cylinders. Corona positive patients experience severe cough as one of the major symptoms in majority of the cases. Cough audio signal of corona has distinguished features as compared to cough signals of other diseases. Hence, cough audio signals of corona has variations in relation to the infection level and it could be used to predict the of the patients emergency level

N. M. Jyothi (✉) · S. Madhusudhanan
Koneru Lakshmaiah Education Foundation, Green Fields Vaddeswaram 522502, India
e-mail: jyothiarunkr@gmail.com

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_26

effectively. The cough sound signal has various acoustic properties which could be extracted digitally by applying the methods of digital signal processing. The retrieved features are pre-processed further to obtain a valuable dataset, which is useful for training the learning model by applying multiclass support vector (SVM) algorithm. SVM is one of the powerful learning techniques, extensively applied in prediction and the classification of test data set. In this research, the acoustic properties of the cough sound signal is studied in detail for digital signal processing and all the implementation is carried out using Python. The model built is successful in prediction of class labels for clinical emergency classification of corona affected patients with 96% accuracy. The entire model is deployed using Python.

2 Literature Survey

Very few research works are done with respect to corona positive cough signal processing, and there is wide scope of research in this topic. Here is brief overlook of previous research conducted in cough audio sound. The novel coronavirus pandemic with its variants has paralyzed the human kind severely with huge loss of human lives all over the world. Hence, there is necessity of quicker and reliable detection of corona positive patients for providing timelier and better care and to prevent the spread of the virus [1]. Many medical literature underline that a system for the automatic, objective, and reliable detection of cough events is important and very promising to detect pathology severity in chronic cough disease [2]. Coughing is presented by an instant air dismissal through the airway which is signalized by a distinctive blow in the form of sound. The cough is a very important symptom of well over 100 diseases and is of medical significance. Changes in its character may have a considerable value in identifying the mechanisms of airway pathology present in respiratory diseases. Cough sound record analysis has significant value in prognosis because its changes may indicate the effectiveness of therapy or the progress of disease [3]. It is well known that the frequency of cough sound varies in different pathological conditions. Its identification could have diagnostic value [4]. Audio-based cough detection has become more common because of its utility in evaluating treatments and as it affects the life quality for individuals with chronic cough [5]. Acoustic analysis of cough with respect to time and frequency was carried out. The intention of this research was to discover whether such analysis of capsaicin-induced cough enables differences between normal subjects to be recognized [6].

The automatic recognition and counting of coughs in sound recordings for the Hull automatic cough counter (HACC) was developed for the analysis of digital audio recordings. HACC uses digital signal processing (DSP) to calculate characteristic spectral coefficients of sound events [7]. Novel features and a logistic regression model (LRM) for classification of wet/dry cough were developed. The method was evaluated on a cough dataset of pediatric coughs [8]. TB (Tuberculosis) screening on the automatic analysis of coughing sound signals was carried out. A database of cough audio recordings was collected and used to develop statistical classifiers. Main results: These classifiers use short-term spectral information to automatically distinguish between the coughs of healthy people and TB positive patients and healthy with an accuracy of 78% [9]. A range of cough sound intensity (CSI) parameters were determined from the cough sound signal. CSI was found out for the actual cough sound duration and also for a constant duration (time constant) to assess the potential for automation [10]. Support vector machine (SVM) is one of the state-of-the-art tools for linear and nonlinear pattern classification. The goal of SVM classifier is, reducing the number of the support vectors without compromising the classification [11]. The mathematical model for support vector machine (SVM) offers strong foundation and upright approach in solving data science problems [12]. A general framework for cough sound analysis developed, which includes automatic cough segmentation, feature extraction, and a general classification design that can be applied to a wide range of pulmonary diseases. For analysis, three evidence-based features were selected (variance, kurtosis, and zero crossing irregularity) [13]. A methodology for automated analysis of cough sound signals using support vector machines (SVM) was developed. Pre-processing is necessary for suppression of noise from cough signals and is the prime concern in accurate prediction of various respiratory disorders [14]. A support vector machines (SVMs)-based method is proposed for content-based audio classification and retrieval. Given a feature set, which in this work is composed of perceptual and cepstral feature, optimal class boundaries between classes are learnt from training data by using SVMs. Matches are ranked by using distances from boundaries [15].

3 Objective

The literature survey clearly indicates that no work is carried out using multiclass SVM in the prediction of clinical emergency classification using cough audio signal. Moreover, corona is a new pandemic and not much work is carried out on corona infected cough. This research is a novel work which aims at predicting classification of emergency level among corona and its variant infected patients using the prerecorded cough audio signal using multiclass SVM.

In this research, an attempt is made to predict classification of clinical emergency treatment of coronavirus infected patients using the recorded cough sound file of the patients. Cough audio signal features such as zero crossing and mel-frequency cepstral coefficients (MFCC), chromo gram (chroma_stft), spectral centroid, spectral roll off, spectral-bandwidth are to be extracted and stored along with patient ID, date, and timings. Using Python for digital signal processing of recorded cough audio file obtained needs to be cleaned and pre-processed and normalized to get training dataset and to make use of 75% of the training data set to build ML model using multiclass classifier SVM in prediction of the class labels with maximum accuracy.

4 Experimental Setup for Building the Proposed Model

The detailed procedure of the model built is explained in the following sections.

Obtaining cough sound–Cough audio signal of the patients can be obtained using the smart phone. As many of the smart phones come with built in sound recording. The nearby health worker can record the cough sound of the already tested corona affected patient for a period of 5 min. In order to get better clarity of the patient's infection criticality, the cough sound recording can be done three times first in morning 7.30 am, second in the afternoon at 1.30 pm, and third in the evening at 7.30 pm. The recording could be assisted by the from the patient's care takers or by patient himself, if he is in a manageable condition. The obtained cough sound file can be saved by name patient ID followed by session and day count. For example, 101_M_day1 for morning session of first day, 101_A_day1 for afternoon session first day, 101_E_day5 for evening session of patient ID 101 for day 5. The experiment was carried out on 100 patients for a period of 30 days.

The naming convention helps in identifying the cough sound file quickly and also in feature selection and extraction. The cough audio file obtained from patient can emailed or shared through What's App for further processing in our research experiment. Cough sound preprocessing, the cough file recorded from the patient using smart mobile phone through Email/What's App is saved in separate directory or folder created individually for each patient. The folder name is patient ID. Every day as long as the patient is hospitalized, the obtained file will be saved in their respective folder, as this helps in building training data set also in deep analysis of individual cough signal for future work.

The cough file so obtained from mobile phone is in MPEG-4 format. With extension .m4a, this audio file is converted to wave format keeping same file name but with extension .wav. The .m4a file is converted to .wav format using suitable online converter. After conversion of .m4a file to audio file, the original file is discarded as it is not needed and only .wav format example 103_day5.wav file stored in the folder 103 for patient id 103. .wav file thus obtained is suitable for audio feature extraction.

Cough audio signal processing using Python—Cough audio digital signal processing representing the sound electronically or digitally. Representing sound in digital format helps in describing the sound features. The visual analysis of the sound is carried out using spectrogram which is image format of the sound with respect time and frequency (see Fig. 1). Magnitude is measured in decibels (dB).

4.1 Audio Features for Classification

In this research work, features of the cough sound are retrieved from the audio file using Python and it's correspond numeric representation is obtained and training data set is built with suitable label which is used by the model for prediction.

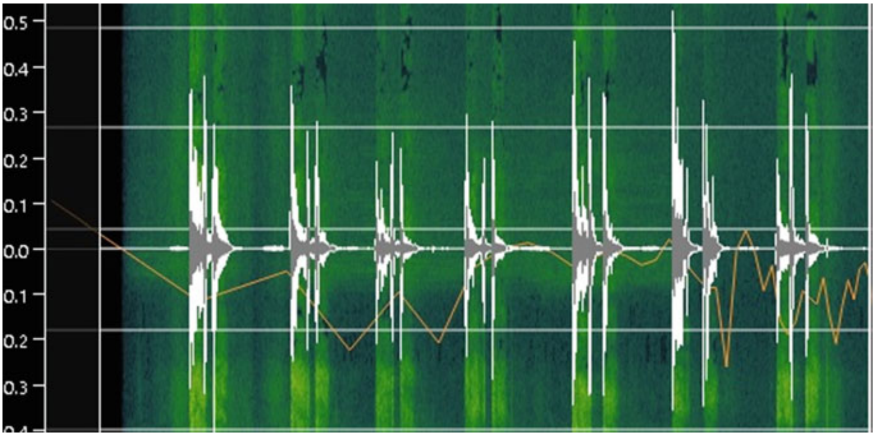


Fig. 1 .wav and spectrogram format of audio file segment

Physical features of audio signal—Cough sound has features which are physical or measurable. It involves zero crossing rate (ZCR), it is count of change in sign of audio signal. Short-time energy which is mean square of signal. Band level energy is energy within the spectrum region. Spectral centroid is the center energy of the spectrum. Spectral roll off represents gross spectral shape. Spectral flux is squared difference of spectral magnitude. Fundamental frequency (F0) is measure of the periodicity wave. Mel-frequency cepstral coefficient (MFCC) provides a compact representation of the short-time spectrum envelope.

Feature extraction using Python—Following experimental procedure is followed

Converting .M4a file to .wav file—Python program is written to convert .m4a cough audio files obtained from patients into .wav format using pydub, ffmpeg, and ffprobe packages. Code snippet of .M4a to .wav conversion is shown (see Fig. 2).

Input to the code is folder containing .M4a cough audio files of corona positive patients, and the output obtained is corresponding .wav files. Finally, all files in that folder will be converted to .wav files.

Noise removal from cough sound file—The cough audio file may consist of noise which could be removed by using following Python code. After processing, the output obtained is noise free cough audio file (see Fig. 3).

The code requires two input files

i. A noise audio file containing sample noise of the audio file. This file consists of sample noise
ii. A cough audio file which has noise that needs to be removed.

Silence zone removal—Cough may be continuous or may come in pause. So silence zone is not needed for audio processing. Hence, it is removed from the audio file using following steps.

Step 1: Raw cough audio file with noise is loaded. The Python code to show the visual representation of the cough audio signal is written and executed.

```python
try:
    track = AudioSegment.from_file(filepath,
            file_extension_final)
    wav_filename = filename.replace(file_extension_final, 'wav')
    wav_path = dirpath + '/' + wav_filename
    print('CONVERTING: ' + str(filepath))
    file_handle = track.export(wav_path, format='wav')
    os.remove(filepath)
except:
    print("ERROR CONVERTING " + str(filepath))
```

for (dirpath, dirnames, filenam... › for filename in filenames › if filename.endswith(tuple(form... › except

Fig. 2 Python code to convert .M4a to .wav

```python
import noisereduce as nr
import librosa
# load data
data,rate = librosa.load(r"C:\Users\ARUN\Desktop\corona paper-3\papr 4\cough.wav")
# select section of data that is noise
noisy_part =data[10000:15000]
# perform noise reduction
reduced_noise = nr.reduce_noise(audio_clip=data, noise_clip=noisy_part, verbose=True)
```

Fig. 3 Python code for noise removal

Silence zones in cough audio are found out and trimmed using librosa a Python module. As, corona patients can be adult males, females, or children maximum decibels of cough sound varies. So, in order to generalize the algorithm and to retain cough sound of child, the silence part is considered below 20 decibels (see Fig. 3). The visual representation of noise free cough audio file is obtained (see Figs. 4 and 5).

```
import librosa.display
import matplotlib.pyplot as plt
file_path = r"C:\Users\ARUN\Desktop\New folder\paper-2\101_M_Day1.wav"
x, sr = librosa.load(file_path)
X = librosa.stft(x)
Xdb = librosa.amplitude_to_db(abs(X))
plt.figure(figsize = (10, 5))
librosa.display.specshow(Xdb, sr = sr, x_axis = 'time', y_axis = 'hz')
plt.colorbar()
```

Fig. 4 Python code for visual representation of audio signal

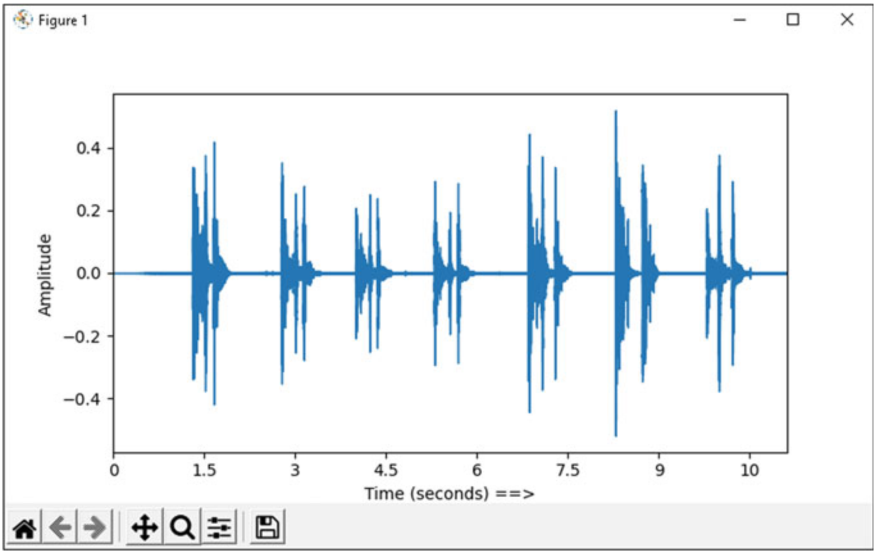


Fig. 5 Output showing visual representation of noise free cough audio signal

Step 2: Convert audio file into decibels and obtain the maximum decibel. Find maximum decibels value in cough audio file, any value below the maximum is to be chosen as silence part. In this experiment, considering 20 dB as minimum loudness of cough, 20 dB is fixed as minimum. Any sound below 20 dB is considered as silent zone and trimmed from the file. Python code is written to convert audio file into decibels and find maximum decibels in the cough audio file. The code along with maximum decibel output (245,248), 22,050 obtained (see Fig. 6).

```
import librosa.display
file_path = r"C:\Users\ARUN\Desktop\New_folder\paper-2\101_M_day1"
x,sr = librosa.load(file_path)
print(x.shape, sr)
```

```
Run: main
C:\Users\ARUN\PycharmProjects\m4a_to_wav\
(245248,) 22050

Process finished with exit code 0
```

Fig. 6 Python code to convert audio file into decibels

Step 3: Remove the silent part (below 20 dB) from the cough audio file. The Python code is written and implemented (see Fig. 7) to remove the entire zone which has energy level below 20 dB. Output obtained is 80 dB (max energy of sound in cough audio file). The program removes all silence zones from the audio clip and returns sound clip with cough sound with no silence zone. The same procedure is applied to all the audio files present in that folder. Code with output is shown (see Fig. 8).

```
wav_convet.py  silence.py  test.py  test1.py  main.py
import librosa.display
import numpy as np
from IPython.display import Audio
x,sr = librosa.load(file_path)
Audio(file_path)
n_fft = 2048
S = librosa.stft(x, n_fft=n_fft, hop_length=n_fft//2)
print(S.shape)
# convert to db
D = librosa.amplitude_to_db(np.abs(S), ref=np.max)
np.max(abs(D))
def displayTime(startFrame, endFrame):
    print(' start time: ' + str(startFrame/sr) + ', end time: ' + str(endFr
nonMuteSections = librosa.effects.split(x)
nonMuteSections
```

Fig. 7 Python code to remove silent zone below the maximum decibels

```
C:\Users\ARUN\PycharmProjects\m4a_to_wav\venv\Scripts\python.exe
[ 0.0000000e+00  0.0000000e+00  0.0000000e+00 ... -9.3999086e-05
 -1.2000898e-04 -7.3801108e-05]
(1025, 240)
 start time: 0.4179591836734694, end time: 11.122358276643991

Process finished with exit code 0
```

Fig. 8 Output showing time interval of silence zone being removed from cough audio file

Other feature extraction from cough audio file—Cough audio file contains zero crossing rate, spectral centroid, spectral-bandwidth, MFCC, RMSE, and spectral roll off chromo gram (Chroma Stft) features. These features of cough play important role in determining the class labels of the patients for the purpose of classification of clinical emergency. As cough features differ from patient to patient, it helps in prediction of class labels. Python code to extract the above-mentioned audio features is written and executed successfully (see Fig. 9).

The output of the above code is cough_features.csv file containing all the necessary features along with patient ID. The program generated .csv file for five patients containing cough audio features (see Fig. 10). In the same way, the cough audio features of all the remaining recorded cough files are extracted in the form .csv file. The program has less time complexity.

```
from librosa import feature
import csv
fn_list_i = [
    feature.chroma_stft,
    feature.spectral_centroid,
    feature.spectral_bandwidth,
    feature.spectral_rolloff
]
fn_list_ii = [
    feature.rms,
    feature.zero_crossing_rate
]
```

Fig. 9 Cough feature extraction

	A	B	C	D	E	F	G	H
1	Patient ID	chroma_stft	spectral_centroid	spectral_bandwidth	spectral_rolloff	rmse	zero_crossing_rate	mfcc
2	1	0.5304858	3657.366	5674.0325	8470.058	-17.6616	0.01605	0.086544
3	2	0.6304845	3625.366	4685.0325	8570.058	-23.6612	0.01605	0.045943
4	3	0.7310658	3525.345	5675.0215	8350.053	-18.6542	0.01804	0.078325
5	4	0.6304858	4627.366	5678.0325	6750.058	-22.6616	0.03695	0.05439
6	5	0.4702683	2624.483	3639.4968	4994.693	-16.3614	0.02301	0.034568
7	6	0.5304858	3627.366	4685.0325	8570.058	-23.6616	0.01405	0.034536
8	7	0.3340734	4584.518	2734.2764	7338.294	-27.2013	0.02871	0.15522
9	8	0.4304858	6367.366	4665.0325	7650.058	-27.2013	0.01305	0.045677

Fig. 10 Output stored in cough_features.csv file

4.2 Feature Selection

Pre-recorded cough audio of the patient is used in classification of the clinical emergency for the purpose of treatment. The cough sound is characterized by zero_crossing_rate, MFCC, spectral_centroid, spectral_rolloff spectral_bandwidth, and chromo gram (chroma_stft). All these cough audio features are used to predict the clinical emergency treatment classification. The recorded cough sound is collected three times (morning, afternoon, and evening) every day for ten minutes duration. Each and stored in the date/session wise folder for building training dataset is collected by 40 patients for five succeeding days. The features of all the received files are extracted by the end of the day and stored in .csv file along with patient ID. All the .csv files obtained each day can be merged into single file at the fifth day for further processing.

The scatter diagram of raw cough sound features (see Fig. 11). It shows high density in the bottom region. For few features peak values for certain features with unequal distribution.

Normalization—The next step is applying normalization. The features selected by the cough audio signal have wide range of different values for every feature. In order to yield an optimal balance between “preserving information” and “overfitting” for different classifiers and also for the ML Model to predict accurately, the features should be normalized. In this experiment, Z-score normalization is applied on the extracted features. The Z-score of all the cough audio features calculated using Pandas and Python code (see Fig. 12) with cough_features.csv as the input.

Reason for choosing *Z*-score normalization is that it handles outliers well. After applying *Z*-score the cough sound features are rescaled *Z*-score squeezes the raw values of the data set in the range of approximately 40,000–100,000 down into a range from roughly −1 to +4 (see Fig. 13).

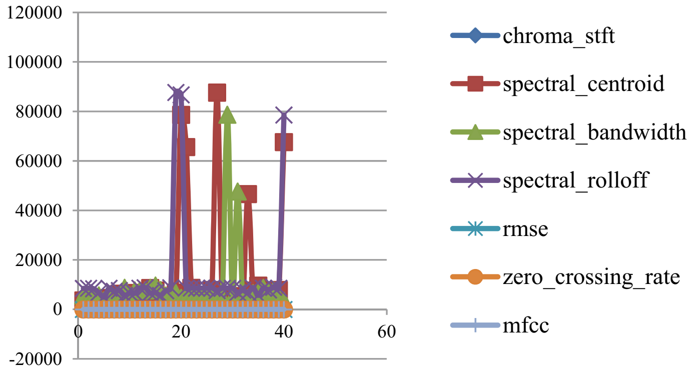


Fig. 11 Scatter diagram showing raw cough audio features

```
from scipy.stats import zscore
import pandas as pd
filename=r'C:\Users\ARUN\Desktop\New folder\paper-2\cough_Features.csv'
norm = pd.read_csv(filename)
zscore_zero_crossing =  zscore(norm['zero_crossing_rate'])
zscore_chroma_stft=zscore(norm['chroma_stft'])
zscore_spectral_centroid=zscore(norm['spectral_centroid'])
zscore_spectral_bandwidth=zscore(norm['spectral_bandwidth'])
zscore_spectral_rolloff=zscore(norm['spectral_rolloff'])
zsore_rmse=zscore(norm['rmse'])
zscore_mfcc=zscore(norm['mfcc'])
print(norm.head())
```

```
   Patient ID  chroma_stft  ...  zsore_rmse  zscore_mfcc
0           1     0.530486  ...   -0.248591     0.797161
1           2     0.630485  ...   -0.243648    -0.770974
2           3     0.731066  ...   -0.254524     0.479721
```

Fig. 12 Python code for calculating *Z*-score of all cough audio features of cough_features.csv

5 Identification of Class Labels and Building Training Data

As everyday cough audio data collected is huge in volume, the entire process of the experiment in this research is fully automated using Python and training dataset class label determination is carried out programmatically, which is faster and accurate than done manually by humans. Data was collected from 100 patients for 30 days at the rate of 3 times per day. This resulted in building rich training data set of size 9000. From each patient, 45 recordings are considered.

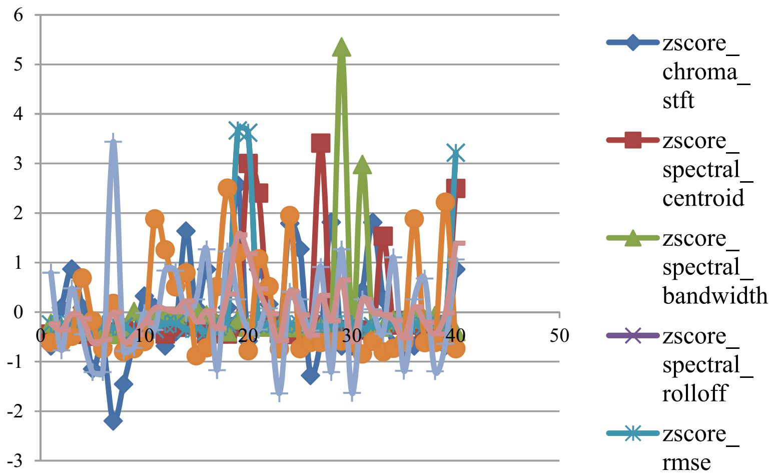


Fig. 13 Scatter diagram of cough audio features after applying *Z*-score

Three types of classes are chosen for classifying the training data set. Corona positive patients are classified based on treatment level as—most urgent (high), moderate (medium), and tolerate (low). The classes are labeled as high (numeric value 1), medium (numeric value 0), and low (numeric value −1). The class name represents the clinical treatment category of corona affected patients.

The cough data set collected from corona positive patients for first five days is used as training dataset for the predicting machine learning model. Training dataset is pre-processed and normalized by the methods mentioned in above steps of this paper. In order to label the training dataset, the below logic is applied.

Class_Value for a data set record = Average (Z-score values of cough audio features for that record).

Class_Value = Total_sum_of (Z_Score values of (Chroma gram + spectral centroid + spectral-bandwidth + spectral roll off + RMSE + zero crossing rate + mfcc)/7.

Any one of the two methods mentioned below could be used to label the data in Z_Score_cough_features.csv file, which contains the Z_Score values of all features extracted from cough audio file.

i. Python code which takes Z_Score_cough_features.csv file as input file and calculates Class_Value of every record row by row is written which labels the data based on rules (see Table 1).
ii. Apply ready formula available in Excel file and get the Class_Value and then label it.

Table 1 Rules for class labels

S. No.	Class_Value	Class label	Class
1	< 0	Low	0
2	≥ 0 && ≤ 0.5	Medium	1
3	> 0.5	High	2

5.1 Proposed Model

The architecture diagram (see Fig. 14) shows the steps used in identifying class labels both by humans and program. In this research, both methods are used. In case of ambiguity in class labeling, human intervention is done and resolved. Ultimately training data set is obtained which can be used to train the model in order to get accurate prediction when test data is input.

6 Emergency Classification and Prediction Using Multiclass SVM—Implementation

In order to classify the corona positive patient's clinical emergency into three classes, multi classification using one versus one approach is applied. Multiclass SVM is best suited machine learning algorithm for classification where more than two classification is required with moderate sample size. Data size sample for the current research, moderate sample size is sufficient. Multiclass SVM is found to be more accurate and faster when compared to other ML algorithms where multiple classification is required. Hence, in this research, multiclass SVM is chosen for prediction purpose.

In our research, three classes are considered for classifying the corona affected patient's emergency level classification. So, the numbers of classes considered for classification are three. The cough audio data frame consists of 9000 data sets. The 75% of the data set is used for training purpose and remaining 25% used as test data.

Training data set is shuffled well using shuffle () function to get uniform distribution of class labels. The sample code snippet used for classification prediction of corona patients emergency treatment is shown (see Fig. 15).

7 Results and Evaluation

Confusion matrix and classification report with precision, recall, and accuracy along with overall average of classification is computed using appropriate built in methods of sklearn package. The output obtained is shown (see Fig. 16).

The proposed model worked successfully and predicted classification with 96% accuracy confusion matrix shows higher number of True Positive values [43 support

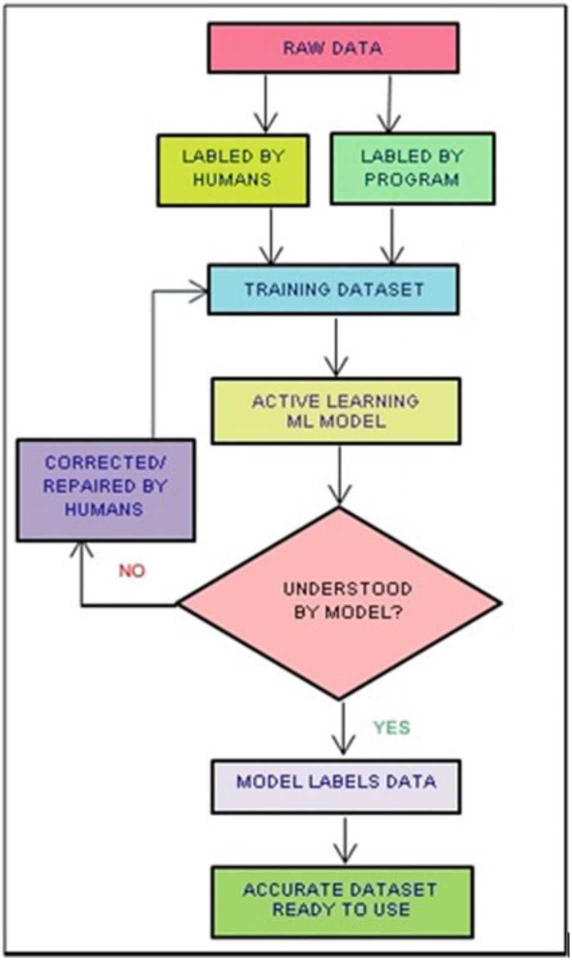


Fig. 14 Proposed model for identifying class labels and building training data

count for class low (0), 44 support count for class Med (1), and 113 support count for class high (2)] and obtained less or zero number of false negative and false positive. Average precision score and micro averages of all three classes predicted is 91%. Recall rate is 100% achieved for class low (0) and class high (2).

The result obtained by the model is compared with the physician's report. The result is found to be matching with respect to the clinical criticality of the patient.

The graph is plotted (see Fig. 17) to depict the confusion matrix obtained in the output. It shows the True Positive and Negative (TP, TN) and False Positive and Negative (FP, FN) values of each class. The graph (see Fig. 18) shows precision, recall, F1-score, and support for each class more clearly.

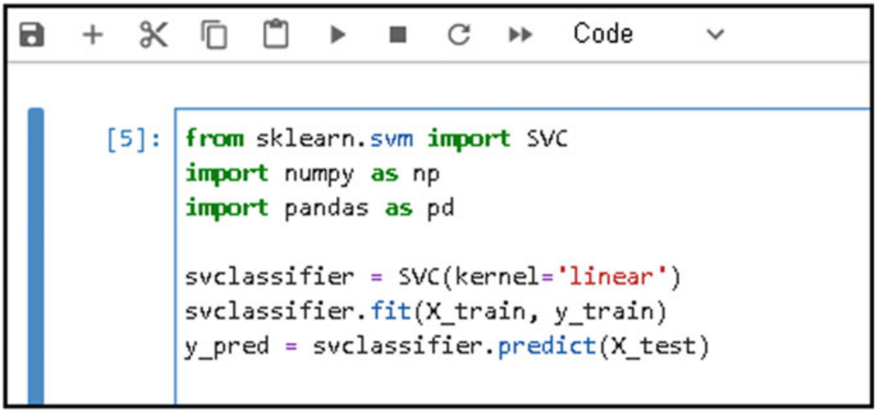


Fig. 15 Python code for predicting classification

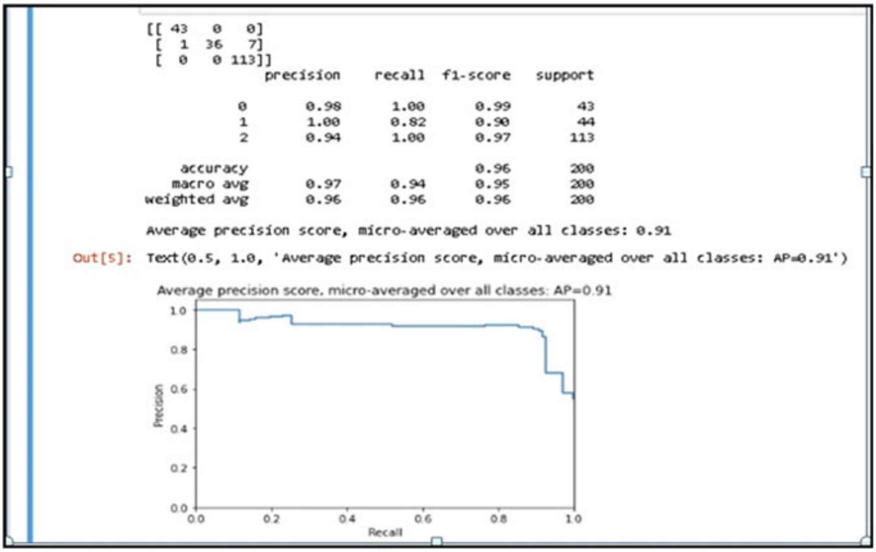


Fig. 16 Output of multiclass SVM showing classification report with confusion matrix and accuracy of prediction

8 Discussion

The multiclass prediction using multiclass SVM on the overall data set resulted in 96% overall accuracy. In this, the model studies and learns from the cough pattern of all the patients and prediction is made using the overall knowledge obtained by the model. After prediction, we can check the prediction made for individual patient by their ids. For a particular patient, classification label with highest count is considered

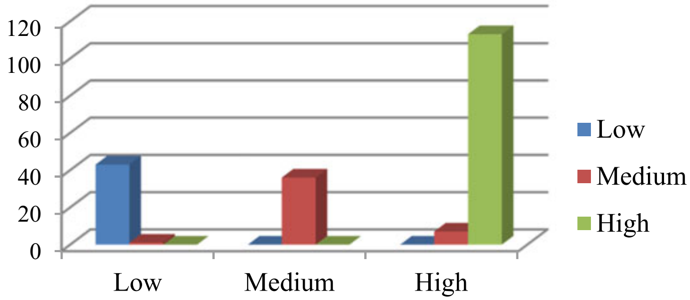


Fig. 17 Bar chart showing confusion matrix percentage

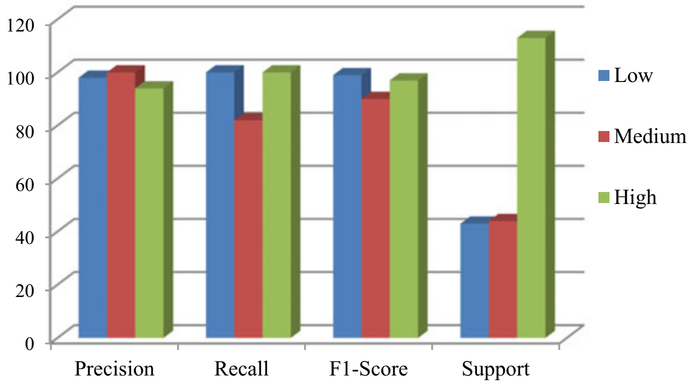


Fig. 18 Bar chart showing precision recall F1-score and support values of classification

as his level of emergency. For example, for patient 101, after the model returns classification labels, if count of class low is 55, class medium is 25, and class high is 20, then we can conclude that his emergency classification level is low as it has highest count. In case of tie in count values among any two or all three classes, or close count (with a difference of less than 5–8) among all three classes is obtained, then we can take the emergency level as high as importance is given for worst case of the patient. In such cases, we need to collect more samples of cough audio and increase the data set size and re run the model.

However, the model could be customized to predict the classification by applying to the multiple cough audio data set of individual patient only. For this, we can collect more cough audio samples for a single patient over varying period of time intervals. Once we obtain sufficient cough audio samples, a data set of size of approximately 100 or more for single patient, prediction can be made for individual patient alone. The above explained method is applied to find the actual classification label. The result obtained is more satisfactory, and no change in the accuracy is observed. Here, the model learns the cough pattern of individual patient and is still more precise and true when compared to the prediction classification where data set consists of cough samples of all patients. Here, separate dataset is to be maintained for individual patient.

9 Conclusion

The multiclass classification SVM model, is successful in predicting the clinical emergency classification of corona patients as low, medium, and high using their cough audio signal. This model is useful to the health care system in classification of clinical emergency treatment level and the hospital can accordingly make arrangements of resources for the purpose of treatment in a more organized way. And also helps patients to be treated more effectively and faster thus saving lives of people and reducing mortality rate of people affected by the contemporary corona pandemic. The entire process of the model building, from feature selection to prediction is carried out programmatically. Python code is used for audio digital signal processing and also in further stages of feature extraction, selection, building training model, and prediction.

Multiclass SVM is found to be ideal algorithm to work with data set consisting of cough audio signal features. The data set required for this research is moderate in size, and it is suitable for multiclass SVM with more time efficiency. The work carried out in this research paper is first of its kind, and it is a novel approach to find the emergency level prediction during pandemic using the cough audio signal. The research contributes to the application of multiclass SVM in the prediction of clinical emergency among corona positive patients using the audio features hidden in the cough signal. The dataset obtained can be more useful in further research and analytics and still advanced models can be applied for deeper analysis of still larger volume of data set.

References

1. AlJame M, Ahmad I, Imtiaz A, Mohammed A (200) Ensemble learning model for diagnosing COVID-19 from routine blood tests. Inform Med Unlcoked 21 (2020)
2. Shi Y, Liu H, Wang Y et al (2018) Theory and application of audio-based assessment of cough. Hindawi J Sens 2018:Article ID 9845321
3. Korpas J, Sadlonova J, Vrabec M (1996) Analysis of the cough sound: an overview. Pulm Pharmacol 9:261–268
4. Korpas J, Vrabec M, Sadlonova J, Salat D, Debreczeni LA (2003) Analysis of the cough sound frequency in adults and children with bronchial asthma. Acta Physiol Hung 90:27–34
5. Larson EC, Lee T, Liu S, Rosenfeld M, Patel SN (2011) Accurate and privacy preserving cough sensing using a low-cost microphone. In: Proceedings of the 13th international conference on ubiquitous computing. ACM, pp 375–384
6. Doherty MJ, Wang LJ, Donague S, Pearson MG, Downs P, Stoneman SAT, Earis JE (1997) The acoustic properties of capsaicin-induced cough in healthy subjects. Eur Respir J 10:202–207

7. Barr SJ, Dane AD, Morice AH et al (2006) The automatic recognition and counting of cough. BioMedical Central. Cough 2:8
8. Swarnkar V, Abeyratne U, Chang A, Amrulloh Y, Setyati A, Triasih R (2013) Automatic identification of wet and dry cough in pediatric patients with respiratory diseases. Ann Biomed Eng 41(5):1016–1028
9. Botha GHR, Theron G, Warren RM, Klopper M (2018) Detection of tuberculosis by automatic cough sound analysis. Inst Phys Eng Med Physiol Meas 39(4):25
10. Lee KK, Matos S, Ward K, Raywood E, Evans DH, Moxham J et al (2012) P158 cough sound intensity: the development of a novel measure of cough severity. Thorax 67(Suppl 2):A130.3–A1A131
11. Manikandan J, Venkataramani B (2010) Study and evaluation of a multi-class SVM classifier using diminishing learning technique. Neurocomputing 73:1676–1685
12. Kumari PR, Bojja P, Galla A, Pratap R (2020) Analysis of mathematical model of support vector machine techniques for early prediction of medical diseases. Int J Pharm Res 12(4):3379–3385
13. Infante C, Chamberlain D, Fletcher R, Thorat Y, Kodgule R (2017) Use of cough sounds for diagnosis and screening of pulmonary disease. In: 2017 IEEE global humanitarian technology conference (GHTC), pp 1–10. https://doi.org/10.1109/GHTC.2017.8239338
14. Bhateja V, Taquee A, Sharma DK (2019) Pre-processing and classification of cough sounds in noisy environment using SVM. In: 4th international conference on information systems and computer networks (ISCON), GLA University, Mathura, UP, India, 21–22 Nov 2019
15. Li SZ, Guo GD (2003) Content-based audio classification and retrieval using SVM learning. IEEE Trans Neural Netw 14(1):209–215. https://doi.org/10.1109/TNN.2002.806626

Binary Classification of Mammograms Using Horizontal Visibility Graph

Anirban Ghosh, Priya Ranjan, Naga Srinivasarao Chilamkurthy, Richa Gulati, Rajiv Janardhanan, and Pooja Ramakant

Abstract Breast carcinoma, the most common cancer in women across the world now accounts for almost 30% of new malignant tumor cases. Despite the high incidence rate, breast cancer mortality has been maintained under control thanks to recent advances in molecular biology technology and an enhanced level of complete diagnosis and standard therapy. The method strives to overcome the clinical dilemma of undetected and misdiagnosed breast cancer, resulting in a poor clinical prognosis. Early computer-aided detection by mammography is an important aspect of the plan. In most of the diagnostic strategies currently in vogue, undue importance has been given to one of the performance metrics instead of a more balanced result. In our present study, we aim to resolve this dogma by first converting the mammograms into their equivalent graphical representation and then finding the network similarity between two such generated graphs. Subsequently, we will also elaborate on the use of horizontal visibility graph (HVG) representation to classify images and use Hamming-Ipsen-Mikhailov (HIM) network similarity (distance) metric to develop

A. Ghosh (✉) · N. S. Chilamkurthy
SRM University AP, Neerukonda, Guntur, Andhra Pradesh 522502, India
e-mail: anirban.g@srmap.edu.in

N. S. Chilamkurthy
e-mail: nagasrinivasarao_ch@srmap.edu.in

P. Ranjan
Bhubaneswar Institute of Technology, Info Valley, Harapur, Odisha 752054, India
e-mail: priyaranjan@bit.edu.in

R. Gulati · R. Janardhanan
Medical and Health Sciences, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu 603203, India
e-mail: richag@srmist.edu.in

R. Janardhanan
e-mail: rajivj@srmist.edu.in

P. Ramakant
King Georges' Medical University, Lucknow, Uttar Pradesh, India

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_27

novel triage mammograms according to the severity of the disease. Our HVG-HIM metric-based classification of mammograms had an accuracy of 88.37%, specificity of 92%, and sensitivity of 83.33%. We also clearly highlight the trade off between performance and processing time.

Keywords Breast tumor · HVG · HIM · Mammograms · Image classification

1 Introduction

Breast cancer is the most diagnosed reproductive cancer among women with an incidence rate of 2.2 million across the world and 1.7 lakh across the Indian subcontinent. Among all, ductal carcinoma in situ (DCIS) and invasive carcinoma are found to be quite common compared to phyllodes tumors and angiosarcoma [1]. It is important to understand however that not all lesions in the breast are cancerous (or malignant) and early diagnosis with appropriate and timely clinical intervention can alleviate the clinical prognosis of the patients.

There has been a recent spurt in several computer-aided diagnostic (CAD) tools for disease detection from medical images where to avoid the inherent instability and convergence issue in process-based approach, people have explored the usage of similarity based tagging using Earth Mover's Distance (EMD) [2–5], HVG [6–9] and similar such techniques. The premise and algorithm used for [2] is similar to our current work except for we use a combination of visibility graph and HIM similarity metric which can handle nuances in images better compared to the less sophisticated EMD approach.The VG algorithm has proven to be a useful way to study the structural properties of time series [10–13], capturing their level of regularity or randomness. The primary advantage of this method compared to others is its low computational cost. This method can be used to detect nontrivial properties from the series, such as fractality [14] and reversibility [15]. The motivation behind using HVG in distinguishing between mammograms containing benign and malignant lesions is to implement a simple, easy to use algorithm which can discern between time series and classify them by using different network attributes such as HIM which takes advantage of the local nature of Hamming distance and global nature of Ipsen-Mikhailov distance. The expectation is that results from the proposed algorithm coupled with other modalities such as breast cytology (Fine Needle Aspiration Cytology), gene expression patterns (Estrogen Receptor, Progesterone Receptor, Human Epithelial Growth Factor-2, Androgen Receptor) along with the oncogenes such as BRCA1/BRCA2 can form the rationale for a multi-modal multi-sensor fusion computational platform for rapid and precision-oriented tagging of breast cancer patients at the community level with a view to significantly improve the health outcomes.

The rest of the paper is organized as follows: in Sect. 2 we elaborate and describe the existing relevant works in literature. Section 3 elaborates on the proposed classification and the processing technique while Sect. 4 validates the efficacy of the proposed method in terms of various performance metrics and finds the trade off between accuracy and time complexity. We finally conclude the paper in Sect. 5.

2 Related Works

Breast cancer detection from mammograms is a classical field now where techniques from image processing have found a direct application in it [16]. This work processes mammogram and MRI and aims to detect the tumorous portion and delineated it by different segmentation methods. It is well known fact that mammogram features are modified by noise, distortion and artifacts and to overcome them, Ref. [17] proposes a skewness reformed complex diffusion based unsharp masking where smoothing filter is adjusted with a skewness reformed complex diffusion (SRCD). Moreover, Ref. [18] starts with the objective to investigate whether machine learning-based models using 3-T multiparametric MRI (mpMRI) can detect Ki-67 and histologic grade in stage I–II luminal cancer and concludes that it demonstrated decent diagnostic capability for detecting both in patients with luminal breast cancers. To improve the state of art of breast cancer detection methods, Ref. [19] proposes multi-modal sensing, i.e., information fusion of breast MRI and mammographic traces for superior diagnostic management. Performance evaluation of contrast-enhanced ultrasound (CEUS) based on radiomics analysis to distinguish benign from malignant breast masses is proposed in [20] and it concludes that CEUS based texture metrics can differentiate between benign and malignant breast masses leading to reduced unnecessary breast biopsies.

There is an extensive body of literature elucidating the stellar role of Visibility graph (VG)-based methods to transform time series signal into networks and then use network attributes to investigate various quantities of interest. For example [21] demonstrates that the hybrid multiplex visibility graph and 2D convolutional neural network (MVG-CNN) method detects sleep states from WFCI data with higher precision. Visibility graph has been used to investigate the interaction between cardiac and respiratory activities in preterm infant sleep [22]. A method for efficient generation of visibility and horizontal visibility graphs have been proposed in [23] to look into difficult problem in bioinformatics, the classification of the secondary structure of low-homology proteins. Experiment report [24], describes a way to differentiate non-fatigue, and fatigue conditions using nonlinear techniques, namely, binary and weighted Visibility Graphs (VG). Effectiveness of visibility graph-based methods is demonstrated in extracting the time varying and specific gene features underlying realistic complex gene expression data from time series [25]. In summary as can be seen quite a few work has explored the usage of visibility graph for medical data analysis to various effects and conclusions. To the best of our knowledge, this is the first attempt to use a visibility graph in conjunction with the HIM metric to classify mammography lesions as benign or malignant. The following are the key contributions of this work:

- Processing of mammograms to represent them as horizontal visibility graphs.
- Using HIM network similarity metric for binary mammogram classification.
- Determination of trade off between performance of the classification algorithm based on considered metric and computation time.

3 Methodology

The current section provides a detailed explanation of the proposed technique after outlining the relevance of the current study in Sect. 2. The section begins by providing a gist of the data used followed by processing of images to obtain visibility graphs and the subsequent HIM distance between them. The details of the training and validation process is provided next and finally concluding the section by providing an overview of the algorithm.

3.1 Gist of Datasets

The mammograms are obtained from the dataset in [26]. The database consisting of 3728 images from 1775 patients contains biopsy confirmed classification of tumors into benign and malignant. Similar to the study in [27], in this study too we examine right breast radiographs of women between the ages of 40 and 50 years, where the effect of the disease has been statistically found to be more common. In the rest of the paper the above conditions will be referred to as *preset criteria*.

3.2 HVG Representation and HIM

The different steps for conversion of a mammogram to its HVG equivalent representation is shown pictorially in Fig. 1. An image is initially downsampled in favor of realistic processing time given our resources. We use downsampled images of size 16×16 and 32×32 for our work. The downsized image is then expressed in matrix form and then stacked as vector for time series representation as shown in Fig. 1. Once the time series representation is obtained the stage is set for conversion to HVG by using the visibility graph algorithm in R programming language. On application of the algorithm we obtain a network of pixels. Different visual representation of the network such as heatmap, histogram, etc is possible to obtain various insights on the spatial distribution of pixel intensity.

Out of multiple network distance available, HIM is used for comparing the visibility graphs due to the fine balance maintained between global and local nature of a network in calculating the distance or similarity using a combination factor xi equal

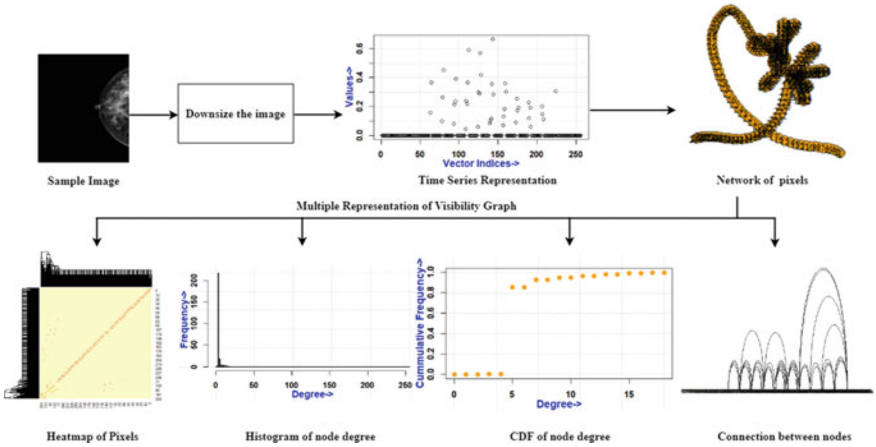


Fig. 1 Different stage of processing for transformation of mammograms to HVG

to 1 as shown in Eq. 1. The Ipsen-Mikhailov (IM) distance takes precedence when $\xi \to \infty$ whereas for $\xi = 0$, Hamming distance (H) takes precedence.

$$d_{\mathrm{HIM}} = \frac{1}{\sqrt{1+\xi^2}}\sqrt{\xi \mathrm{IM}^2 + H^2} \tag{1}$$

3.3 Training and Performance Analysis

In addition to observing the effect of image size to accuracy and processing time we also intend to investigate the effect of cardinality of training set on efficacy of the proposed method. In that direction we have used training sets of different sizes for implementation of the proposed work. The sets are differentiated based on their cardinality and are classified as

- **Case I**: containing 10 images in each set,
- **Case II**: where the cardinality of each training set is 20 and
- **Case III**: in which each set consists of 50 images.

In training the pairwise HIM distance between images of the same group are found to create a HIM matrix as shown in Fig. 2. In the HIM matrix the image which has the lowest average HIM distance is selected as the representative of the group to be used for disease tagging as explained in Sect. 3.4. In performance analysis we find the efficacy of the proposed technique in terms of accuracy, specificity, sensitivity, precision and $F1$ score which are calculated from the confusion matrix using the equations in [28]. True positives (TP) and true negatives (TN) are considered as the outcomes, where the proposed method correctly identifies an image with malignant

	a10	a11	a12	a13	a14	a15	a16	a17	a18	a19	Average	Rank
a10	0	0.004659	0.006064	0.009923	0.00659	0.005192	0.008937	0.00671	0.009376	0.008844	0.006629557	9
a11	0.004659	0	0.004646	0.006822	0.005465	0.004997	0.006429	0.004821	0.006268	0.006	0.005010727	8
a12	0.006064	0.004646	0	0.004877	0.003829	0.007379	0.004195	0.002146	0.005467	0.00433	0.004293254	3
a13	0.009923	0.006822	0.004877	0	0.002298	0.010053	0.004528	0.00521	0.002163	0.002393	0.004826593	7
a14	0.00659	0.005465	0.003829	0.002298	0	0.007144	0.004302	0.004194	0.003449	0.002834	0.004010504	1
a15	0.005192	0.004997	0.007379	0.010053	0.007144	0	0.008872	0.006735	0.009313	0.008917	0.006860179	10
a16	0.008937	0.006429	0.004195	0.004528	0.004302	0.008872	0	0.003673	0.003539	0.002749	0.004722367	6
a17	0.00671	0.004821	0.002146	0.00521	0.004194	0.006735	0.003673	0	0.00527	0.004355	0.004311483	4
a18	0.009376	0.006268	0.005467	0.002163	0.003449	0.009313	0.003539	0.00527	0	0.002047	0.004689219	5
a19	0.008844	0.006	0.00433	0.002393	0.002834	0.008917	0.002749	0.004355	0.002047	0	0.004246795	2

Fig. 2 HIM matrix of abnormal image cohort for *Case I* with 16×16 image size

and benign lesion respectively.In addition we also do a comparative study of the accuracy and required processing time for the two different sizes of image. A random mix of 43 images satisfying the *preset criteria* and different from the images used for training is used for the performance analysis. The overview of our implementation is presented in the following section.

3.4 Overview of Implementation

An overview of the workflow of the proposed method is shown in Fig. 3. A certain number of images as per the *preset criteria* and different cases are first segregated into two training sets as described in Sect. 3.3. In each set the images are first converted to their Horizontal Visibility Graph (HVG) representation the details of which is explained in Sect. 3.2. The pairwise Hamming-Ipsen-Mikhailov (HIM) network similarity (distance) metric between HVG equivalent of the images forms the HIM matrix which determines the representative of the particular group as indicated in yellow in Fig. 2.

In the next phase HIM distance between the graphical equivalents of each image from the validation set and those from the two training sets is calculated where a lower HIM value indicates higher similarity between the graphs or networks being compared and vice versa. Thus as shown in the decision box of Fig. 3, if the evaluated HIM distance between a test image and the representative image of the healthy training group is lower than when compared with the other representative then the decision is taken in favor of the former and the test image is tagged as having benign lesion.

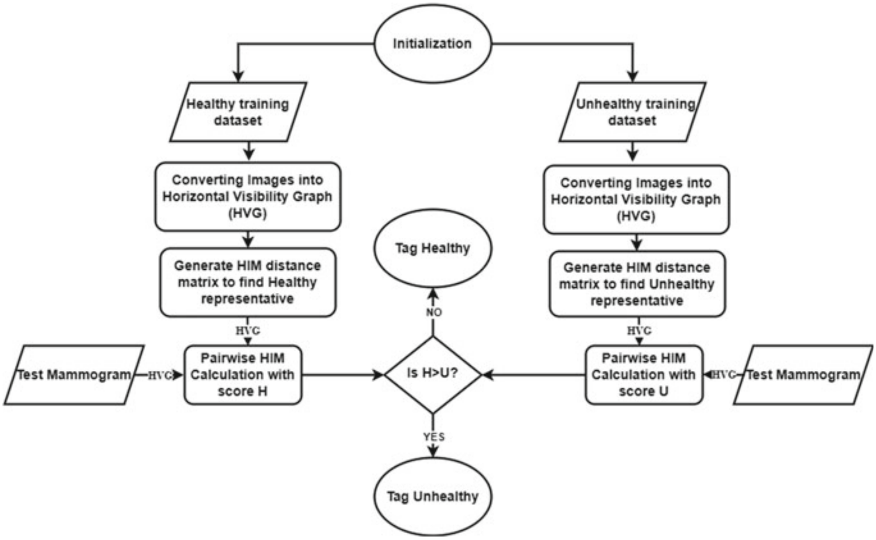


Fig. 3 Overview of implementation

4 Results

In this section we first present the efficacy of the proposed algorithm for the different cases when the image size are 32×32 and 16×16 respectively. We conclude this section by finding the effect of image size on accuracy and computation time.

4.1 Results for Image Size 32×32

The results from the confusion matrix and performance metric comparison as shown in Table 1 can be summarized as

- *Case III* outperforms all other cases in terms of all the performance metric by quite a distance. This validates the fact that improving the diversity in the training set balances the effect of outliers in each set.
- *Case II* outperforms *Case I* in terms of most of the performance metric except for sensitivity. As is observed from the confusion matrix it is due one extra incorrect detection of image with malignant lesion in *Case II* that the balance shifts.

Table 1 Confusion matrix and performance comparison with image size 32 × 32

	Actual labels		
Predicted labels	*Case I*		
		Benign	Malignant
	Benign	16	5
	Malignant	9	13
	Case II		
		Benign	Malignant
	Benign	19	6
	Malignant	6	12
	Case III		
		Benign	Malignant
	Benign	23	3
	Malignant	2	15
Metric	***Case I***	***Case II***	***Case III***
Accuracy (%)	67.44	72.09	88.37
Sensitivity (%)	72.22	66.67	83.33
Specificity (%)	64	76	92
Precision (%)	59.09	66.67	88.24
$F1$ score (%)	65	66.67	85.71

4.2 *Results for Image Size* 16 × 16

The results as reflected in Table 2 can be summarized as follows

- The representatives in *Case II and III* are found to be same and hence in this case there is no improvement in performance despite having different sized training set which is in contrast to the previous scenario.
- *Case II and III* outperforms *Case I* in most of the metric except sensitivity, a trend which was also observed for bigger sized images. Here two extra incorrect detection of images with malignant lesion leads to degradation in sensitivity from 88.89% in *Case I* to 77.78% in *Case II and III*.

4.3 *Computation Time Versus Accuracy*

R programming language was used in a laptop with Intel(R) Core(TM) i5-8250U CPU @ 1.60 GHz and 1.8 GHz was used for all computations. The computation time in Table 3 refers to the cumulative time required to generate the HIM matrix for both groups in a given case and to tag all the test images in the validation set. As can be clearly seen there is a clear trade off between processing time and image size

Table 2 Confusion matrix and performance comparison with image size 16 × 16

	Actual labels		
Predicted labels	*Case I*		
		Benign	Malignant
	Benign	9	2
	Malignant	16	16
	Case II and III		
		Benign	Malignant
	Benign	14	4
	Malignant	11	14
Metric	***Case I***	***Case II and III***	
Accuracy (%)	58.14	65.12	
Sensitivity (%)	88.89	77.78	
Specificity (%)	36	56	
Precision (%)	50	56	
$F1$ score (%)	64	65.12	

Table 3 Trade off between computation time and accuracy

Image size		*Case I*	*Case II*	*Case III*
32 × 32	Computation time (min)	140	350	1280
	Accuracy (%)	67.44	72.09	88.37
16 × 16	Computation time (min)	10	35	328
	Accuracy (%)	58.14	65.12	65.12

across different cardinality of training sets. An interesting observation is the better performance of bigger image (32 × 32) even with smaller training sets (*Case II*) and approximately comparable processing time compared to smaller image in *Case III*.

5 Conclusion

A combination of static algorithm like HVG and HIM can provide a fruganomic computational platform capable of providing precision-oriented screening of breast cancer patients at community level. This computational platform could be easily ported as a web-service or a low-end mobile health application to essentially tag the community-based evidences with hospital-based registries. Such an integrated strategy would form the rationale for a niche-specific surveillance platform, which might also be used to screen pre-pubertal children with higher BMIs and dense breasts for their susceptibility to breast cancer. Taken together, our present study provides a seamlessly efficient computational platform for tagging mammographic images of breast lumps into benign and malignant lesions with precision. This is in stark contrast to the black box nature and inherent instability of process-based approaches. In the current work, we included images of a specific orientation containing biopsy tagged lumps from a particular age group for training and classifying them as benign and malignant. In future deliberations we intend to use HVG-based time series analysis across age groups, in which the clinical multimedia images are mapped to visibility graphs describing the corresponding states and linking the successive states with the digital outputs from the other two modalities including breast cytology and breast cancer-associated gene expression patterns. Such a strategy will not only convert a dynamic time series into a time varying network but can also produce network of networks providing rich insight on the short term and long term predictions about the 'breast health' of Indian women. Our approach has potential applications of being scaled not only in the 742 districts across the Indian subcontinent but also in resource-limited healthcare ecosystems prevalent in other Low-and-Middle Income Countries (LMICs) across the world.

References

1. American Cancer Society. https://www.cancer.org/cancer/breast-cancer/about/what-is-breast-cancer.html
2. Ghosh A, Ramakant P, Ranjan P, Deshpande A, Janardhanan R (2021) EMD based binary classification of mammograms. In: Proceedings of 2nd international symposium on computer vision and machine intelligence in medical image analysis (ISCMM)
3. Shrivastav KD, Arambam P, Das AM, Saeed S, Kaul U, Ranjan P, Janardhanan R (2020) Earth mover's distance-based automated geometric visualization/classification of electrocardiogram signals. Trends Commun Cloud Big Data 75–85
4. Khatri A, Jain R, Vashista H, Mittal N, Ranjan P, Janardhanan R (2020) Pneumonia identification in chest X-ray images using EMD. Trends Commun Cloud Big Data 87–98
5. Shrivastav KD, Das AM, Singh H, Ranjan P, Janardhanan R (2018) Classification of colposcopic cervigrams using EMD in R. In: International symposium on signal processing and intelligent recognition systems. Springer, Singapore, pp 298–308
6. Ji H, Xu T, Wu W, Wang J (2016) Visibility graph analysis on EEG signal. In: 9th international congress on image and signal processing, biomedical engineering and informatics (CISP-BMEI), Datong, pp 1557–1561. https://doi.org/10.1109/CISP-BMEI.2016.7852963
7. Zhu G, Li Y, Wen P (2014) Analysis and classification of sleep stages based on difference visibility graphs from a single-channel EEG signal. IEEE J Biomed Health Inf 18(6):1813–1821
8. Speranza S, Sebastiano S, Lucas L, Daniele M (2017) Visibility graphs for fMRI data: multiplex temporal graphs and their modulations across resting-state networks. Netw Neurosci 1(3):208–221
9. Madl T (2016) Network analysis of heart beat intervals using horizontal visibility graphs. In: 2016 computing in cardiology conference (CinC). IEEE, pp 733–736
10. Zhou YW, Liu JL, Yu Z-G, Zhao NQ, Anh V (2014) Fractal and complex network analyses of protein molecular dynamics. Phys A 416:21–32

11. Acosta B, Pastén D, Moya P (2019) Reversibility of turbulent and non-collisional plasmas: solar wind. In: Proceedings IAU symposium, vol 15, pp 363–366
12. Gutin G, Mansour T, Severini S (2011) A characterization of horizontal visibility graphs and combinatorics on words. Phys A 390(12):2421–2428
13. Telesca L, Lovallo M (2012) Analysis of seismic sequences by using the method of visibility graph. Europhys Lett 97:50002p1–50002p4
14. Lacasa L, Luque B, Ballesteros F, Luque J, Nuno JC (2008) From time series to complex networks: the visibility graph. Proc Nat Acad Sci 105:4972–4975. pmid:18362361
15. Lacasa L, Nuñez A, Roldán É, Parrondo JMR, Luque B (2012) Time series irreversibility: a visibility graph approach. Eur Phys J B 85:217
16. Sahni P, Mittal N (2019) Breast cancer detection using image processing techniques. In: Kumar M, Pandey R, Kumar V (eds) Advances in interdisciplinary engineering. Lecture notes in mechanical engineering. Springer, Singapore. https://doi.org/10.1007/978-981-13-6577-5_79
17. Kumar A, Kumar P, Srivastava S (2022) A skewness reformed complex diffusion based unsharp masking for the restoration and enhancement of Poisson noise corrupted mammograms. Biomed Sig Process Control 73:103421
18. Song SE, Cho KR, Cho Y et al (2022) Machine learning with multiparametric breast MRI for prediction of Ki-67 and histologic grade in early-stage luminal breast cancer. Eur Radiol 32:853–863. https://doi.org/10.1007/s00330-021-08127-x
19. Soleimani H (2021) Information fusion of magnetic resonance images and mammographic scans for improved diagnostic management of breast cancer
20. Varghese BA, Lee S, Cen S et al (2022) Characterizing breast masses using an integrative framework of machine learning and CEUS-based radiomics. J Ultrasound. https://doi.org/10.1007/s40477-021-00651-2
21. Zhang X, Landsness EC, Chen W, Miao H, Tang M, Brier LM, Culver JP, Lee J-M, Anastasio MA (2022) Automated sleep state classification of wide-field calcium imaging data via multiplex visibility graphs and deep learning. J Neurosci Methods 366:109421
22. Zhang D, Long X, Lin X, Werth J, Wijshoff R, Aarts RM, Andriessen P (2021) Characterizing cardiorespiratory interaction in preterm infants across sleep states using visibility graph analysis. J Appl Physiol 130(4):1015–1024
23. Zervou MA, Doutsi E, Tsakalides P (2021) Visibility graph network of multidimensional time series data for protein structure classification. In: 2021 29th European signal processing conference (EUSIPCO), pp 1216–1220. https://doi.org/10.23919/EUSIPCO54536.2021.9616113
24. Makaram N, Karthick PA, Gopinath V, Swaminathan R (2021) Surface electromyography-based muscle fatigue analysis using binary and weighted visibility graph features. Fluctuation Noise Lett 20(02):2150016
25. Chen J, Wang Z, Zheng H, Liu L, Zhu Z, Ying S, Wei Y (2019) DGFE-VG: dynamic gene feature extraction via visibility graph. In: Proceedings of the third international conference on medical and health informatics, pp 119–124
26. Cui C, Li L, Cai H, Fan Z, Zhang L, Dan T, Li J, Wang J (2021) The Chinese mammography database (CMMD): an online mammography database with biopsy confirmed types for machine diagnosis of breast. Cancer Imaging Arch. https://doi.org/10.7937/tcia.eqde-4b16
27. Ghosh A, Ramakant P, Ranjan P (2021) EMD based binary classification of mammograms with novel leader selection technique. In: 2021 IEEE 2nd international conference on applied electromagnetics, signal processing, and communication (AESPC), pp 1–6. https://doi.org/10.1109/AESPC52704.2021.9708458
28. Tufail AB, Ma YK, Zhang QN (2020) Binary classification of Alzheimer's disease using sMRI imaging modality and deep learning. J Digit Imaging 33:1073–1090. https://doi.org/10.1007/s10278-019-00265-5

An Intelligent Technique for Detecting Diabetic Retinopathy by Comparative Analysis Based on Deep Learning

Hrushikesh Shukla and Siddhivinayak Kulkarni

Abstract Diabetic retinopathy is an eye disorder that affects people who have diabetes. Diabetic retinopathy occurs when high blood sugar levels harm the blood vessels in the eye. These blood vessels tend to swell and leak. They also can close, preventing blood from flowing through. Irregular new blood vessels are formed on the retina and eyesight becomes weak and blurred. Diabetic retinopathy can affect people with any type of diabetes (type 1, type 2, or gestational diabetes). Therefore, it is vital to detect diabetes retinopathy at an early stage. In this paper, we outlined the idea of diabetic retinopathy and highlighted the relevance of its methodologies, along with an exhaustive comparative study that helped us to shortlist the best performing model.

Keywords Diabetic retinopathy detection · CNN · Image preprocessing · Deep learning · Fundus images

1 Introduction

Diabetic retinopathy is an eye disease that induces blood vessel changes in the retina. The retina is the back portion of human eye that transforms light into images. Blood vessels may swell, burst, or bleed, often leading to changes in vision or blindness. Generally, both eyes are affected. Diabetic retinopathy can scare and harm the retinal system if left untreated. The primary cause of vision loss for diabetes patients is diabetic retinopathy. This is the primary cause of blindness in all adults in the United States [1]. Anyone with diabetes is at high risk of developing diabetic retinopathy. Hence, early detection of diabetic retinopathy is very important.

H. Shukla (✉) · S. Kulkarni
School of Computer Engineering and Technology, Dr. Vishwanath Karad MIT World Peace University, Pune 411038, India
e-mail: shukla123hrushikesh@gmail.com

S. Kulkarni
e-mail: siddhivinayak.kulkarni@mitwpu.edu.in

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_28

We summarized the definition of diabetic retinopathy and highlighted the significance of its methodologies in this paper. We looked at several articles for this report. The majority of the methodologies we aimed at diagnosing diabetic retinopathy as well as its severity. Some typical features of retinal scans when the disease is detected include hemorrhages, exudates, microaneurysms, and cotton wool spots. We have also proposed a methodology and compared our results that helped us shortlist the best performing model.

The rest of the article is as follows: We address earlier studies in Sect. 2. Then, in Sect. 3, we propose an architecture to detect the presence followed by the results in Sect. 4 and conclusions in Sect. 5.

2 Literature Review

In any system based on machine learning, the three crucial steps involved are data preprocessing, developing a model, and evaluating the model across the test dataset. The following subsections discuss the methodologies that illustrate the above stages of machine learning-based model creation.

2.1 Data Preprocessing

The data is cleaned preprocessed before providing it as an input to the deep learning model. This increases the algorithm's efficiency while also preventing the occurrence of undesirable bias and variance. Kauppi et al. presented a dataset and ground reality for the diagnosis of diabetic retinopathy, as well as assessment protocols. In this article, the authors also clearly stated the difference between abnormalities present in the eye when diagnosed with diabetic retinopathy like hemorrhages, exudates, and others. According to the authors, the receiver operating curve (ROC), and weighted error rate (WER) are the best output metrics for the classification role. They also defined some baseline performances [2]. Sinthanayothin et al. also worked on an unique dataset. They collected their dataset of 112 digital fundal images for detection of non-proliferative diabetic retinopathy (NPDR). Hemorrhages, microaneurysms, and hard exudates were all isolated for this reason. Their technique involves converting RGB images to hue saturation, intensity (HSI) images and using dynamic, local, and contrast strategies to improve picture quality. The authors then applied recursive region growing and moat operator algorithms for better isolation of the features [3]. The growth of irregular vessels is a common indication of Proliferative Diabetic Retinopathy (PDF). Using two separate line detection methods, Welikala et al. proposed a methodology for detecting these unusual vessels. The first line segmentation technique employs a standard line operator, while the second uses a novel line operator to detect false non-vessel edges. Isolating the green channel from the images, noise reduction, and adaptive histogram normalization techniques are also part of the image preprocessing. Furthermore, the SVM classification algorithm was applied to both methods independently, and then the findings were combined [4]. In [5], the authors devised a method for segmenting exudates from fundus photos that would aid in the diagnosis of diabetic retinopathy. Their approach entails converting RGB images to HSI images, then applying median filtering and contrast limited adaptive histogram equalization (CLAHE) to reduce noise and improve image contrast. They also used morphological reconstruction to detect the exudates after removing the optical disc from the images. Human experts then checked the detected exudate images to assess the system's accuracy.

In [6], the authors proposed a method for automated diabetic retinopathy screening using images from a retinal scan. The preprocessing entails using a limited adaptive contrast enhancement technique to optimize the intensity of the images before detecting the retinal images' usual structures. The bright exudates were isolated using recursive region growing (RRG), adaptive intensity thresholding (AIT). The dark exudates were recognized using the moat operator. Furthermore, a neural network model was used to classify whether exudate features were typical or atypical. Yun et al. proposed an approach to detect diabetic retinopathy using artificial neural networks. They applied histogram equalization for enhancement of contrast, while exudates were easily discriminated against using morphological operators. Thresholding followed these procedures to extract features from the images after which a neural network model was used to classify the images into their respective classes [7]. Palavalasa and Sambaturu developed a unique approach to identify the hard exudates from fundus images. The images were first preprocessed by applying CLAHE transformation. Further preprocessing includes detecting the possible candidates by subtracting the background from the photographs. Then the false candidates were recognized and removed from the images using the technique based on the decorrelation stretch. The mentioned method proved efficient in isolation of the exudates [8]. In [9], the authors conducted a comparative study on retinal fundus images using various preprocessing techniques such as histogram equalization (HE), contrast limited adaptive histogram equalization, and so on (CLAHE). They compared the performance of only resized images, resized images with HE transformation, and resized images with CLAHE transformation. The results revealed that pictures preprocessed with the CLAHE transformation generated the best results. In [10], the authors used three significant measures to remove characteristics such as blood vessels, microaneurysms, and stiff exudates. The data points were classified using the Support Vector Machine (SVM) algorithm. SVM consistently outperforms other deep learning techniques.

2.2 Development of Deep Learning Models

In addition to data preprocessing, it is necessary to develop the correct deep learning model and optimum hyperparameters to ensure that the predictions are accurate. To detect diabetic retinopathy, Quellec et al. [11] developed a deep learning-based

ConvNet architecture. ConvNets can produce heatmaps that can help isolate areas that cause diabetic retinopathy, such as hemorrhages. They used the Kaggle diabetic retinopathy (DR) dataset to train their model and the DiaretDB1 dataset to assess its performance. The multiclass classification was used by Pratt et al. [12] to detect various stages of diabetic retinopathy. For improved efficiency, preprocessing techniques such as color normalization, augmentation, and resizing were applied to the fundus images. The authors have used a convolutional neural network they built to detect the stage of diabetic retinopathy with a 75% accuracy and 95% specificity. Gargeya and Leng developed a fully automatic DR detection algorithm for red, green, and blue fundus images using a deep learning approach. They scaled the image pixel values to values between 0 and 1. The model was validated using 5-fold stratified cross-validation on the local data collection. This test procedure has trained 5 different models, each holding a separate validation bucket of approximately 15,000 images. Average metrics were obtained from 5 test runs based on the respective held-out data by comparing the model's predictions with the gold standard defined by the panel of experts [13]. In [14] found image preprocessing using a variety of filtering mechanisms to improve the image's features. Another approach discussed in this study was the extraction of statistical features from images. Both CNN and Deep Neural Networks (DNN) models are effective in terms of pictures, as CNN's CPU training time is affected in the study. In this case, DNN outperforms CNN in terms of training accuracy as well as validation accuracy.

Xu et al. concentrated on classifying retinal images into typical images and diabetic retinopathy images. By using a label-preserving transformation, data augmentation was used to artificially expand the images. On the marked dataset, they applied translation, stretching, rotation, and flipping. For classification, the authors used a convolutional neural network [15]. In [16], the authors used the most efficient CNN architectures to detect diabetic retinopathy. The image preprocessing included the normalization of the saturation values of each figure, as well as the standardization of measurements and the removal of noise. They evaluated various neural networks based on convolution. VGG16noFC2 gave them the highest accuracy possible. Chetoui and Akhloufi used images from retinal scans to conduct multiclass classification on the stages of diabetic retinopathy. They decreased the size of some pictures and omitted others that were of poor quality. They used EfficientNet-B7 with an optimized layer of Global Average Pooling (GAP) for classification. To improve performance, they tested their architecture on two datasets [17]. In [18], the authors proposed a deep learning-based approach for detecting diabetic retinopathy. They implemented this system on two different publicly available datasets. Authors first graded the images by using medical professionals and then applied transfer learning mechanisms for diagnosing whether the patient is suffering from diabetic retinopathy or not. The outputs of ten neural network models were ensembled, and performance was measured using the metrics like sensitivity, specificity, and area under the receiver operator characteristic curve.

Chaturvedi et al. proposed a technique for early stage identification of five-grade severity for diabetic retinopathy. They used ImageDataGenerator to perform a 0.15 random zoom and then scale the images down. They used a Dense Convolutional

Network. As the activation function for nonlinearity, they used sigmoid [19]. In [20], the authors implemented image augmentation for preprocessing. They investigated AlexNet , ResNet18, SqueezeNet. They finalized CNN models like AlexNet VGG16, Res Net18, Squeeze Net, VGG19, Google Net. In [21], the authors resampled images to 819 * 614 and normalized them later. They used ResNet50 as a backbone network. Their architecture is made up of 3 layers. They chose ResNet50 for the outer layer, RexNext for the intermediate layer, and split attention for the inner layer, which allows for channel-wide reweighing and actively adjusting filters in various channels, enabling the network to change the size of the acceptance area dynamically. Patil and Kulkarni used a transfer learning approach to identify the type of diabetic retinopathy. The main preprocessing techniques used are image augmentation and resizing. The authors then used the pre-trained InceptionNet3 in connection with global average pooling layers and a series of fully connected layers to determine the image category [22].

2.3 Exceptional Techniques

As compared to a deep learning approach, some algorithms tuned for the isolation and classification of diabetic retinopathy may perform better. Shankar et al. proposed a method to classify fundus images into various groups of diabetic retinopathy using synergic deep learning. The images are preprocessed by eliminating noise and applying histogram equalization to enhance their efficiency. Further, they applied a synergic deep learning model based on ResNet and evaluated the model's performance across the metrics like accuracy, sensitivity, and specificity [23]. In order to automatedly identify fundus images into two groups with or without an RDR in [24], the authors suggested a deep-seated learning technique based on the diagnostic approach of human ophthalmologists. They scaled down the image and converted the pixel values for pre processing. They used augmentation as well. To extract features, InceptionV3 is used. A novel convolutional neural network with Siamese-like architecture is proposed. In [25], the authors proposed a deep learning enhanced algorithm for automated DR detection to significantly improve performance. IDx-DR X2.1 is an automatic DR detection system. It consists of two parts: client software that runs at the point of care and analytics software that runs on IDx's server and is reviewed by the authors.

Tymchenko et al. suggested a multistaged architecture for multiclass classification of the stages of diabetic retinopathy. Image augmentation, applying Gaussian blur were the steps taken to preprocess the images. They also used a multistaged deep learning architecture to classify the data. The outputs of convolutional neural networks that used global average and global max-pooling were fed into dense layers, which extracted additional features. To perform classification, the predictions of these fully connected layers were provided to three distinct models. The first model classified the data using one-hot encoding and treated the classes as nominal. The second model treated the labels numerically and generated numerical results, while the third model treated the categories ordinally and created ordinal outputs. By combining the outcomes of all of these models, the final prediction was made [26]. The authors proposed a novel and fast method for detecting diabetic retinopathy using a smartphone and a portable fundus camera in [27]. This system makes it simple to collect fundus images and identify the stage of the disease as well as the existence of defects in the eye. The authors used the above technique to collect fundus images from several patients in India. The photos were then graded and identified by clinical experts before being sent to an international agency that used deep learning models to detect the categories. The models' results revealed that image collection from a smartphone was almost as effective as actual retinal scans. In [28] the authors used all three color channels and cropped and resized all of the images to 512 * 512. They also use affine transformations to improve the training dataset, such as random zooming, random translations, and random rotations. They employed Bayesian Neural Networks. Kassani et al. used Xception as a backbone for deep feature extraction. The inception module, depth-wise separable convolution layers, and residual blocks make up their architecture. They adopted L1 and L2 regularization to help solve the problem of overfitting [29]. The authors cropped images and resampled them to 640 * 640 in [30]. They proposed the EyeCheck algorithm. In this algorithm, all pixels that appear to be in a red lesion are detected using a pixel feature classification-based algorithm. Extracted features were processed with a KNN classifier to assign a probability and show the likelihood that the lesion is red.

3 Proposed Methodology

This section discusses the flow of the proposed system to detect diabetic retinopathy. Figure 1 shows the flow of the system. The system receives retinal scan pictures and conducts preprocessing on them. Image resizing, channel selection, and Cluster Limited Adaptive Histogram Equalization (CLAHE) are all used in this process. After that, the improved images are sent to a Convolutional Neural Network (CNN)-based model for classification, and the results are recorded.

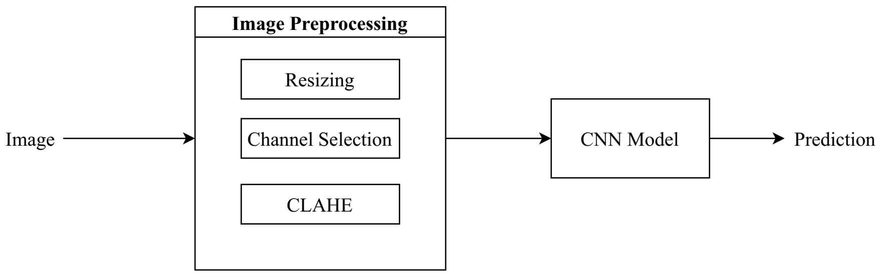
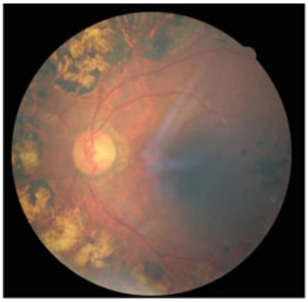


Fig. 1 The architecture for detection of diabetic retinopathy

(a) Normal Fundus Image



(b) DR Infected Fundus Image

Fig. 2 Sample fundus images

The system showcased in Fig. 1 accepts fundus images as input. The image was captured with a fundus camera, which uses light reflection to create a two-dimensional image of the human retina. Diabetic retinopathy can be detected with this retinal scan image, also known as a fundus image (DR). Figure 2 shows the contrast between normal and DR-infected fundus images. The photographs are then transferred to the image preprocessing area to be further enhanced.

In the image preprocessing section, we are applying three distinct techniques namely image resizing, channel selection, and CLAHE. The input images are color images and contain three channels red, green, and blue. The size of input images may also vary depending on a variety of factors. Hence, we applied an image resizing technique that reduces the size of the picture to (224×224) pixels.

Further, we calculate mean pixel intensity (MPI) per channel as the ratio sum of pixel intensities to the total number of pixels in the layer. We select the optimal channel for the classification based on the calculated MPI. Then the shortlisted layer is enhanced by applying CLAHE operation. The histogram equalization technique has drawbacks like over brightness that may reduce the image quality. CLAHE transformation, an adaptive histogram equalization approach that clips the pixel bins above a specified threshold and distributes those pixels throughout the other bins, can be used to overcome these disadvantages.We further used a combination of CNN and Artificial Neural Networks (ANN) for detecting whether the eye is normal or infected by DR. The system gives output in the range of 0 to 1. The image is infected with DR if the result is near 1, otherwise, it is normal.

4 Comparative Study

The following section details the comprehensive comparison investigation we conducted to diagnose diabetic retinopathy. Figure 3 depicts the flow of the comparison investigation. In this comparative study, we used an open-source dataset and performed multiple combinations of image preprocessing operations. After these photos

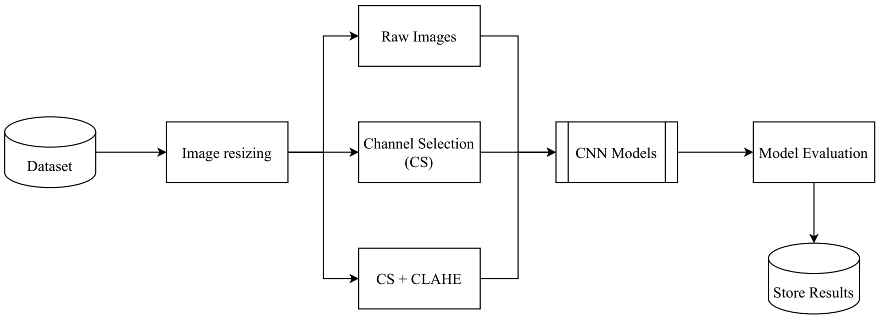


Fig. 3 Setup of comparative study

were provided as input to several transfer learning models, the performance of each model for each of the picture preprocessing techniques was measured and saved. The upcoming subsections discuss about all these components in detail.

4.1 Dataset and Image preprocessing

For this study, we decided to focus on the recently published APTOS2019 dataset available on kaggle [31]. This dataset contains color photos of normal and diabetic retinopathy-infected eyes in five separate classes. There are 1805 photos of normal class patients and 1857 images of DR patients. We applied multiple preprocessing techniques like image resizing, channel selection (CS), and Contrast Limited Adaptive Histogram Equalization (CLAHE) techniques. We created three alternative sets of the same dataset for our comparative analysis. All of the images in the collections were resized. We called the first set of pictures Raw as they were not been altered in any way. The photos after applying CS were in the second set, while the images after performing both CS and CLAHE techniques were in the third. After that, the CNN models were given to each of the collections to evaluate.

4.2 CNN Models and Training

The comparative study focuses majorly on transfer learning and the effect of different image preprocessing methods. We employed nine distinct pre-trained CNN models and replaced the fully connected layers with the ones we developed for the classification. We trained each of the models for 15 epochs and used made use of max-pooling layers. For optimization of the neural network, we employed adam optimizer while the loss of the model was evaluated using the binary cross-entropy function. The fully connected layers were similar across all the nine models, and for every model, we trained and compared the results across all the three collections of the dataset that were generated in the previous section.

4.3 Model Evaluation

The predictions by the models are evaluated across multiple performance metrics like accuracy, precision, recall, $f1$ score, Cohen's Kappa score and AUC. Further these metrics are recorded and stored for comparative analysis.

Accuracy is the ratio of correctly predicted samples to the total samples used for testing. The mathematical expression to derive accuracy from values of confusion matrix are is given as Eq. 1

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

Where the number of true-positive samples is TP, the number of true negative samples is TN, and the number of false-positive and false-negative samples is FP and FN, respectively. The proportion of relevant instances to the total number of instances chosen is known as precision. The recall is defined as the proportion of relevant instances chosen to the total number of relevant instances. The formulae for measuring precision and recall are illustrated in Eqs. 2 and 3.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

The $F1$ score is the harmonic mean of precision and recall. The $F1$ Score lies in the range of [0 to 1]. The mathematical formula to calculate the $F1$ score is shown in Eq. 4.

$$F1\,\text{Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

AU-ROC is also widely used to assess the classifier's efficiency. Receiver operating curve plots true-positive and false-positive rates to show the efficiency of classification models at various thresholds. The classifier is perfect if the AU-ROC is 1. The Cohens Kappa score is the score of agreement between the two raters. This score lies in the range of $[-1 \text{ to } 1]$. If the score is 1 then the two raters are in complete agreement while the score of -1 signifies that the two raters are in complete disagreement.

Table 1 Experimental results

S. No.	Algorithms	Parameters	Raw images	Processed images (CS)	Processed images (CS + CLAHE)
1	DenseNet169	Accuracy	0.969	0.978	0.98
		Cohen's kappa	0.937	0.956	0.959
		Precision	0.969	0.973	0.975
		Recall	0.967	0.983	0.983
		AUC	0.995	0.994	0.995
		$F1$ score	0.968	0.978	0.979
2	VGG19	Accuracy	0.945	0.929	0.956
		Cohen's kappa	0.891	0.858	0.913
		Precision	0.952	0.935	0.951
		Recall	0.936	0.92	0.961
		AUC	0.983	0.981	0.987
		$F1$ score	0.944	0.927	0.956
3	VGG16	Accuracy	0.954	0.941	0.962
		Cohen's kappa	0.907	0.883	0.923
		Precision	0.958	0.93	0.959
		Recall	0.947	0.953	0.964
		AUC	0.987	0.983	0.99
		$F1$ score	0.953	0.941	0.961
4	ResNet50	Accuracy	0.926	0.896	0.937
		Cohen's kappa	0.853	0.792	0.874
		Precision	0.887	0.886	0.929
		Recall	0.975	0.906	0.945
		AUC	0.974	0.955	0.976
		F1 score	0.929	0.896	0.937
5	EfficientNetB0	Accuracy	0.507	0.507	0.507
		Cohen's kappa	0	0	0
		Precision	0	0	0
		Recall	0	0	0
		AUC	0.339	0.402	0.296
		$F1$ score	0	0	0
6	EfficientNetB7	Accuracy	0.507	0.507	0.507
		Cohen's kappa	0	0	0
		Precision	0	0	0
		Recall	0	0	0
		AUC	0.5	0.348	0.51
		$F1$ score	0	0	0
7	XceptionNet	Accuracy	0.967	0.97	0.958
		Cohen's kappa	0.934	0.94	0.915
		Precision	0.98	0.954	0.946
		Recall	0.953	0.986	0.97
		AUC	0.991	0.993	0.991
		$F1$ score	0.966	0.97	0.958

(continued)

Table 1 (continued)

S. No.	Algorithms	Parameters	Raw images	Processed images (CS)	Processed images (CS+CLAHE)
8	InceptionV3	Accuracy	0.955	0.956	0.949
		Cohen's kappa	0.91	0.913	0.899
		Precision	0.946	0.951	0.938
		Recall	0.964	0.961	0.961
		AUC	0.988	0.993	0.989
		$F1$ score	0.955	0.956	0.949
9	InceptionResNetV2	Accuracy	0.955	0.969	0.963
		Cohen's kappa	0.91	0.937	0.926
		Precision	0.946	0.977	0.972
		Recall	0.964	0.958	0.953
		AUC	0.986	0.991	0.989
		$F1$ score	0.955	0.968	0.962

4.4 Results

Table 1 displays the findings of the comparison investigation. The table shows which of the nine models were used, including VGG19, XceptionNet, and others. Each of these models was given the dataset with varying levels of preprocessing, and the findings are published. The table clearly shows that the models' performance has improved after employing preprocessing techniques. On preprocessed pictures with CS and CLAHE, DenseNet169 had the best performance. InceptionV3 and InceptionResNetV2 came after it. EfficientNet B0 and EfficientNet B7, on the other hand, fared poorly. As a consequence, we can say that DenseNet169 is the best option for designing a system to detect Diabetic Retinopathy. Figure 4 shows the accuracy score achieved by the transfer learning models on raw images while Figs. 5, and 6 shows the model accuracies on CS and CS with CLAHE processed images. The graphs depict that the DenseNet169 gave the best accuracy followed by XceptionNet and InceptionResNetV2. The models like EfficientNetB0 and EfficientNetB7 performed poorly.

5 Conclusion

In this paper, several research papers are surveyed for the detection of diabetic retinopathy and developed a novel technique to detect diabetic retinopathy using a deep learning approach. An exhaustive comparative study is performed that depicts the deployment of the CLAHE technique to increase the overall performance. In this comparative study, we compared the performance of nine different transfer learning models and found out that the best performance with an accuracy of 98% is achieved

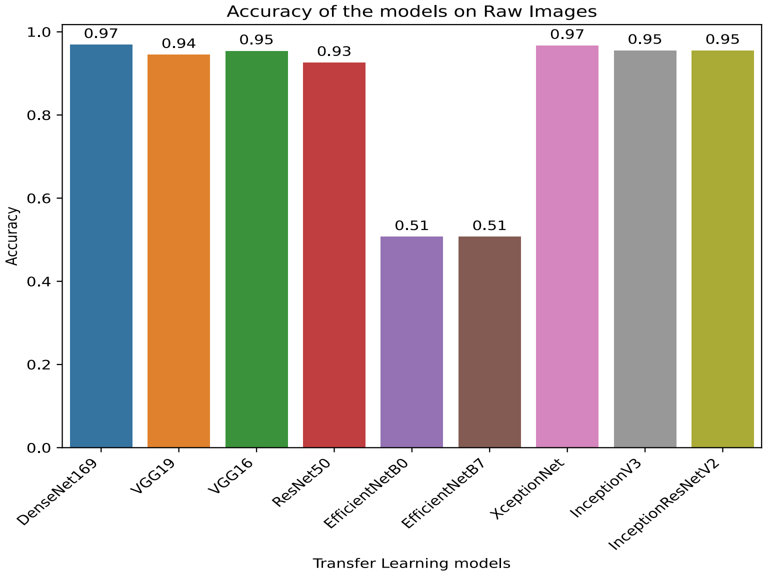


Fig. 4 Accuracy score on raw images

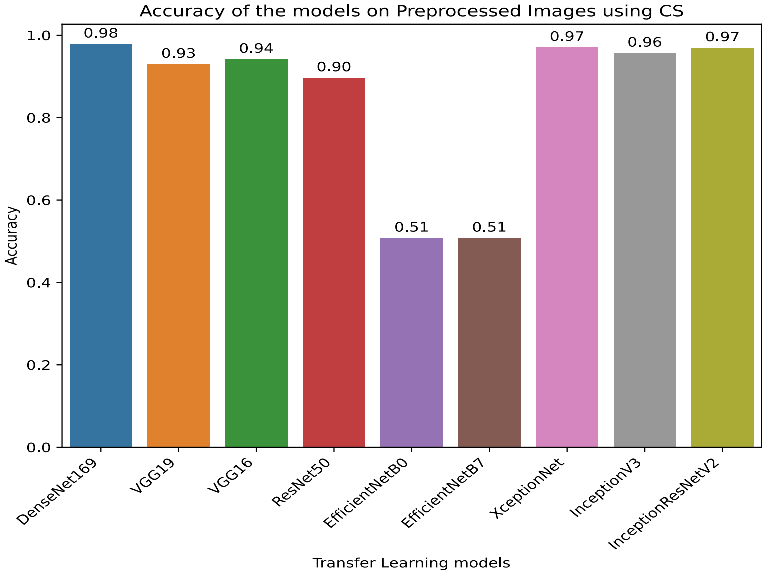


Fig. 5 Accuracy score on images with CS transformation

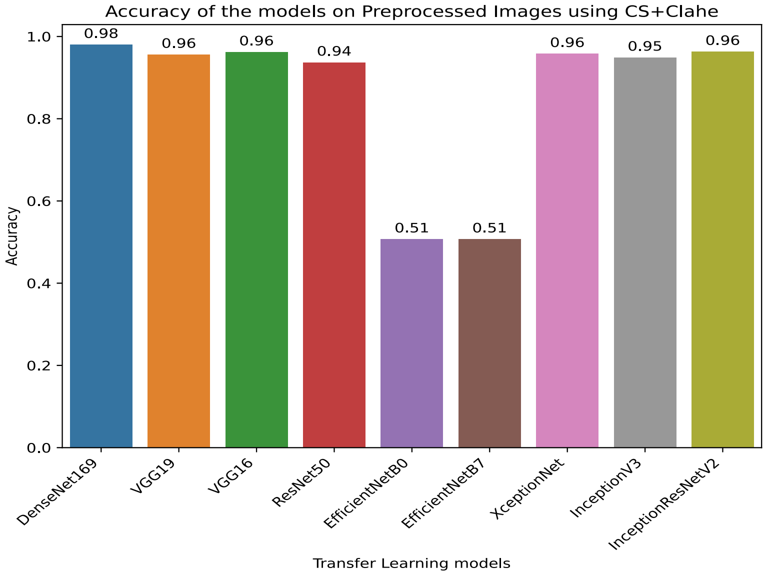


Fig. 6 Accuracy score on images with CS + CLAHE transformation

by using DenseNet169. The future goals of this project include building a system that could detect multiple ocular diseases and detect the severity of diabetic retinopathy disease.

References

1. Diabetic retinopathy: how to prevent vision loss. https://www.webmd.com/diabetes/diabetic-retinopathy. Accessed: 2021-03-28
2. Kälviäinen R, Uusitalo H (2007) Diaretdb1 diabetic retinopathy database and evaluation protocol. In: Medical image understanding and analysis, vol 2007, p 61
3. Sinthanayothin C, Boyce JF, Williamson TH, Cook HL, Mensah E, Lal S, Usher D (2002) Automated detection of diabetic retinopathy on digital fundus images. Diabet Med 19(2):105–112
4. Welikala R, Dehmeshki J, Hoppe A, Tah V, Mann S, Williamson TH, Barman S (2014) Automated detection of proliferative diabetic retinopathy using a modified line operator and dual classification. Comput Methods Programs Biomed 114(3):247–261
5. Sopharak A, Uyyanonvara B, Barman S, Williamson TH (2008) Automatic detection of diabetic retinopathy exudates from non-dilated retinal images using mathematical morphology methods. Comput Med Imaging Graph 32(8):720–727
6. Usher D, Dumskyj M, Himaga M, Williamson TH, Nussey S, Boyce J (2004) Automated detection of diabetic retinopathy in digital retinal images: a tool for diabetic retinopathy screening. Diab Med 21(1):84–90
7. Yun WL, Acharya UR, Venkatesh YV, Chee C, Min LC, Ng EYK (2008) Identification of different stages of diabetic retinopathy using retinal optical images. Inf Sci 178(1):106–121

8. Palavalasa KK, Sambaturu B (2018) Automatic diabetic retinopathy detection using digital image processing. In: 2018 international conference on communication and signal processing (ICCSP). IEEE, pp 0072–0076
9. Hemanth DJ, Deperlioglu O, Kose U (2020) An enhanced diabetic retinopathy detection and classification approach using deep convolutional neural network. Neural Comput Appl 32(3):707–721
10. Carrera EV, González A, Carrera R (2017) Automated detection of diabetic retinopathy using svm. In: 2017 IEEE XXIV international conference on electronics, electrical engineering and computing (INTERCON). IEEE, pp 1–4
11. Quellec G, Charrière K, Boudi Y, Cochener B, Lamard M (2017) Deep image mining for diabetic retinopathy screening. Med Image Anal 39:178–193
12. Pratt H, Coenen F, Broadbent DM, Harding SP, Zheng Y (2016) Convolutional neural networks for diabetic retinopathy. Procedia Comput Sci 90:200–205
13. Gargeya R, Leng T (2017) Automated identification of diabetic retinopathy using deep learning. Ophthalmology 124(7):962–969
14. Dutta S, Manideep B, Basha SM, Caytiles RD, Iyengar N (2018) Classification of diabetic retinopathy images by using deep learning models. Int J Grid Distrib Comput 11(1):89–106
15. Xu K, Feng D, Mi H (2017) Deep convolutional neural network-based early automated detection of diabetic retinopathy using fundus image. Molecules 22(12):2054
16. García G, Gallardo J, Mauricio A, López J, Del Carpio C (2017) Detection of diabetic retinopathy based on a convolutional neural network using retinal fundus images. In: International conference on artificial neural networks. Springer, Berlin, pp 635–642
17. Chetoui M, Akhloufi MA (2020) Explainable diabetic retinopathy using efficientnet. In: 2020 42nd annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, pp 1966–1969
18. Gulshan V, Peng L, Coram M, Stumpe MC, Wu D, Narayanaswamy A, Venugopalan S, Widner K, Madams T, Cuadros J et al (2016) Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. JAMA 316(22):2402–2410
19. Chaturvedi SS, Gupta K, Ninawe V, Prasad PS (2020) Automated diabetic retinopathy grading using deep convolutional neural network. arXiv preprint arXiv:2004.06334
20. Khalifa NEM, Loey M, Taha MHN, Mohamed HNET (2019) Deep transfer learning models for medical diabetic retinopathy detection. Acta Inf Med 27(5):327
21. Naseem U, Khushi M, Khan SK, Waheed N, Mir A, Qazi A, Alshammari B, Poon SK (2020) Diabetic retinopathy detection using multi-layer neural networks and split attention with focal loss. In: International conference on neural information processing. Springer, Berlin, pp 26–37
22. Patil S, Kulkarni P (2020) Machine learning technique for feature extraction and segmentation of retinal blood vessels. Int J Recent Technol Eng 9(1):952–955. https://doi.org/10.35940/ijrte.a2223.059120
23. Shankar K, Sait ARW, Gupta D, Lakshmanaprabu S, Khanna A, Pandey HM (2020) Automated detection and classification of fundus diabetic retinopathy images using synergic deep learning model. Pattern Recogn Lett 133:210–216
24. Zeng X, Chen H, Luo Y, Ye W (2019) Automated diabetic retinopathy detection based on binocular siamese-like convolutional neural network. IEEE Access 7:30744–30753
25. Abràmoff MD, Lou Y, Erginay A, Clarida W, Amelon R, Folk JC, Niemeijer M (2016) Improved automated detection of diabetic retinopathy on a publicly available dataset through integration of deep learning. Invest Ophthalmol Visual Sci 57(13):5200–5206
26. Tymchenko B, Marchenko P, Spodarets D (2020) Deep learning approach to diabetic retinopathy detection. arXiv preprint arXiv:2003.02261
27. Rajalakshmi R, Subashini R, Anjana RM, Mohan V (2018) Automated diabetic retinopathy detection in smartphone-based fundus photography using artificial intelligence. Eye 32(6):1138–1144
28. Filos A, Farquhar S, Gomez AN, Rudner TG, Kenton Z, Smith L, Alizadeh M, de Kroon A, Gal Y (2019) A systematic comparison of Bayesian deep learning robustness in diabetic retinopathy tasks. arXiv preprint arXiv:1912.10481

29. Kassani SH, Kassani PH, Khazaeinezhad R, Wesolowski MJ, Schneider KA, Deters R (2019) Diabetic retinopathy classification using a modified xception architecture. In: 2019 IEEE international symposium on signal processing and information technology (ISSPIT). IEEE, pp 1–6
30. Abràmoff MD, Reinhardt JM, Russell SR, Folk JC, Mahajan VB, Niemeijer M, Quellec G (2010) Automated early detection of diabetic retinopathy. Ophthalmology 117(6):1147–1154
31. Your machine learning and data science community. https://www.kaggle.com/

AI-Based Health Management System

Swadhin Nagulpelli, Akash Chavan, Aniket Kandalkar, and Smita Kulkarni

Abstract Hospital management system (HMS) has become a network with abundant storage, allowing different types of systems to share their health records within the hospital. Earlier, it was difficult to access these records and maintain a proper database of all the patients visiting the hospital, but now, with the help of HMS, the doctors get all the records of the patient before the patient entered the cabin, and the time of both patients and doctor is saved. This paper aims to design a dashboard for the hospital using HTML and CSS and Python Django. Python Django is used for maintaining the database and routing of the HTML pages. This dashboard provides automation to various activities within the hospital and simplifies the workflow of healthcare professionals and their interaction with the patients and the doctors. With the help of machine learning (ML), different models like XGBoost, max voting, and KNN have been compared and deployed to predict various diseases to save time, and also to analyze further data on HMS, which can be a significant reference for doctors to provide proper advice to their patients.

Keywords Hospital management system (HMS) · Dashboard · Python Django · Machine learning · Doctor · Patient

1 Introduction

In the modern world, a smarter and more intelligent monitoring system is important in the healthcare system. A hospital management system (HMS) plays a vital role in hospitals by automating the daily tasks like registering and discharging a new patient, appointments with the doctor, generating the bill, and maintaining proper communication and coordination between hospital staff, doctors, and patients. The main goal of HMS is to promote the smooth functioning of hospital administrations and transactions between the doctor and patient both in rural and urban hospitals. According to the practical workflow of the hospital, a user-friendly interface has been

S. Nagulpelli (✉) · A. Chavan · A. Kandalkar · S. Kulkarni
MIT Academy of Engineering, Alandi, Pune, India
e-mail: sunagulpelli@mitaoe.ac.in

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_29

designed for patients to enter their details and to collect the health parameters of the patients. Through HMS, this data are taken before the patient reaches the cabin of the doctor. This saves a lot of time for both doctor and patient. Later, this data are stored in the database of the hospital which can be used for further analysis.

Diabetes is widely prevalent, and early detection of this disease is crucial. Otherwise, the patient can suffer various problems, including permanent loss of sight. A person is diagnosed with diabetes when their body doesn't produce enough insulin. Most diabetic patients have either type 1 or type 2 of the disease. The bodies of type 1 diabetics do not produce enough insulin, while the bodies of type 2 diabetics do not respond to insulin when compared to a normal person. Here, machine learning models will ease the workload of doctors since the number of diabetic patients is steadily increasing. In this paper, different machine learning models have been deployed on the hospital's dashboard to predict whether the patient is suffering from any certain disease like diabetes and heart disease, and these models predict diseases by using certain health parameters of the patients. The system is user-friendly and can be accessible by admin, doctors, and patients where all the data will be safe.

The paper is organized as follows: In Sect. 2 review of the related research work, Sect. 3 describes the proposed methodology for the development of the hospital management system (HMS); Sect. 3.1 describes the overall working of the dashboard; Sect. 3.2 includes the machine learning models used to predict diseases and their comparison for better accuracy, and Sects. 4 and 5 includes the final result table and discussion and conclusion of the paper.

2 Literature Review

Healthcare systems make use of different technologies like IoT, RFID, and edge devices for monitoring the real-time health conditions of the patients and for the smooth functioning of the day-to-day activities of the hospitals. The database of the patients visiting the hospital is helpful for the data analysis for various data scientists. In paper [1], heterogeneous data have been integrated and made further data analysis on HMS using the physical data collected by regular check-ups for improving doctors' advice for patients. In paper [2], a model was presented that uses edge devices to automate the problems of patients' vital data collection, processing, and delivery. In paper [3], they proposed RFID-based conceptual framework for a smart hospital management system that provides safety and security of patient data management systems. This RFID technology is used for hospital supply chain management, tracking of movable devices, medication, staff duties, and attendance. In paper [4], they introduced a system that measures patients' normal parameters like ECG, respiratory airflow, and SpO2 and sends these extracted data to a remote server. This data are being observed and analyzed by specialists and doctors in real-time to monitor the patient's state. In paper [5], a hospital management information system (HIS) is developed which provides data for directors, tracks the workload for the medical workers, and improves the workers' efficiency. In paper [6], the system was developed to build an adequate system for hospitals to serve critical patients with a real-time feedback method. So, in this project, they have used machine learning models to predict the condition of the patients, and if the condition of the patient deteriorates, then an SMS will be sent to the doctor immediately. In paper [7], there is an application that includes a Chatbot, big data, and sensor tracking devices to monitor and process the health data of the patients. This application connects various hospitals so that the user can choose the set of hospitals for their consultation. In paper [8], they have used IoT devices to measure the temperature and heart rate of patients and store it in a database.

All the technologies discussed above revolve around the same fact of developing an efficient and reliable system for the smooth functioning of the regular processes and the workflow of the hospitals. This paper proposes a system that will help both patients and doctors and even the hospital's staff members for better functioning of the hospital management system.

3 Methodology

As per the requirement of the hospital's workflow, a user-friendly Web site has been designed to collect the health parameters of the patients, book an appointment, and predict whether the person is diabetic or not.

The Web site has been created using the Python Django framework. It is Python's Web-based framework to build Web applications. This framework uses a Python programming language to link the pages with each other. The front-end part is designed using HTML and CSS, and the backend part is done using Python Django. The framework also uses Python language to manage the database.

There are various Python files generated by default when we install the Django library. The manage.py file is responsible for running the Web application on the local server and also to find errors in the code. The urls.py selects a view whenever there is an incoming request URL. The views.py decides whether to send the user to a different page or not. The settings.py file consists of all the information regarding which type of database language is used, and it also contains information of the path of the templates folder inside which all the HTML and CSS files are stored. The models.py file is responsible for storing the data in the database.

A '.SAV' file needs to be created to deploy the machine learning model on the Web site. '.SAV' file is a saved progress file that saves the trained ML models in binary format and then predicts the outcome of any new input data using the trained ML model. The Web site runs on the local server, generally on port 8000.

Figure 1 shows the overall flow of the system how the patient will log in till he gets the final prescription and overall generated bill of his treatment. The doctor will control the request of appointments and accordingly send the prescription to the patient within a few minutes. The admin can control the overall flow of the system's interface and database.

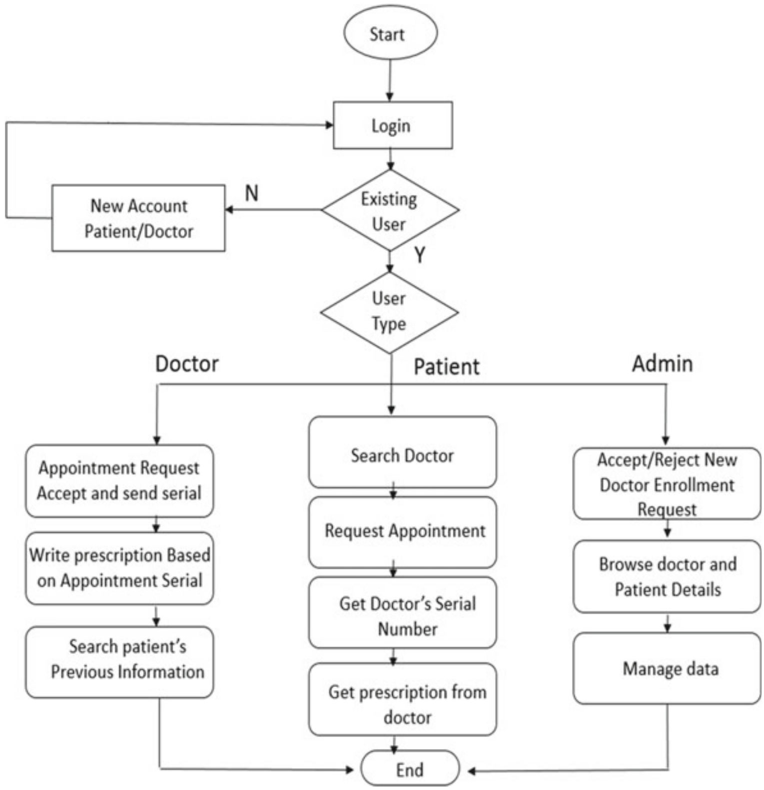


Fig. 1 Proposed HMS system

3.1 Hospital Management System

The hospital management system consists of a dashboard that is created for the admin, doctors, and patients. The admin has the authority to permit the appointments and can manage other hospital-related work from the Web site. The patient can book appointments and can-do disease prediction using the 'Disease Prediction' tab on the navigation bar. The doctors can edit their consultation timing according to their schedule through the Web site.

The dashboard shown in Fig. 2 contains the navigation bar from which the user can directly access various functions just by clicking on it. The navigation bar is common for all the pages. In the navigation bar, the 'Home' button is used for revisiting the dashboard from any other page; the 'Admin' option is used by admin to log in using username and password; the 'Doctor' option is used by the doctor for doctor's-related activities, and the same is with the 'Patient' option for patient related activities. One feature is 'disease prediction', by clicking on it the user can get to know about various diseases just by entering some health parameters, for example, glucose, BMI, age,

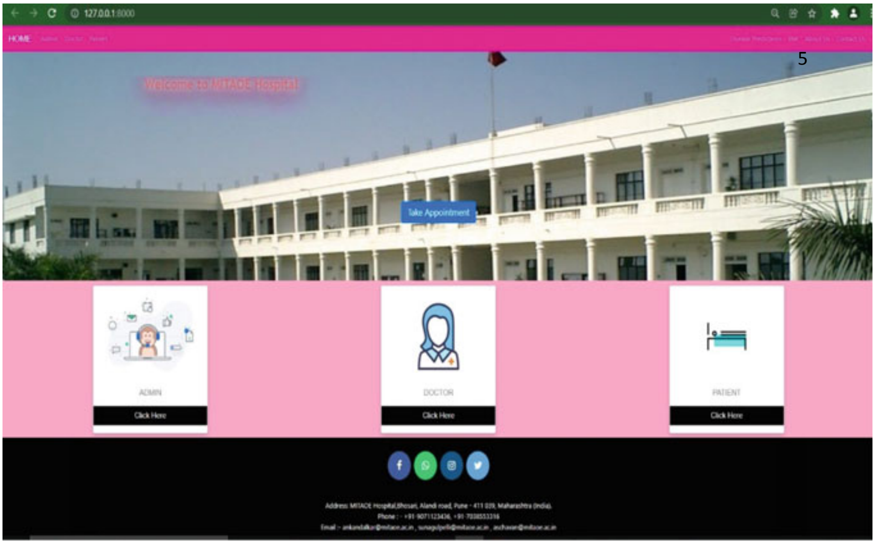


Fig. 2 HMS dashboard

pregnancies for diabetes. After disease prediction, the next option is the BMI feature which calculates the BMI of the user using the weight and height of the user. The ‘About Us’ function gives the information regarding the hospital, and the ‘Contact us’ function is very useful for any user as the user can drop a message with his name and the query. ‘Take Appointment’ feature is provided for easy use of the user to book an appointment for consulting a doctor. Easy access for the login of admin, doctor, and patient, the user can directly choose respective categories according to the type of user. The footer part helps the user access information and contacts the admin using various social media platforms. Whenever a new user is created, it has to be confirmed by the admin; then only, the account will get activated. Doctors can see the patient who has registered and what symptoms the patient is facing and also can provide prescriptions and can discharge the patient.

3.2 Machine Learning Model Selection

The machine learning models are used to predict whether a patient is having diabetes or not. Diabetes prediction is used to give a rough insight to the patient of whether they should consult the endocrinologist or not, and thus, they can take protective measures accordingly. The dataset is taken from Kaggle having the name ‘Pima Indians Diabetes’ database. It contains the data of Indian people, and there are in total 9 features in the dataset.

The following machine learning models are chosen for diabetes prediction to get optimum accuracy.

1. **K-Nearest Neighbors (KNN):** This is a supervised machine learning model where the variable is assigned to a particular outcome depending on the outcomes of its neighbors. So, if $k = 3$, that means the output of the 3 nearest neighbors of the variable is observed, and thus, the output of the new variable is predicted. The nearest neighbors are calculated using various distance formulas like the Euclidean, Manhattan, or hamming distance. Hamming distance is used for categorical variables, and the rest of the formulas are used for continuous variables. So, KNN is a simple model which is very easy to implement.
2. **Random Forest:** Random forest is also a supervised machine learning model which is a combination of many decision trees. The random forest model works great with high-dimensional data. It is also faster to train than the decision tree model as it is a combination of many decision trees. The decision trees might overfit the data but using a combination of decision trees to form a random forest model tends to eliminate overfitting. There are many hyperparameters in random forest models which can be tuned to increase the accuracy of the model. max_features, n_estimators, min_samples_leaf, max_depth are some of the hyperparameters of the random forest which are commonly tuned to increase the accuracy.
3. **Bagging (Max voting):** Bagging technique is also called as bootstrap aggregation technique. In this ensemble technique, different decision tree models contribute toward the output, so if 3 decision tree models are used, each model will give its output, and thus, our final output will be the mode of the outputs given by the models. For example, in the case of max voting, if 2 models give the output as 0 and 1 model gives the output as 1, then the majority will be considered as the final output, i.e., 0. The datasets given to the models are the sub-datasets of the original dataset. The process of prediction takes place concurrently. For classification, majority voting is used, and for regression, averaging, weighted averaging, and rank averaging are used to get the final output. As the output of the diabetes dataset is of classification type, the max voting technique is used for predicting the output. Figure 3 below gives us visual insights about the bagging approach.

4. **Boosting (XGBoost):** The process of output prediction of different decision tree models in this technique takes place sequentially. Initially, we assign a weight to each record, but the weights are based on the probability. Initially, the weights of all the records are equal. As per misclassification, weight is updated and is applied to the next decision tree, and based on voting, the final output of the classifier is selected. All the decision trees in the XGBoost technique have fixed depth. So, this technique helps us to get low bias and low variance.

Certain terminologies are important in selecting a particular model.

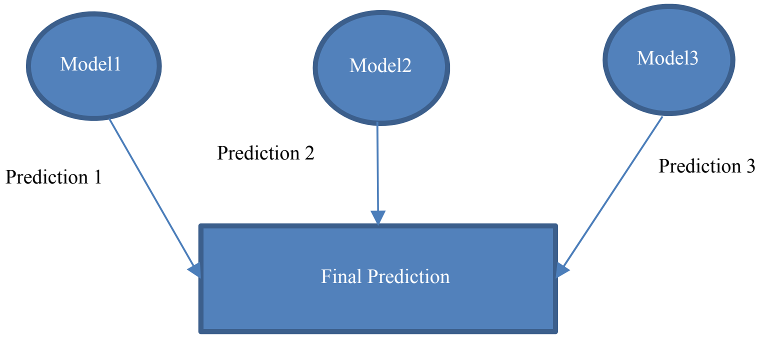


Fig. 3 Bagging approach for disease prediction

Table 1 Confusion metrics table

		Actual values	
		Positive	Negative
Predicted values	Positive	True positive (TP)	False positive (FP)
	Negative	False negative (FN)	True negative (TN)

Confusion Metrics: The confusion metrics give us an insight into how our model predicted the output. Table 1 below describes the same.

In the diabetes dataset case,

- True positive (TP): If a patient is diabetic and the model predicted as diabetic.
- False positive (FP): If a patient is non-diabetic and the model predicted as diabetic.
- True negative (TN): If a patient is non-diabetic and the model predicted as non-diabetic.
- False negative (FN): If a patient is non-diabetic and the model predicted as diabetic.

The evaluation metrics for classification, which are important for model selection, are listed below:

Accuracy: Accuracy is the overall efficiency of the model. The accuracy is calculated for the testing and training part of the model.

$$\text{Accuracy}(\%) = \frac{\text{True positive} + \text{True negative}}{\text{True positive} + \text{True negative} + \text{False positive} + \text{False negative}} * 100$$

Precision: It is the ratio between the true positives and all the positives

$$\text{Precision}(\%) = \frac{\text{True positive}}{\text{True positive} + \text{False positive}} * 100$$

We need to minimize the false-positive values to get a higher precision. Thus, in precision, false-negative rate is high.

Recall: It is the ratio between the true positives and all the positives

$$\text{Recall}(\%) = \frac{\text{True positive}}{\text{True positive} + \text{False negative}} * 100$$

We need to minimize the false-negative values to get a higher recall. Thus, in recall, false-positive rate is high.

F1 Score: The F1-score is the combination of precision and recall. It is the harmonic mean of precision and recall. It is maximum when precision and recall are equal.

$$\text{F1 - Score} = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}}$$

The model is considered good if its F1-score is high for both the training and testing phase.

3.3 Training and Testing of ML Models

- **Preprocessing:** The diabetes dataset has no null values, but there are some features like the insulin, glucose, and skin thickness which have 0 as a value in it as having 0 in these features is practically not possible. So, 0 is replaced by the mean or mode of that particular feature. This completes our preprocessing stage making the dataset clean and ready for training.
- **Feature Selection:** The correlation matrix depicts the correlation of all the features with one another. Few features have been used from the dataset to predict diabetes and to reduce complexity. The features selected have the best correlation with the diabetes outcome. Thus, the features used are glucose, age, BMI, and pregnancies as our features to detect diabetes with glucose having the highest correlation with diabetes.
- **Standardization:** The above 4 features are transformed into the range 0–1. The MinMaxScaler in the sklearn library helps us to perform this transformation. Standardization helps to give equal importance to all the features.
- **Training and Testing:** Using the machine learning models, the dataset was trained, and hence, the accuracy and evaluation metrics of the models for training and testing were observed, and the best model was selected accordingly.

4 Results and Discussion

The below result shown in Table 2 will give us an insight about how the models have performed in the diabetes dataset. The accuracy, precision, recall, and F1-score of different models are listed in the table.

0 Represents Diabetes Prediction as False, and 1 Represents Diabetes Prediction as True.

As the problem statement of the machine learning model is diabetes prediction, 'recall' of the model should be high, as the users cannot afford false-negative cases but can afford the false positives, that means even though the patient is not diabetic, the model predicts that the patient is diabetic and after consulting with the doctor the patient will know the truth, but if the patient is diabetic and the model showed the patient as non-diabetic, then this will be a serious issue as the patient may not consult the doctor regarding diabetes, and his diabetes stage may increase. So, the model should have higher recall along with accuracy.

The result table is shown below.

Table 2 shown above describes that the random forest and bagging technique have relatively smaller accuracy of recall metrics, while the XGBoost technique and k-nearest neighbor model have good accuracy of recall metrics. XGBoost has a high variance which can result in overfitting. A higher variance occurs when the model performs well on the training data but is unable to perform well on the testing data. Due to the higher complexity of XGBoost, it may lead to overfitting. On the other hand, KNN has a low variance which means it performs well on the training and testing data. By comparing both the models, we can conclude that KNN will be better than the XGBoost technique as it has low variance. Thus, KNN has a good accuracy as well as a good recall metric. Therefore, the kNN model is deployed on the Web site.

Table 2 ML model performance for diabetes prediction

Model	Testing						
	Accuracy (%)	Precision (%)		Recall (%)		F1 score (%)	
		0	1	0	1	0	1
KNN ($k = 9$)	84.42	87.38	76.74	90.65	70.21	88.99	77.85
Random forest	79.87	83.92	69.04	87.85	61.7	85.84	65.16
Bagging (Max voting)	82.47	83.89	77.77	92.52	59.57	88.00	67.46
Boosting (XGBoost)	79.22	85.71	65.30	84.11	68.08	84.90	66.67

5 Conclusion

The hospital management system is essential for maintaining a database of doctors, patients, and hospitals. In the proposed research work, HMS is proposed along with a machine learning model which would help the work at the hospitals to be seamless and efficient. The data collection of the patient through developed HMS will be easy to use for doctors and hospital management. As machine learning models are deployed in the HMS system, it helps to predict the patient's disease which saves the time of the doctor, and it will help for further treatment. The k-nearest neighbors (KNN) are the machine learning model deployed on the system to predict diabetes as it has a relatively higher accuracy and recall metrics than other ML models. The Web site is designed using the Python Django framework which helps to maintain the database and route the HTML pages. With the help of HMS, the administrative process like billing and patient records is also handled smoothly and efficiently. Thus, HMS can prove to be a great asset to the hospitals.

Acknowledgements We would like to thank Prof. Smita Kulkarni for her expert guidance and continuous encouragement throughout to see that this project meets its target from its commencement to its completion.

References

1. Yu L-J, Wang H-Q, Gou L, Tian Y, Li J-S (2015) Data analysis on health management systems for improving Doctor's advice on patients. In: 2015 7th International conference on information technology in medicine and education. https://ieeexplore.ieee.org/document/7429099
2. Jaiswal K, Sobhanayak S, Turuk AK, Bibhudatta SL, Mohanta BK, Jena D (2018) An IoT-cloud based smart healthcare monitoring system using container based virtual environment in edge device. In: 2018 International conference on emerging trends and innovations in engineering and technological research (ICETIETR). https://ieeexplore.ieee.org/document/8529141
3. Mahmood N, Shah A, Waqas A, Bhatti Z, Abubakar A, Malik HAM (2014) RFID based smart hospital management system: a conceptual framework. In: The 5th international conference on information and communication technology for the Muslim world (ICT4M). https://ieeexplore.ieee.org/document/7020594
4. Zilani KA, Yeasmin R, Zubair KA, Sammir MR, Sabrin S (2018) R^3HMS, an IoT based approach for patient health monitoring. In: 2018 International conference on computer, communication, chemical, material and electronic engineering (IC4ME2). https://ieeexplore.ieee.org/document/8465482
5. Ma L, Zhao H, You SJ, Wenyong G (2018) Analysis and design of hospital management information system based on UML. In: AIP conference proceedings, vol 1967, p 040012. https://doi.org/10.1063/1.5039086. Published Online 23 May 2018
6. Neloy AA, Alam S, Bindu RA, Moni NJ (2019) Machine learning based health prediction system using IBM cloud as PaaS. In: 2019 3rd International conference on trends in electronics and informatics (ICOEI). https://ieeexplore.ieee.org/document/8862754
7. Ponmalar A, Maansi S, Mahalakshmi S, Shalini M, Madhavan R (2021) Mobile application for hospital management system. In: 2021 5th International conference on intelligent computing and control systems (ICICCS). https://ieeexplore.ieee.org/document/9432286

8. Jain Y, Gandhi H, Burte A, Vora A (2020) Mental and physical health management system using ML, computer vision and IoT sensor network. In: 2020 4th International conference on electronics, communication and aerospace technology (ICECA). https://ieeexplore.ieee.org/abstract/document/9297447

Fast Automated Detection of COVID-19 from CT Images Using Transfer Learning Approach

Jyoti Mante, Swarupa Deshpande, and Prerna Patil

Abstract A long clinical testing period is one of the key elements for the COVID-19 pandemic's fast spread. Controlling the spread of COVID-19 requires early detection and diagnosis. Chest X-ray (CXR), for example, is an imaging technology that helps to speed up the identifying procedure of COVID-19 in patients. As a result, our goal is to create an automatic CAD system that can recognize COVID-19 samples from healthy people and COVID patients using CT scans. We used transfer learning (TL) approach, i.e., modified Visual Geometry Group (VGG19) and compared our proposed system results with other machine learning (ML) and deep learning (DL) approaches in order to discover the best one for this job. The proposed technique and various DL and ML models are tested using the COVID-CT dataset, where 80% of images are utilized for training and 20% for testing purpose. Our proposed TL technique achieves 97.83% classification accuracy with average precision, recall, and F1-score of 98.33, 97.67, and 97.67, respectively.

Keywords COVID-19 · Deep learning · Machine learning · Transfer learning

1 Introduction

The WHO declared COVID-19, a pandemic caused by SARS-CoV-2 in March 2020. COVID-19 is very infectious and transform into other viruses. COVID-19 treatment needs early detection. The reverse transcription polymerase chain reaction (RT-PCR) technique is used at initial phases to detect COVID symptoms; however, it is a time-consuming approach with limited sensitivity. X-rays and CT scans of the chest were utilized to determine the morphological patterns of lung lesions connected to

J. Mante (✉) · S. Deshpande · P. Patil
School of Computer Engineering and Technology, MIT World Peace University, Pune, India
e-mail: jyoti.khurpade@mitwpu.edu.in

S. Deshpande
e-mail: swarupa.deshpande@mitwpu.edu.in

P. Patil
e-mail: prerna.patil@mitwpu.edu.in

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_30

the COVID-19. The accuracy of COVID-19 diagnosis by CT scans is depends on experts. DL approaches have been examined as a tool for diagnostic automation and assistance. CT scans give comprehensive pictures of organs, soft tissues, bones, and blood arteries, and it also allows doctors to view the form, size, texture of interior structures, and density. CT scans, unlike traditional X-rays, provide a series of slices of a specific body location without overlaying the various body components. CT scans, as comparison with traditional X-rays, provide a far more comprehensive image of the patient's health. This detailed information is used to determine whether or not a medical ailment exists, as well as its breadth and specific location. A number of TL-based techniques for COVID-19 screening in CT scans have recently been presented for these reasons. The absence of excellent quality comprehensive datasets is the key constraint for completing studies like the ones described above [1].

Testing has been limited to persons who have indications, and in many cases, symptoms, leading to a shortage of resources and technology in several countries [2]. Even in the most developed countries, the situation has placed enormous pressure on national health-care institutions and workers which makes detecting and tracking possible cases considerably more difficult. Artificial intelligence (AI) algorithms approaches are implemented to track the people's activities and how they consider the social distance standards.

COVID-19 necessitates prompt to stop the further spread from one person to another as successfully diagnose early can prevent the spread of COVID-19. If it is detected early on, it is curable and may save a person's life. COVID-19 stage determines the appropriate therapy. COVID-19's diagnostic test sensitivity is restricted owing to specimen processing abnormalities. Coloring techniques for annotating X-ray and CT images were used to train the TL model. This technique achieves good performance and delivers high scores for several statistical indices (F1-scores > 97.83%). As a consequence of the NN-based regression, there are strong links between the lesion locations in the images and provides major clinical indications. The proposed technique can be used in clinical practice as a computer-aided diagnostic tool for COVID-19. SARS-CoV-2 dataset [3] is utilized for the experimental findings.

1.1 Project Contribution

1. The creation of a robust and new approach for DL employing TL methods for identifying and classifying CTx by extracting essential features on a standard dataset is one of the research's main accomplishments.
2. We altered the architecture of the VGG19 by adding additional filters to layers, and changing last output layer which improved features and reduced noise. With Adam optimizer, as well as mean square error loss functions, we applied transfer learning strategies. This allowed for the identification of a COVID or non-COVID.

3. Also we provide an in-depth examination of important performance indicators and performed the fine-tuning on pre-trained systems.
4. Different ML, DL, and TL algorithms outcomes are compared.

In addition, the document is organized as in the Sect. 2, existing techniques are presented; system design, algorithms, and methodology are detailed in the third part; dataset definition, evaluation parameters, testing procedure, and results analysis were reviewed in fourth section; and lastly, the conclusion is stated.

2 Literature Review

2.1 Data Processing and Segmentation

In order to identify COVID-19 images from normal and COVID patients, Rahaman et al. [1] recommended using deep convolution neural network (CNN). The study tested 15 pre-trained CNN models on the ImageNet dataset and compared their findings to find a model that is suitable for this purpose. To extract features and perform classification, the transfer learning approach is used. Finally, the network is fed 90 unseen test pictures to assess the proposed method's performance. By studying CT chest scan pictures, Hussein et al. suggested a platform that encompasses various layers of examination and categorization of normal and pathological characteristics of COVID-19. The platform begins by augmenting the training dataset with a trustworthy collection of photos, identifying suspicious parts in images, and analyzing these regions in order to get correct categorization. The collected findings demonstrate that the proposed architecture is 95% accurate [4]. Subramaniam et al. advise pre-processing of X-ray lung images to increase classification performance over raw images to detect COVID-19. Researchers utilized X-ray scans of the lungs to extract the features and differentiate the left and right lungs. Suggested DNN outperformed the other models. [5]. Silva et al. [6] devised a voting-based DL approach for COVID-19 screening. This method groups images from every patient using a voting system. The method has been validated using two largest COVID-19 CT datasets. Several obstacles and concerns linked to DL implementations for COVID-19 medical image processing were presented by Bhattacharya et al., which are likely to stimulate additional research in limiting the epidemic and managing the crisis, resulting in smart healthy cities [7].

A CNN is trained using pseudo-coloring and X-ray and computed tomography images is expert-like in performance and scores high on several statistical indices [8]. In COVID-19, Ai et al. explore the diagnostic usefulness and consistency of chest CT in contrast to the RT-PCR test. COVID-19 diagnosis is done using chest CT scan dataset which provides a high sensitivity. In epidemic areas, CT images be the key technique for COVID-19 identification [9]. Amyar and colleagues present a novel multitask DL model for identifying COVID-19 patients and segmenting COVID-19 lesions from chest CT images. Different datasets were utilized to complete three learning tasks: segmentation, classification, and reconstruction [10]. COVID-Net, a DCNN designed for detecting COVID-19 instances from chest X-ray (CXR) images was presented by Wang et al. they used an open source dataset of CXR images [11]. The problem's taxonomy is exploited using a hierarchical classifier. Luz et al. [12] are trained on COVID-19 dataset. Authors employ a second dataset to test the approach's generalization. Hearst et al. proposed SVM [13] for classification problem, SVM is a nonlinear classification model. The kernel in this model implicitly transfers inputs data into high-dimensional feature spaces.

3 Proposed Work

3.1 Proposed System

In the presented paper, transfer learning-based DL algorithms are accessed for identifying CTx of COVID-19 into COVID and non-COVID class accurately. The system is improved via the use of numerous designs in order to get the best outcome. Figure 1 depicts the architecture of a system. The dataset contains COVID-19 data. ML, DL, and TL are used to train the system using the input dataset. The test dataset is fed into the system, which evaluates it and generates graphs based on the findings. We have used data pre-processing, resizing images, and 80–20% train test split ratio for input. The system will evaluate and check the COVID-19 and non-COVID cases.

3.2 COVID-19 Classification with Deep Learning Algorithm (CNN)

CNN model is used initially, in which the input images are processed via a series of layers, including convolutional, pooling, flattening, and fully connected layers, before the CNN [14, 15] output, which classifies the images, is generated. Following the building of CNN models from the ground up, we will use image augmentation approaches to fine-tune the model. The following are the fundamental components of a CNN: (a) convolution layer, (b) activation functions (ReLU), (c) pooling layer, and (d) fully connected layer (Fig. 2).

3.3 COVID-19 Classification with Transfer Learning Algorithm (VGG19)

TL is a prominent approach for transferring information from one to another domain in the computer vision and DL fields. In the situation of restricted processing

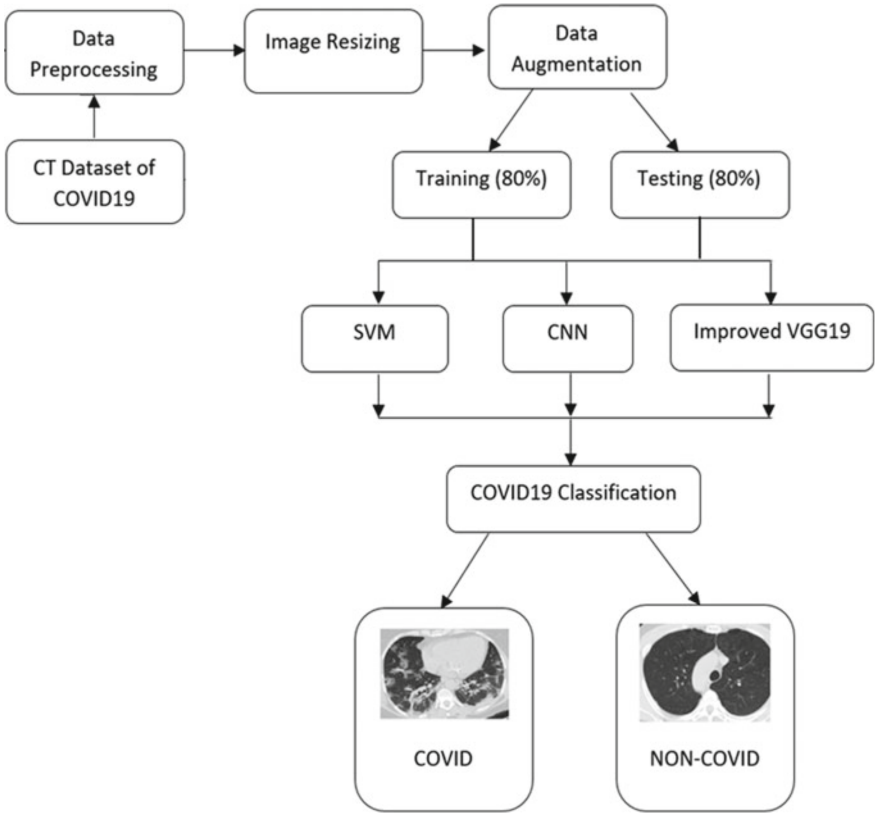


Fig. 1 Proposed system architecture

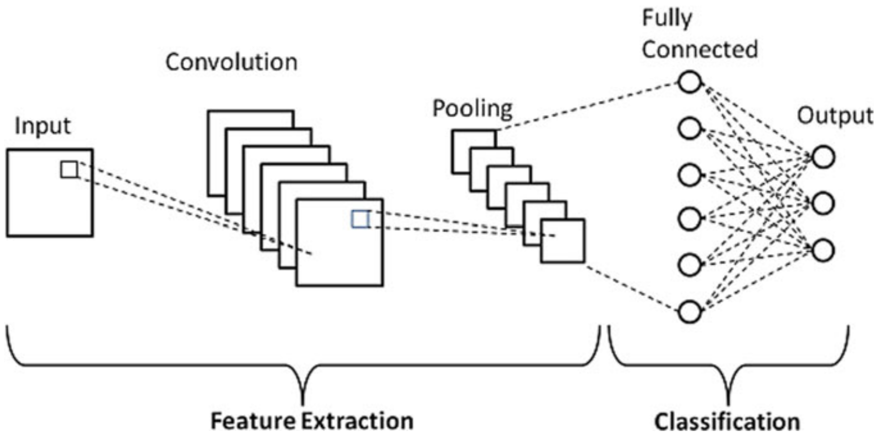


Fig. 2 CNN classification model

resources, TL enables users to employ pre-trained weights from image domain. Here, we used modified version of VGG model to learn and predict the COVID-19 symptoms in human. The VGG model, or VGGNet, that upholds 19 layers is likewise referred as VGG19. The "19" represents the number of weight layers in the model (convolutional layers). VGGNets uses most important features of CNN. As a result, one of the pre-trained models—VGG19 will be used to categories images for both training and validation data.

3.4 Algorithm: UPDATED Transfer Learning VGG19

It is a really uncomplicated model to use, with a simple model structure to grasp. To classify fresh CT images, we used the VGG19 model. To begin, the picture must be stacked and shrunk to 224 × 224, as specified in the proposed model, with pixel values scaled in the manner specified by the model. Because the model operates on a range of tests, components of a stacked images should be multiplied by one for a single image with 224 × 224 pixels and three channels. The model then be layered and a prediction generated. Before the yield layer, the VGG19 model's last layers are totally associated layers. These layers offer important information for developing another model for image order or other PC vision tasks. Figure 3 displays the VGG19 model's layout.

In VGG19 model, we have update,

1. A new sequential model is generated first. Then, except for the last layer, loop through each of the layers in VGG19 model and append every layer to the new sequential model (SQM).
2. We have now replicated the whole VGG19 model (excluding the output layer) to a new sequential model (SQM).
3. After that, we will go over each layer in our new sequential model (SQM) and make them non-trainable. This freezes the weights and other trainable parameters in every layer from being modified or trained when we subsequently pass in our COVID and non-COVID pictures. Because the COVID and non-COVID pictures were previously inserted in the earlier phases, we would not want to relearn these layers. As a result, VGG19 performs a good job at classifying these groups. We just want to modify the model in such a way that the output layer knows COVID

Fig. 3 VGG19 model

Table 1 VGG19 layers distribution

Layers name	Number of layers
Convolutional layers	16
Pooling layers	5
Dense layers	3

and non-COVID classifications and nothing else. As a result, we do not want any re-training to take place on the previous levels.

4. Finally, at the end, new output layer is added, which consists of just two nodes: COVID and non-COIVD. This output layer will be the model's sole trainable layer.
5. We can now look at a summary of our model and see that it is identical to the original VGG19 model, with the exception that the output layer now has only two nodes instead of 1000, and the number of trainable parameters has decreased dramatically since we froze all the parameters in the previous layer.

The following Table 1 shows the number of layers used for the development of VGG19 architecture.

4 Result and Discussion

4.1 Evaluation Metrics

The evaluation methodology is based on the confusion matrix. F1-score, accuracy, recall, precision, sensitivity, and specificity are used to evaluate proposed model performance. The following variables are used to construct the measures:

$$\text{Accuracy} = \frac{T_\text{p} + T_\text{n}}{T_\text{p} + T_\text{n} + F_\text{p} + F_\text{n}}$$

$$\text{Sensitivity} = \frac{T_\text{p}}{T_\text{p} + T_\text{n}}$$

$$\text{Specificity} = \frac{T_\text{n}}{F_\text{p} + T_\text{n}}$$

$$\text{Precision} = \frac{T_\text{p}}{T_\text{p} + F_\text{p}}$$

$$\text{Recall} = \frac{T_\text{p}}{T_\text{p} + T_\text{n}}$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Table 2 Tuning parameters

PARAM	Tuning (CNN)	Tuning (VGG19)
droupout_prob	0.01	0.02
Optimizer	Adam	RMSProp
Epoch	25	25
Batch size	10	2
Loss function	Mse	binary_crossentropy
Activation function	ReLU/softmax	ReLU/sigmoid

Here, $T_{\mathrm{n}}, T_{\mathrm{p}}, F_{\mathrm{p}}, F_{\mathrm{n}}$ represent to true negative, true positive, false positive, and false negative.

4.2 Result Analysis

Different ML, DL, and TL are trained and tested on dataset [3]. Algorithm such as SVM, CNN, and improved VGG19 are used to detect the COVID-19. Table 2 lists the specific tuning parameters used throughout the project's execution.

Table 3 shows the performance parameters comparison of algorithms, and Fig. 4 shows the graphical representation of all parameters values. From experimental analysis, we can clearly see that VGG19 outperforms ML as well as DL algorithms by good margin in terms of accuracy, recall, precision, and F-measure. Figure 5 shows the graphical representation of those parameters values.

Figure 6 shows training and validation comparison graph of CNN and VGG19. We have train and test system by keeping 25 epochs. From the results, we can clearly say that with increasing number of epoch both CNN and VGG19 improves accuracy but VGG19 outperforms CNN in terms of accuracy by good margin. Figure 6 shows the training and validation loss comparison graph of CNN and VGG19. VGG19 has very less loss at the end of 25th epoch which is below 0.02.

From the comparative analysis, we can say that VGG19 outperforms CNN because of

- VGG19 model simplicity makes the network seriously engaging.
- VGG utilizes very small receptive fields (3×3 with a stride of 1). Since there are currently three ReLU units rather than only one, the choice capacity is more discriminative.

Table 3 Performance parameter comparison of ML, DL, and TL algorithms

	Precision	Recall	F-measure	Accuracy
SVM	83	78	79	85.09
CNN	90	88	88	92.14
VGG19	98	97	97	97.83

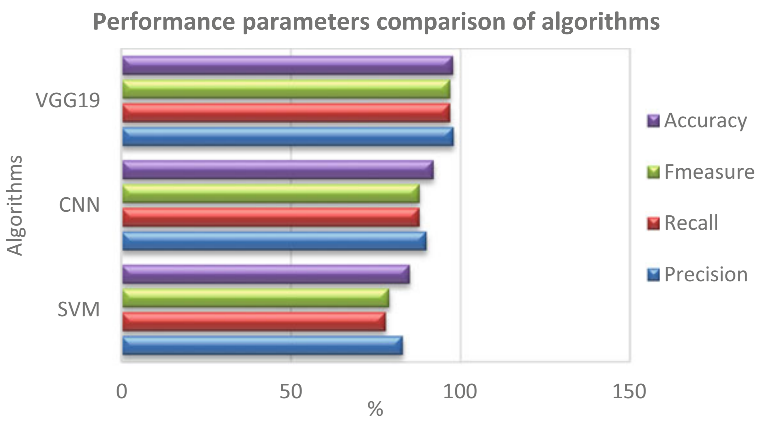


Fig. 4 Accuracy graph comparison of various researcher's models

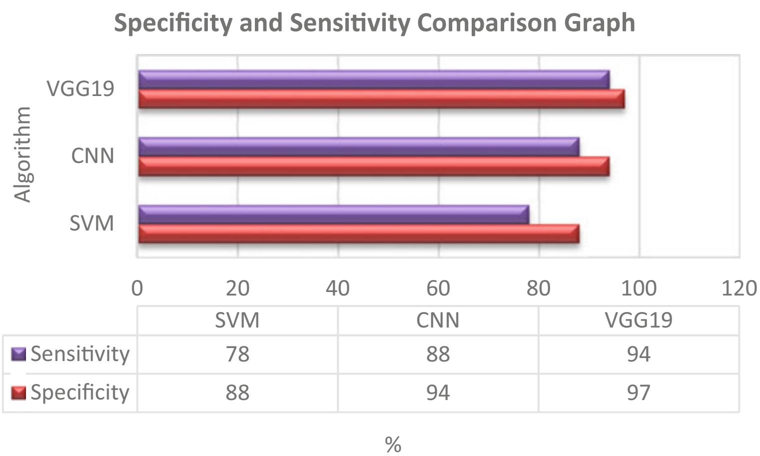


Fig. 5 Specificity and sensitivity comparison graph

- VGG fuses 1 × 1 convolutional layers to settle on the decision function more nonlinear without changing the receptive fields.
- The small size convolution channels permit VGG to have large number of weight layers; with more number of layers the framework execution gets gotten to the next level.

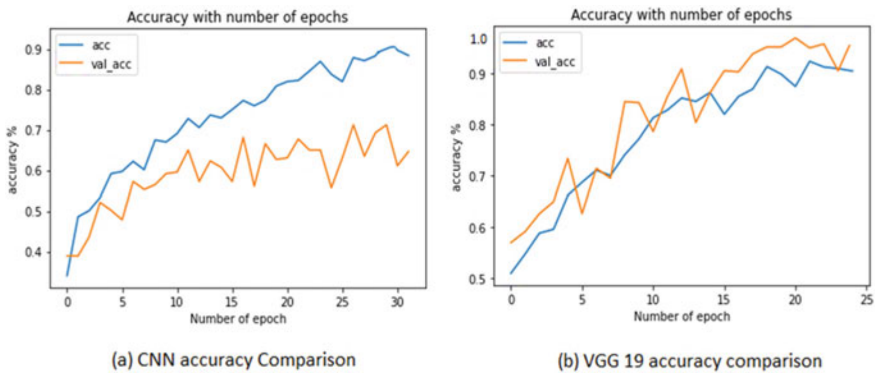


Fig. 6 Training and validation accuracy comparison graph of **a** CNN **b** VGG19

5 Conclusion

An improved VGG19 model for detecting COVID-19 symptoms in CT scans images is presented in our work along with a ML (SVM) and DL (CNN)-based technique. Publicly accessible dataset SARS-CoV-2 dataset is used for the experiments. The proposed system achieves comparable results to state-of-the-art approaches and has the greatest accuracy of 97.33%. Both the CNN and VGG19 models have been fine-tuned. According to our findings, approaches for detecting COVID-19 in CT scans are improved which can be considered as a clinical alternative. The suggested improved VGG19 technique favors false positive and false negative detection, which helps to improve accuracy.

References

1. Rahaman MM, Li C, Yao Y, Kulwa F, Rahman MA, Wang Q, Qi S, Kong F, Zhu X, Zhao X (2020) Identification of COVID-19 samples from chest X-Ray images using deep learning: a comparison of transfer learning approaches. J Xray Sci Technol 28(5):821–839
2. Zhao J, Zhang Y, He X, Xie P (2020) Covid-CT-dataset: a CT scan dataset about covid-19. arXiv Preprint. arXiv:2003.13865
3. Soares E, Angelov P, Biaso S, Froes MH, Abe DK (2020) Sars-cov-2 CT-scan dataset: a large dataset of real patients CT scans for sars-cov-2 identification. medRxiv
4. Kaheel H, Hussein A, Chehab A AI-based image processing for COVID-19 detection in chest CT scan images. Front Commun Netw
5. Subramaniam U, Subashini MM, Almakhles D, Karthick A, Manoharan S (2021) An expert system for COVID-19 infection tracking in lungs using image processing and deep learning techniques. BioMed Res Int 2021(1896762):17
6. Silva P, Luz E, Silva G, Moreira G, Silva R, Lucio D, Menotti D (2020) COVID-19 detection in CT images with deep learning: a voting-based scheme and cross-datasets analysis. Inform Med Unlocked 20:100427. ISSN: 2352-9148

7. Bhattacharya S, Reddy Maddikunta PK, Pham QV et al (2021) Deep learning and medical image processing for coronavirus (COVID-19) pandemic: a survey. Sustain Cities Soc 65:102589. https://doi.org/10.1016/j.scs.2020.102589
8. Liang S, Liu H, Gu Y et al (2021) Fast automated detection of COVID-19 from medical images using convolutional neural networks. Commun Biol 4:35
9. Ai T, Yang Z, Hou H, Zhan C, Chen C, Lv W, Tao Q, Sun Z, Xia L (2020) Correlation of chest CT and RT-PCR testing for coronavirus disease 2019 (COVID-19) in China: a report of 1014 cases. Radiology 296(2):E32–E40
10. Amyar A, Modzelewski R, Li H, Ruan S (2020) Multi-task deep learning based CT imaging analysis for COVID-19 pneumonia: classification and segmentation. Comput Biol Med 126:104037. ISSN: 0010-4825
11. 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
12. Luz E, Silva P, Silva R et al (2021) Towards an effective and efficient deep learning model for COVID-19 patterns detection in X-ray images. Res Biomed Eng
13. Hearst MA, Dumais ST, Osuna E, Platt J, Scholkopf B (1998) Support vector machines. IEEE Intell Syst Appl 13(4):18–28
14. Tammina S (2019) Transfer learning using VGG-16 with deep convolutional neural network for classifying images. Int J Sci Res Publ (IJSRP) 9(10). ISSN: 2250-3153
15. Patil R, Bellary S (2021) Transfer learning based system for melanoma type detection. Revue d'Intelligence Artificielle 35(2):123–130

AI Based Optimization Algorithms and Applications

Survey of Stochastic Number Generators and Optimizing Techniques

Pooja Nahar, Prasad Khandekar, Minal Deshmukh, Harpreet Singh Jatana, and Uday Khambete

Abstract Stochastic computing (SC) is an error-tolerant computing technique where computation takes place on randomized statistical bit streams. The main feature is that standard digital logic gates with low power and low area can be used to implement complex arithmetic functions. However, these advantages need to be compared against slow speed and less precision. Also, the area required by stochastic number generators (SNGs) is high as compared with the area required to implement computation circuit. So, if multiple SNGs are used, then the advantage of low area cannot be extracted. So, there is a need to design good quality, reliable and higher throughput SNGs. Out of different approaches that are used to design SNG, use of low discrepancy sequences and randomization functions has better results with respect to throughput and correlation management, respectively, at the cost of area. So, more research using these two approaches can lead to design of efficient SNG. This paper reviews the background, fundamental concepts, basic arithmetic operations using SC, various approaches to design efficient SNG's by managing correlation in stochastic computing circuits, advantages, and drawbacks followed by conclusion and future challenges.

Keywords Stochastic computing · Approximate computing · Stochastic arithmetic · Random number sources · Random number generator · LFSR · Correlation · Low-power design

P. Nahar (✉)
School of Electronics and Communication Engineering, Dr. Vishwanath Karad MIT World Peace University, Pune, India
e-mail: poojanahargandhi@gmail.com

P. Khandekar · H. S. Jatana
Dr. Vishwanath Karad MIT World Peace University, Pune, India
e-mail: Prasad.khandekar@mitwpu.edu.in

M. Deshmukh
Vishwakarma Institute of Information Technology, Pune, India
e-mail: Minal.deshmukh@viit.ac.in

H. S. Jatana · U. Khambete
Semiconductor Laboratory, Chandigarh, India
e-mail: udayk@scl.gov.in

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_31

Background

The historical advances in stochastic computing (SC) can be emphasized mainly in four different phases [1]. 1950s were the early years which describe mainly the work done by John V. Neumann. In the 1960s and 1970s, a lot of advancements were made in the technique by researchers like B. R. Gaines, W. J. (Ted) Poppelbaum, Chusin Afuso, John Esch, Sergio Ribeiro, and others who developed the software and the hardware in which SC was used to do the arithmetic operations. The first conference on SC took place in 1978. After that, there was not much research done until 2000. In the year 2001, Howard Card put out two papers on SC in the IEEE [2, 3]. After that, the interest in SC increased. Subsequently, workshops on SC were held at the University of Waterloo in 2016 and in Montreal in 2017. Section 1 of this paper describes the initial work done in SC mainly from the years 1950s to 1970s. Section 2 describes recent development in SC. Section 3 shows errors, ways to minimize these errors, impact of correlation on stochastic computation and ways to manage/manipulate correlation in stochastic circuits. Section 4 covers the optimization techniques for SC circuits. Section 5 and 6 covers the advantages and drawbacks, conclusion and future scope, respectively.

1 Initial Work

There is a lag between application of input and the response at the output of every network. Error is an essential part of such circuit. As shown in [4], this error can be used for synthesizing reliable logic elements using unreliable components for the process under consideration. Similarly, redundancy is also present in biological neural models. This redundancy can be used to compensate component error. Hence, the use of bundles instead of single wire for carrying information was suggested. But, the probability of energizing a wire in a bundle needs to be random. This is because without randomness, error can be amplified instead of getting cancelled out. This bundling of wires can then be used to carry out arithmetic operations like addition (using stochastic multiplexer) and multiplication (based on Sheffer's stroke, like stochastic multiplier). The need for random selection of inputs gave rise to third form of computing apart from digital and analog, which was done using random pulse sequence (RPS) combined with standard logic gates to realize various arithmetic and logical functions. This facilitated huge reduction in area and cost of a computational element.

Afuso as shown in [3, 4] developed RPS for SC. But, since the system required a local random pulse generator, a better system called as synchronous random pulse generator (SRPS) was developed by Afuso under the guidance of Prof. Poppelbaum at the University of Illinois. RPS was discrete; SRPS was digital. Following Fig. 1 shows production of RPS and SRPS. A noise diode is used to generate random noise. The occurrence of noise spikes is then detected by an adjustable detection voltage level and amplified. The output of difference amplifier is sampled by clock. A pulse

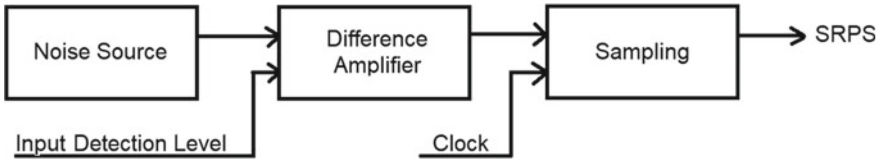


Fig. 1 Simplified SRPS [4]

of standard height and length is generated at the output of the sampling circuit only if it has detected a pulse in the previous slot. This pulse train generated at the output of the sampling device is completely random, and its frequency can be controlled by the detection level. But, buffering in SRPS lead to un-randomizing. So, clocked random pulse sequence (CRPS) was developed. CRPS utilized mapping of numbers in the range $[-1, 1]$ onto range $[0, 1]$ which permitted all operations to be performed with only combinational circuitry, thus preserving the probability distribution.

These RPS was used to design different types of pulse processing systems as seen in [4] like time stochastic processing, bundle processing, ergodic processing, and burst processing. So, all the software algorithms were established and simulated on traditional computer, but applications like machine learning required specific complex hardware structure which was not available at that time. The main challenge being the design of an element which would store and vary the stored value for long periods. The element should also be able to be used as a weight to multiply with other variables. This system needed to occupy less area and low cost because these elements were needed in large numbers in a practical system. To design such a system, analog, electro-chemical devices, or standard digital components using semiconductor devices were the options available. The traditional analog integrates and multiply circuits are not stable and require more cost in terms of area. Electro-chemical devices or transfluxors are not reliable and require refined external circuitry to function properly. The high speed, good stability, small size, low cost, and large-scale integration features of semiconductor devices made them the obvious choice for designing of the computing elements. von Neumann and Pierce [4] shows examples of few stochastic computers developed during the 1960s such as portable stochastic computer (POSTCOMP), adaptive digital element (ADDIE), regular array of stochastic computing element (RASCEL), TRANSFORMATRIX, and autonomous processing element (APE). All these computers used one of the pulse processing systems and perform calculations using probability logic. However, with the advent of large-scale integration (LSI) in the late 1960s, interest in SC decreased until the year 2000. The ability of stochastic circuits to effectively perform tasks like decoding in communications and implementing complex computing tasks at low hardware cost in artificial neural network has rekindled the interest of researchers in this field. Next section describes how SC is used today in computation and various methods in which a SC circuit can be designed and what parameters need to be taken into consideration while doing so.

2 Recent Developments in Stochastic Computing

For performing stochastic computation, the binary number under consideration is first converted into stochastic domain. All the calculations or functions that need to be performed are done in stochastic domain, and the result is then converted back to binary if the required output is supposed to be in binary domain. Following Fig. 2 [5] shows a traditional stochastic number generator (SNG). It is used to convert a binary number into a stochastic number and vice versa. A comparator is used as a probability conversion circuit (PCC) that compares the binary number of m bits with a random number of m bits generated by the random number source (RNS). For every binary number B greater than R, a one is produced at the output. The number X is called as a stochastic number (SN) and has a probability pX. It is defined as the ratio of total number of 1s to total number of bits in a stochastic bit stream. E.g., A sequence $X = 10{,}101{,}100$ has Px = Probability of 1s in X as (4/8), i.e., 0.5. All the arithmetic operations are done in the stochastic domain using the generated SN's. The final output can then be converted to binary format by using a counter that will count the number of 1s in the sequence X. The randomness and length N determine the accuracy of created SN. Ideally, a stochastic number generator should be able to generate random binary sequence where each new bit is independent of the previous bit. Various RNS have been used like noise diode, LFSR [6], NLFSR [7], nanodevices like magnetic tunnel junction (MTJ) [8], and memristors. Some of which are true RNS, while others are pseudo-RNS. Similar to one in Fig. 2, there have been different RNG's that are designed like weighted binary RNG as proposed by Gupta and Kumaresan in [6], 4-bit binary SNG by Van Daalen et al. [9], and sampling-based RNG [10].

Using above circuit, all the basic arithmetic operations, operations using memory and design of finite state machines can be done as shown in Fig. 3. Multiplication is performed using an AND gate. Addition (scaled) is done using a multiplexer. Similarly, division is performed by using a counter in the specific arrangement using D flip-flop, and subtraction is performed using an EX-OR gate. The range of the inputs should be between [0, 1] to be able to be represented as a probabilistic bit

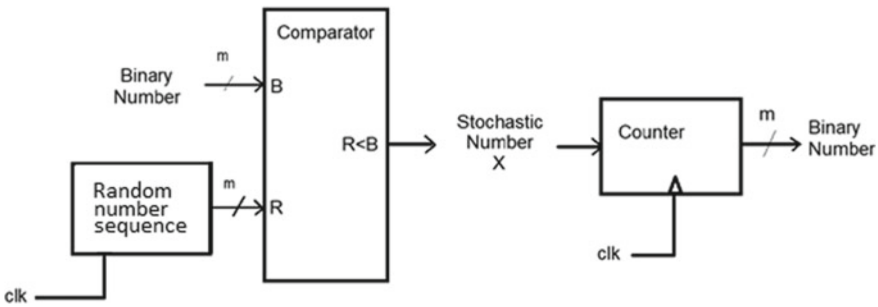


Fig. 2 Binary to stochastic and stochastic to binary converter [5]

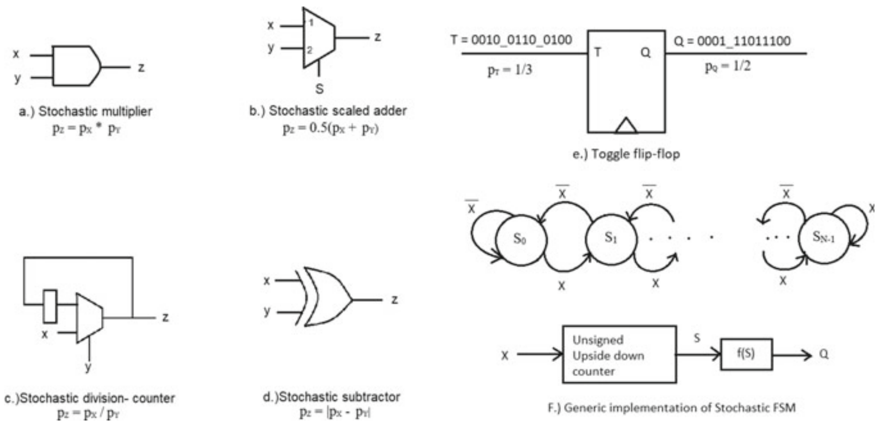


Fig. 3 Arithmetic, sequential, and FSM operations using SC [1] **a** multiplication, **b** scaled addition **c** division **d** subtraction **e** sequential element—flip-flop **f** FSM

sequence. If not, the inputs need to be scaled. To represent a polynomial or non-polynomial function such as trigonometric functions whose co-efficient do not lie in the interval [0, 1], they are converted to Bernstein polynomials. For operations using memory, consider a toggle flip-flop (TFF). The TFF has the output probability of 1/2 irrespective of any non-zero input. So, TFF can be used to generate a stochastic number with a fixed probability thus ruling out the requirement of an RNG. Finite state machine (FSM) implementation of SC is shown in Fig. 3f where an up/down counter is used to obtain the state-transition. The counter will increment at $X = 1$ and decrement for $X = 0$. F is the mapping of output which can be realized using a combinational function of S.

3 Errors and Correlation in Stochastic Computing

Stochastic circuits can be designed using two different approaches. One is reconfigurable stochastic circuits, and other is fixed stochastic circuits. The inputs of reconfigurable stochastic circuits are programmable, and the circuit can be reused to implement different functions, whereas fixed stochastic circuits can be used for a dedicated function implementation. But, since SC is an approximate form of computing, there are various known sources of errors [1] as shown in Fig. 4.

In above circuit, C is the core of the stochastic circuit. It can either be sequential or combinational logic circuit. $X1$, $X2$.... Xn are the input stochastic numbers of length N and are assumed to be independent of each other. To convert the input X to a desired probability of the value of X, the ancillary inputs $R1$, $R2$,.....Rk with constant value Ri $= 0.5$ and are used. Rk's and Xn's are assumed to be independent of each other. Any physical errors caused due to defect in hardware or sources like

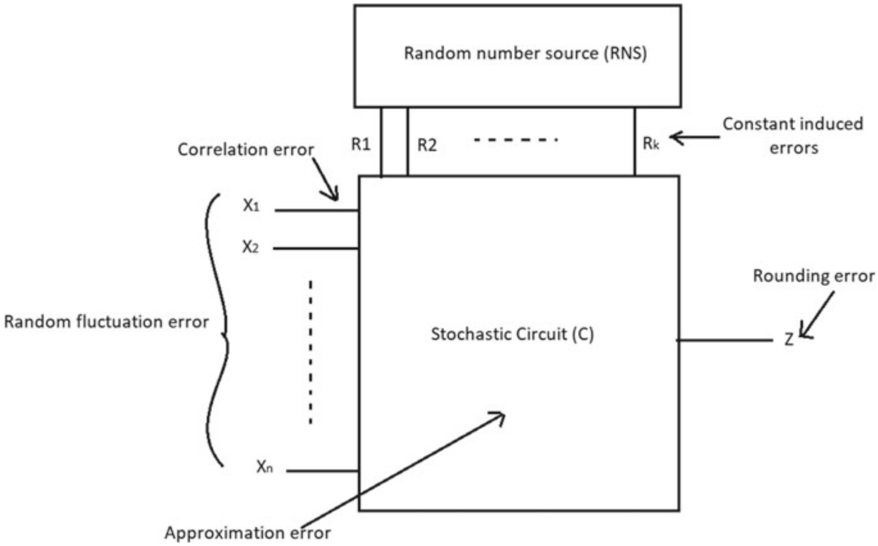


Fig. 4 Generic stochastic circuit with known sources of errors [1]

radiation are not shown in Fig. 4. The errors and the ways to overcome errors are described in detail as below:

Rounding Errors: These are errors caused due to quantization. For, e.g., a bit stream of Y bits can constitute of $(Y + 1)$ numbers as SN = $\{0, 1/Y, 2/Y, ..., (Y - 1)/Y, 1\}$. If a number under consideration Z is not in this set, then it is rounded to the closest value in the set of SN. For example, with $Y = 8$ and $Z = 0.130$, we must round off Z to $1/8 = 0.125$. The precision can be increased by increasing the length of SN.

Approximation Errors: These errors occur because not all functions can be realized exactly by SC. It is necessary to approximate them to something that is achievable. All stochastic function values are scaled to be in the interval [0, 1]. Without Rk's as inputs, stochastic functions using one variable that are synthesizable are only trivial case of X and $1-X$. So, functions like square root, squaring, sin(X) can be done by using Bernstein approximation.

Random Fluctuation Errors: These errors are caused because the N bit SN formed by using an RNG is pseudo-random in nature. These errors can be specified by calculating the mean square error. As the length of stochastic bit stream is increased, random fluctuation errors and rounding errors tend to decrease.

Constant-Induced Errors: These errors are caused because of the ancillary inputs. However, these errors can be reduced by transferring the ancillary inputs to sequential sub-circuits C so that their behavior can be traced. An algorithm CEASE is established to take out this error. CEASE replaces the ancillary inputs by sequential components equal to a mod counter with multiple moduli. It releases the weighted input bits when the counter-overflows.

Correlation Errors: Errors caused by correlation are due to intercommunication between the bit streams that occurs during computation which introduce dependencies and similarities between different bit streams. Correlation is also caused mainly due to three factors: RNS which are not able to produce good quality independent SNs, sharing of RNS to decrease the hardware cost of circuit and temporal dependencies introduced by sequential circuits. So, as the circuit size increases, correlation also increases. To quantify correlation, the stochastic computing (SCC) for a pair of SNs, X and Y, is defined as follows in Eq. 1:

$$\mathrm{SCC}(X,\ Y) = \begin{cases} \frac{p_{X \wedge Y} - p_X \cdot p_Y}{\min(pX,\ pY) - pXpY} & \text{if } p_{X \wedge Y} > p_X \cdot p_Y \\ \frac{p_{X \wedge Y} - p_X \cdot p_Y}{pXpY - \max(pX + pY) - 1,\ 0} & \text{otherwise} \end{cases} \quad (1)$$

where $P_{X\wedge Y}$ is $P(X(t) = 1, Y(t) = 1)$ for all t. Bit streams that are positively correlated, i.e., the 1s and 0s of a bit stream overlap with each other have SCC value as 1. Negatively correlated means that the 1s and 0s of the bitstreams under consideration do not overlap at all, and SCC value is -1. Non-correlated or independent bit streams are the ones where each bit stream is obtained from different Bernoulli sequences as the random number source giving SCC value as 0. There are several ways to avoid or manipulate correlation as below:

(a) **Regeneration**: In this method, the corrupted SN is converted into binary format and then given to a SNG to generate a new SN from it. This method has added to more hardware, more latency and desirable properties like progressive precision can be lost. [11]. Figure 5a shows a regeneration-based decorrelator.
(b) **Isolation**: In this method, D flip-flop is introduced at the bit stream frequency into the line containing corrupted SN. So, it delays the corrupted SN by one clock cycle. If the corrupted SN under consideration has bits which are independent, naturally, the new delayed version of the corrupted SN and the original SN are independent of each other. Figure 5b shows isolation-based decorrelator.
(c) **Shuffle-based Decorrelation**: Shuffling will re-randomize the bits in the corrupted SN thus helping reduce cross-correlation and autocorrelation. Complete elimination of correlation is not possible by this method because the newly generated SN will be correlated with original SN. Figure 5c shows an example of shuffle-based decorrelation.

(a) **Correlation Insensitive**: There are few examples of stochastic circuits which are not sensitive to correlation. It means, there is no need to bother about the auto- or cross-correlation in such circuits. It means, the circuit will function as desired even with correlated inputs. Such stochastic computing circuits are called as correlation-insensitive circuits.
(b) **Correlation Injection by Synchronizer and Desynchronizer**: Many stochastic circuits require uncorrelated inputs. But, circuits that implement functions which require the SN's to be either positively or negatively correlated. E.g.,

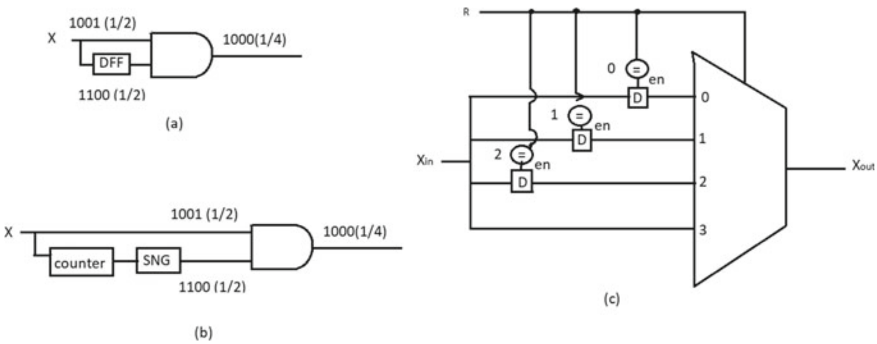


Fig. 5 **a** Regeneration-based decorrelator [11], **b** Isolation-based decorrelator [11], **c** Shuffle-based decorrelator [1]

to calculate the difference, an Ex-OR gate requires maximally correlated inputs. A type of SNG called synchronizer that can control the amount of correlation is used in such circuits. But, regenerating SNs with desired correlation levels between a stochastic system incurs a high hardware cost and introduces latency. Similarly, a desynchronizer can be used to decrease the correlation between the SN's. Errors due to autocorrelation can be managed by CEASE.

4 Optimization of Stochastic Circuits

Many attempts are being made to optimize the SC circuits with respect to various factors. The random number generator is one of the main blocks of SC which occupies more hardware area and needs to be optimized. The second part is correlation, which if managed properly can save a lot of hardware area, decrease latency, and increase the accuracy. The third part is the connecting wires between RNS and PCC. Various experimentations with respect to RNG have been done in various research papers until now. They are listed as below:

(i) Circular shifting of output of LFSR to be able to use a shared RNG [12].
(ii) Permutation of output of shared LFSR in reverse lexicographic order instead of circular shifting [13].
(iii) Use of S-box at the LFSR's output and then share it to multiple RNGs. This method increases hardware complexity as the S-box is a complex combinational circuit [14].
(iv) LFSR with nonlinear S-box function to show better autocorrelation and cross-correlation properties as compared to LFSR [15].
(v) Use of spintronic devices instead of LFSR to decrease the area and power consumed by the stochastic core [16].
(vi) Designing SNG's by sharing an RNS based on randomization function [17].

(vii) Use of LUT-based Quasi SNG where low discrepancy (LD) sequences are used to develop stochastic numbers [18]. Since LD sequences are uniformly spaced 1 s and 0 s, this SNG has better accuracy, convergence, throughput as compared to LFSR, and a less random fluctuations and execution time because of parallel implementation. The drawbacks include need for optimization with respect to area.

(viii) Bit flipping of output of LFSR that makes the generated SN's correlation insensitive and allows sharing of RNS [19]

Similarly, correlation can be managed by using correlation-insensitive circuits or correlation manipulative circuits like isolators, synchronizers, desynchronizer, decorrelators, regenerators, or judiciously selecting the seed of LFSR for generating RNS.

Considering above all design aspects, SC has been applied to applications where many arithmetic operations need to be done. This is because, simple logic gates are used to perform these operation, and hence, resultant decrease in the area and power is in 100 s as compared to traditional deterministic computing. Also, since the precision required in such applications is low, use of excessively long SN's can be avoided thus limiting the latency.

5 Advantages and Drawbacks

SC utilizes randomness as a valuable resource and converts probability values into another statistical bit stream. It lacks place value, i.e., no LSB or MSB as compared to deterministic computing. This makes the computation robust. Apart from this, other advantages of SC are small area, less power, error tolerant, supports parallelism; standard digital components can be used to realize the circuits. Its primary disadvantages are correlation induced inaccuracy, random number fluctuations, high latency; randomness sources are costly, and the theory is not yet fully understood. The design of a stochastic circuit involves complex trade-offs among accuracy, computing time, and hardware area cost.

6 Conclusion and Future Challenges

From the various methods to design a SNG described above, use of LD sequences for generating RNS gives higher throughput. Also, use of randomization function at the output of LFSR gives better correlation management. However, both of these methods claim additional area cost. So, use of either or both of above methods to design SNG without the addition of area or latency overhead need to be explored more for future work.

Also, automatic synthesis of optimally correlated deterministic RNS for design SNG's is also a challenge.

The domains to which SC can be successfully applied are artificial neural networks (ANNs), image processing, control systems, DSP, and decoding of modern error correcting codes [5]. Application of SC in implantable medical devices still remains a challenge for the researchers.

References

1. Gaudet VC, Gross WJ, Smith KC (2019) Stochastic computing: techniques and applications. Springer, Cham
2. Afuso C (1968) Analog computation by random pulse sequences. University of Illinois, Urbana, Illinois, Department of Electrical engineering
3. Poppelbaum WJ (1976) Statistical processors. In: Rubinoff M, Yovits MC (ed) Advances in computers, vol 14. Elsevier, Department of Computer Science University of Illinois Urbana, Illinois, pp187–230. Available: https://doi.org/10.1016/S0065-2458(08)60452-0
4. von Neumann J, Pierce RS (1952) Lectures on probabilistic logics and the synthesis of reliable organisms from unreliable components. Delivered at the California Institute of Technology, January 4–15, Caltech Eng. Library, QA. 267.V6
5. Alaghi, Hayes JP (2013) Survey of stochastic computing. ACM Trans Embed Comput Syst 12(2):1–19
6. Gupta PK, Kumaresan (1988) Binary multiplication with PN sequences. IEEE Trans Acoust Speech Sig Process 36:603–606
7. Sanath Potnuru SRPVS, Venkatesh TS, Agrawal S (2021) Design and implementation of efficient stochastic number generator. In: Fourth international conference on electrical, computer and communication technologies (ICECCT), pp 1–7
8. Perach B, Kvatinsky S (2020) An asynchronous and low-power true random number generator using STT-MTJ. In: 2020 IEEE international symposium on circuits and systems (ISCAS)
9. Van Daalen M, Jeavons P, Shawe-Taylor J, Cohen D (1993) Device for generating binary sequences for stochastic computing. Electron Lett 29:80
10. Karadeniz MB, Altun M (2017) Sampling based random number generator for stochastic computing. In: 2017 24th IEEE international conference on electronics, circuits and systems (ICECS), pp 227–230
11. Alaghi, Qian W, Hayes JP (2018) The promise and challenge of stochastic computing. IEEE Trans Comput-Aided Des Integr Circuits Syst 37(8):1515–1531
12. Ichihara H, Ishii S, Sunamori D, Iwagaki T, Inoue T (2014) Compact and accurate stochastic circuits with shared random number sources. In: 2014 IEEE 32nd international conference on computer design (ICCD), pp 361–366
13. Salehi SA (2019) Low-correlation low-cost stochastic number generators for stochastic computing. In: 2019 IEEE global conference on signal and information processing (GlobalSIP), pp 1–5
14. Neugebauer F, Polian I, Hayes JP (2018) S-box-based random number generation for stochastic computing. Microprocess Microsyst 61:316–326, Elsevier
15. Neugebauer F, Polian I, Hayes JP (2017) Building a better random number generator for stochastic computing. In: Euromicro conference on digital system design (DSD), pp 1–8
16. Angizi S, He Z, Bai Y et al (2018) Leveraging spintronic devices for efficient approximate logic and stochastic neural networks. In: Proceedings of 2018 great lakes symposium on VLSI (GLSVLSI'18). ACM, New York, p 6
17. Tawada M, Togawa N (2020) Designing stochastic number generators sharing a random number source based on the randomization function. In: 2020 18th IEEE international new circuits and systems conference (NEWCAS), pp 271–274

18. Seva R, Metku P, Choi M (2017) Energy efficient FPGA based parallel Quasi-stochastic computing. J Low Power Electr Appl
19. Abdellatef H, Khalil-Hani M, Shaikh-Husin N et al (2021) Low-area and accurate inner product and digital filters based on stochastic computing. In: Signal processing, vol 183

Dynamic Programming Approach to Solve Real-World Application of Multi-Objective Unbounded Knapsack Problem

Aayush P. Khandekar and Aniket Nargundkar

Abstract Knapsack problem is classified as a combinatorial optimization problem with the consideration of the optimal object being a part of the predefined set of finite objects allowed to be placed in the knapsack. The unbounded knapsack problem allows the repetition of objects, while demanding that the sum of the values of the objects in the optimal solution does not exceed the weight of the knapsack. In this paper, food order optimization problem is modeled as a multi-objective unbounded knapsack problem, as the problem has multiple objectives which need to be achieved simultaneously. Optimizing the number of given non-vegetarian dishes and maximizing the number of servings is considered as objective functions. These objectives are to be satisfied restricting to the budget constraint. Dynamic programming approach is applied to generate an optimal solution while satisfying the set constraints. The proposed approach successfully returns an optimal solution for all test cases.

Keywords Unbounded knapsack · Dynamic programming · Combinatorial optimization · Food order optimization · Multi-objective problem

1 Introduction

The concept of optimization has been around for decades, if not centuries. Optimization can be defined as the process of finding the best possible solution (also known as the optimal solution) of a problem with respect to the priorities set for a given problem. While solving an optimization problem, an objective function is encountered and the goal is to either maximize or minimize the objective function. For example, a football coach might want the team to run as much as possible, and

A. P. Khandekar (✉)
Vishwakarma Institute of Technology, Pune 411037, India
e-mail: akhandekar582@gmail.com

A. Nargundkar
Symbiosis Institute of Technology, Symbiosis International (Deemed University), Lavale, Pune 412115, India
e-mail: aniket.nargundkar@sitpune.edu.in

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_32

hence, the objective function is to maximize the running distance. Coach also needs to improve the team's passing accuracy. Needless to say, the team can only practice for a limited amount of time, and hence, the coach has to set some limits on how to improve the passing accuracy and still make the team run as much distance as possible to improve their stamina. These limits are called constraints.

The knapsack problem is a classic example of an optimization problem. In the knapsack problem, a list of items is presented with their associated values and the goal is to maximize the value of the knapsack while not exceeding the total weight capacity the knapsack can handle. There are different versions of knapsack such as 0-1 knapsack, unbounded knapsack [1], and fractional knapsack [2]. In the 0-1 knapsack problem, each item in the given list can be used only once. The unbounded knapsack is a version of the knapsack problem, in which the repetition of items is allowed. Hence, the number of permutations and combinations of the possible solutions is more as compared to that of the 0-1 knapsack. Sahni [3] presented a series of increasingly accurate algorithms which require polynomial time complexity and linear space storage to obtain approximate solutions to the 0-1 knapsack problem. Li et al. [4] obtained high-quality solutions by implementing the unbounded knapsack problem using genetic algorithm based on two techniques; heuristic operator which utilizes problem specific knowledge and preprocessing technique. Most of the time, the only objective of the knapsack problem is to maximize the value of the knapsack, and therefore, such problems can be classified as single-objective problems.

Multi-objective knapsack is a knapsack that has multiple objectives. Each objective is given a certain priority which helps in defining the importance of the objective. The objectives with the highest priority are completed first, and then, the lower priority objectives are set to be achieved. Food order optimization can be considered as a multi-objective unbounded knapsack problem. There are multiple objectives that can be considered in this problem which are presented in the methodology of this paper. The items present in the menu can be repeated while keeping the spending amount, i.e., budget same. Hence, it is classified as a multi-objective knapsack problem.

An optimization problem can be solved in many ways. There are different algorithms and approaches developed to solve such problems. Some socio-inspired optimization algorithms include cohort intelligence (CI) [5], particle swarm intelligence (PSO) [6], genetic algorithm (GA) [7], ideology algorithm (IA) [8], etc. In this paper, the concept of dynamic programming is applied to solve the food order optimization problem since it guarantees an optimal solution as the output. In dynamic programming, the complex problem is broken down into a collection of simpler subproblems which are solved once. Their solutions are stored to avoid repetitive computations. Dynamic programming has benefited the field of optimization immensely. It has been applied to various real-life applications such as multi-reservoir system optimization [9], blood platelet production [10], ship voyage optimization [11], inventory optimization [12] to note a few.

2 Methodology

Food order optimization can be categorized as a multi-objective unbounded knapsack problem since there are multiple factors that need to be optimized. Some of the parameters can be maximizing the number of servings, getting a specific number of non-vegetarian dishes, maintaining a specific number of calories, optimizing the dishes based on the time required to prepare them, etc. Such objectives are to be satisfied while maintaining the budget. The condition of under-utilization of the budget is an acceptable condition, whereas the over-utilization of the budget should not take place.

The food order optimization problem can be described as follows: given a set of I items presented in the menu, each item i, $i = 1, \ldots, I$ has been allocated with the number of servings s_i, cost of the item c_i, non-vegetarian parity n_i and vegetarian parity v_i, B is the total budget allocated for every order. The mathematical formulation of the problem is as follows. Equation 1 represents the objective function for non-veg dishes, while Eq. 2 is for vegetarian dishes. Equation 3 is the budget constraint. Equation 4 describes the calculation of cost associated with selected combination.

.

$$\text{Maximize } f_1(s) = s_1 n_1 + \ldots + s_I n_I, n_i \in \{0,\ 1\} \tag{1}$$

$$\text{Maximize } f_2(s) = s_1 v_1 + \ldots + s_{\mathrm{I}} v_{\mathrm{I}}, v_i \in \{0,\ 1\} \tag{2}$$

$$\text{Subject to } f(c) \le B \tag{3}$$

$$f(c) = (c_1 n_1 + \ldots + c_{\mathrm{I}} n_{\mathrm{I}}) + (c_1 v_1 + \ldots + c_{\mathrm{I}} v_{\mathrm{I}}) \tag{4}$$

In this problem, two objectives are considered which are, optimizing the number of given non-vegetarian dishes and maximizing the number of servings. These objectives are to be satisfied while optimizing the budget parameter. The budget should not exceed the given budget and if it does, the solution is disregarded and not considered as an optimal solution.

There are constraints set while solving this problem. The number of non-vegetarian dishes should neither exceed the desired amount nor should it be below the input provided. While meeting the number of servings objective, the minimum servings accepted is the number of servings which is provided. Any number above the provided input is accepted since the goal is to maximize the number of servings in the given budget. These constraints are mathematically presented in Table 1.

Table 2 presents a sample dataset used for testing the program. The program has been developed in Python using the dynamic programming approach. Optimizing the number of non-vegetarian dishes has been given a higher priority as compared to maximizing the number of servings. As a result, the program optimizes the number of non-vegetarian dishes as per the input provided. Once the required number dishes

Table 1 Mathematical modeling of constraints

Sr. No.	Objective	Variable notation	Constraint equation	Priority
1	Budget	B	B = input()	$P1$
2	Non-vegetarian dishes	N	N = input()	$P2$
3	Servings	S	$S \geq$ input()	$P3$

Table 2 Sample dataset

Item ID	Cost	Serves	Veg/non-veg
1	100	1	0
2	150	1	1
3	200	1	0
4	250	2	1
5	300	2	1

are finalized, the program shifts toward optimizing the second objective which is to maximize the number of servings. The program strictly selects only the vegetarian dishes since the required number of non-vegetarian dishes have been met. Optimizing the vegetarian dishes takes place within the budget reduced after satisfying the first objective.

3 Results and Discussions

The program developed for optimization using the dynamic programming approach in Python successfully provides an optimal solution for each of the test cases. The program has been tested on a sample dataset given in Table 2. Out of the several conducted tests, four test cases have been presented in Table 3. Every test case consists of two parameters, namely the number of non-vegetarian dishes (highest priority) and the minimum number of servings per order (second highest priority). A budget of 1000 units has been set. For the first test case, the program selects 8 dishes which satisfies the test case. It strictly selects the best item id which satisfies the condition for the number of non-vegetarian dishes while keeping a track of the number of servings. A total of 8 servings have been presented in the first test case which satisfies the servings objective (minimum 5 servings). In terms of the spending, the program spends a total of 950 units since any increase in the order would have resulted in the violation of the total budget set. Similarly, test cases two and three prove that the program has produced an optimal solution while considering the set constraints. As for the test case 4, the program does not output a solution since it is impossible to select 5 non-vegetarian dishes without violating the budget constraints, i.e., spending over the budget. In order to solve test case 4, it is evident that there should be an increase in the budget or there should be a reduction in the cost of the

Table 3 Solution with dynamic programming approach

Sr. No.	Test Case		Total spending	Total servings	Total items selected	Item ID	Cost breakout	Servings breakout
	Non-vegetarian dishes	Number of servings						
1	1	5	950	9	8	4	250	2
						1	100	1
						1	100	1
						1	100	1
						1	100	1
						1	100	1
						1	100	1
						1	100	1
2	2	7	1000	9	7	4	250	2
						4	250	2
						1	100	1
						1	100	1
						1	100	1
						1	100	1
						1	100	1
3	3	5	950	8	5	4	250	2
						4	250	2
						4	250	2
						1	100	1
						1	100	1
4	5	5	0	0	0	NA	NA	NA

items. Regardless, the program provides an accurate answer in test case 4, which is, that an optimal solution cannot be obtained.

4 Conclusion

In this paper, a real-world example of a multi-objective unbounded knapsack problem has been solved. The solution of this problem is obtained using the dynamic programming approach. It is evident from the obtained results that the program developed generates an optimal solution for all the test cases. The results have been verified based on the set constraints in the problem. The program does not violate any constraints and outputs an optimal solution. Furthermore, it strictly focuses on the higher priority tasks first, i.e., correctly selecting the exact number of non-vegetarian dishes. It does so while still optimizing the non-vegetarian dishes in terms of the cost and servings. Then, the program proceeds to complete the lower priority task while still satisfying the number of servings objective.

In the near future, similar problems can be solved using advanced optimization algorithms. A goal programming approach can also be developed, in which the model strictly focuses on the values given in the dataset for the input for the deviational variables.

References

1. Kellerer H, Pferschy U, Pisinger D (2004) The unbounded knapsack problem. In: Knapsack problems. Springer, Berlin, Heidelberg, pp 211–234
2. Ishii H, Ibaraki T, Mine H (1977) Fractional knapsack problems. Math Program 13(1):255–271
3. Sahni S (1975) Approximate algorithms for the 0/1 knapsack problem. J ACM (JACM) 22(1):115–124
4. Li K-L, Dai G-M, Li Q-H (2003) A genetic algorithm for the unbounded knapsack problem. In: Proceedings of the 2003 international conference on machine learning and cybernetics (IEEE Cat. No.03EX693), vol 3, pp 1586–1590. https://doi.org/10.1109/ICMLC.2003.1259749
5. Kulkarni J, Durugkar IP, Kumar M (2013) Cohort intelligence: a self supervised learning behavior. In: 2013 IEEE international conference on systems, man, and cybernetics, pp 1396–1400. https://doi.org/10.1109/SMC.2013.241
6. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95—international conference on neural networks, vol 4, pp 1942–1948. https://doi.org/10.1109/ICNN.1995.488968
7. Mirjalili S (2019) Genetic algorithm. In: Evolutionary algorithms and neural networks. Studies in computational intelligence, vol 780. Springer, Cham. https://doi.org/10.1007/978-3-319-93025-1_4
8. Huan TT, Kulkarni AJ, Kanesan J et al (2017) Ideology algorithm: a socio-inspired optimization methodology. Neural Comput Appl 28:845–876. https://doi.org/10.1007/s00521-016-2379-4
9. Li X et al (2014) A parallel dynamic programming algorithm for multi-reservoir system optimization. Adv Water Resour 67:1–15
10. Haijema R, van der Wal J, van Dijk NM (2007) Blood platelet production: optimization by dynamic programming and simulation. Comput Oper Res 34(3):760–779
11. Zaccone R, Ottaviani E, Figari M, Altosole M (2018) Ship voyage optimization for safe and energy-efficient navigation: a dynamic programming approach. Ocean Eng 153:215–224
12. Karimi M, Sadjadi SJ (2022) Optimization of a multi-Item inventory model for deteriorating items with capacity constraint using dynamic programming. J Indus Manag Optim 18(2):1145

Probabilistic Harmony Search Algorithm: Fitness Proportionate Selection Variants

Ravi Yadav, Sumanth Vullamparthi, Aashi Tapadia, Anand J. Kulkarni, and Preeti Kale

Abstract Harmony search algorithm is a meta-heuristic, nature-inspired optimization algorithm that tries to mimic real-life improvisations that musicians use to generate a harmony that is more pleasing to hear. This paper presents and compares three different types of harmony search algorithms. We start with the implementation of the original HSA. Further, two different modifications were made to the original HSA, the first one uses fitness proportionate selection of harmonies from the HM and is known as biased Roulette harmony search algorithm (BRHSA) and the second one builds on BRHSA by adding simple mathematics to further guide the algorithm toward the desired solution and is called guided biased Roulette harmony search algorithm (GBRHSA). These three variants of HSA were applied on four benchmark test functions on the same machine, and the results obtained after 30 runs were noted down for comparison. It was observed that the results given by the three variants had no specific trend in terms of best result or computational time and the performance of a particular variant was subject to parameters like the kind of function and its search domain. The results presented in this paper can be used as a foundation for the future works that will be done on this algorithm and can help derive an apt variant of HSA for solving a particular problem.

R. Yadav (✉) · S. Vullamparthi · A. Tapadia · P. Kale
School of CET, Dr Vishwanath Karad MIT World Peace University, Pune, India
e-mail: 1032180592@mitwpu.edu.in

S. Vullamparthi
e-mail: 1032180741@mitwpu.edu.in

A. Tapadia
e-mail: 1032180622@mitwpu.edu.in

P. Kale
e-mail: preeti.kale@mitwpu.edu.in

A. J. Kulkarni
Institute of Artificial Intelligence, Dr Vishwanath Karad MIT World Peace University, Pune, India
e-mail: anand.j.kulkarni@mitwpu.edu.in

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_33

Keywords Harmony search algorithm · Fitness proportionate selection · Biased Roulette wheel · Guided biased Roulette wheel · Benchmark test functions · Comparison

1 Introduction

Optimization algorithms are highly efficient algorithms that aim to find the solution to complex problems like traveling salesman problem, scheduling problems, etc. There are many mathematical optimization techniques like linear programming, dynamic programming, existing that have been used for solving many optimization problems. But there are a few drawbacks to these techniques like, many of these fails when the number of variables increases, or when the optimization function is non-differentiable and these also take a lot of memory and time for execution. Hence, to overcome these limitations, another set of optimization algorithms have been developed which have been inspired from nature. These algorithms are called nature-inspired algorithms or heuristic algorithms [1]. These algorithms have been proven to find solutions to high dimensional optimization problems in lesser time and using a lower memory space. Here are some of the most commonly used nature-inspired algorithms—genetic algorithm [2–4], simulated annealing [5], ant colony optimization algorithm [6–8], bees algorithm [9–11], etc. One such algorithm that has recently gained popularity in solving real-world problems is harmony search algorithm (HSA) [1, 12–15]. HSA has been widely used to solve various real-world problems some of them include: laying down of water pipelines in the city of Hanoi [1], structural engineering [16], threshold segmentation [15], vehicle routing, and university timetable preparation.

This algorithm has been inspired from the musicians in an orchestra who try to improvise a harmony which is pleasant to human ears when heard, i.e., it is an esthetically pleasant harmony. The parallels that can be drawn between finding the right synchronous harmony and solving an optimization problem is esthetically pleasant harmony is the solution of the optimization problem that we are trying to reach. There can be any number of instruments in the orchestra that are being harmonized. These instruments can be thought of as the decision variables of the optimization problem. We will see the detailed working of the algorithm in the coming sections.

This paper first provides our implementation of the original HSA. It also highlights the limitations of the original implementation and then goes on to describe our modification to the algorithm. It further discusses the results obtained by applying the original HSA and the two variants on four test functions for 30 runs followed by a conclusion that summarizes the intent and the flow of the paper.

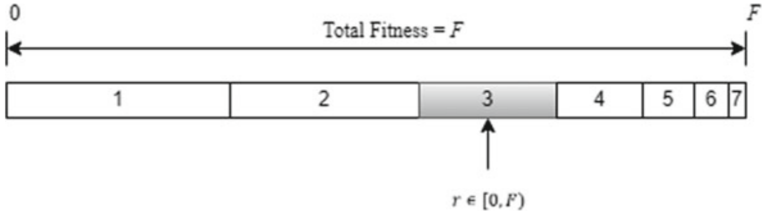


Fig. 1 Fitness proportionate selection

2 Methodology

In this section, we discuss the modification to the traditional HSA and how these modifications were made. The traditional HSA has been replicated the way it is done in the original experiments [12]. In our implementation, we intend to replace the traditional method of selecting harmonies from the harmony memory (HM) in a more probabilistically optimum way for reasons mentioned in Sect. 3.2. This modification has been inspired from the Roulette wheel selection method.

2.1 *Fitness Proportionate Selection*

Fitness proportionate selection can be best explained using the example of a Roulette wheel. A Roulette is a casino game which has a wheel having 37 pockets of equal size. This indicates that there is an equal probability that any of the pockets gets selected after every play a person makes.

The Roulette wheel selection is a similar idea but in this case, the probability of a pocket being selected is always not the same. Given below is a representation of a probability array (Fig. 1).

Fitness is assigned to possible solutions by fitness function. The fitness level is used to associate a probability of selection with each possible solution [17]. A part of the wheel is assigned to each of the possible selections based on their fitness value. This is done by normalizing them to 1 by using the formula given in pseudo-code. Then, a random selection is made similar to how the Roulette wheel is rotated.

2.2 *Biased Roulette Wheel HSA (BRHSA)*

The idea behind the biased Roulette wheel HSA or BRHSA method is that, while selecting the next harmony from the harmony memory, the probability of selecting a good harmony should be greater as compared to probability of selecting a bad harmony. Since the harmony memory is sorted in such a way that the most optimum solution always occurs on top and the worst solution lies at the bottom of the HM, it is clear that fitness proportionate selection can be implemented to achieve probabilistic optimum selection of better harmonies in every iteration. Each row in the HM is assigned a probability threshold value which is stored in the array roulette_prob. The fitness function used to calculate these probabilities is given below

$$P_i = \frac{\sum_{i=0}^{\mathrm{HMS}} F_i}{F_i}$$

For every decision variable d, when HMCR is compared with a random number and when it is greater than the random number, another random number is generated which is then compared with the elements of roulette_prob to get the value of i such that rouletterProb $[i + 1] > r \geq$ rouletteProb$[i]$. This i is then used to select the HM$[i][d]$ element which becomes a part of the new harmony.

2.3 Pseudo-Code for BRHSA

Step 1: *Parameter setting (HMS, HMCR, NOI).*

Step 2: *Initializing the HM using below given equation and calculating fitness function value of each harmony vector*

$$X_i(k) = \mathrm{lowerBound}_k + r \times \left(\mathrm{upperBound}_k - \mathrm{lowerBound}_k\right)$$

where $r \in (0,\ 1)$, for $i = 1,\ 2 \ldots \mathrm{HMS};\ k = 1,\ 2,\ 3, \ldots d$ *where d is the number of decision variables of the function.*

Step 3: *Sort the HM based on function values.*

Step 4: *Finding the next harmony vector (set of decision variables) to find the solution*

For ($k = 1$ to d)

Generate a random number r between 0 and 1

If r > HMCR:

$$\text{Calculate } X_{\mathrm{new}}(K) = \mathrm{lowerBound}_k + r \times \left(\mathrm{upperBound}_k - \mathrm{lowerBound}_k\right)$$

Else:

Assign probabilities to all harmony vectors using below given equation and store them in a roulette_array

$$P_i = \frac{\sum_{i=0}^{\text{HMS}} F_i}{F_i}$$

After getting this array for minimization change all values using formula $(1 - P_i)/(N - 1)$ *(N−1 is to normalize the values between 0 and 1)*

Generate a random number r and find the corresponding probability interval in roulette_array and select the $X(K)$ *from HM having the same index value.*

End for

Step 5: *If newly generated harmony has better function value than worst function value in HM. Then, remove the worst harmony vector and insert this new vector in HM such that it remains sorted.*

Step 6: *Repeat step 4 and 5 NOI times.*

Step 7: *The first index of the sorted HM is the solution to the problem.*

2.4 Guided Biased Roulette HSA (GBRHSA)

In GBRHSA, a modification is made to BRHSA as all the values in HM converge after a certain number of Iterations, so once the values of HM have converged, new values of decision variables are calculated by subtracting and adding a very small proportion from the best values of decision variables already present in HM. Then, all possible combinations of these decision variables are used to find function values. The combination with least function value is selected as the best harmony vector and is inserted in the HM. This process is repeated till all values of HM are different but better.

2.5 Pseudo-Code for GBRHSA

Step 1: *Parameter setting (HMS, HMCR, NOI).*

Step 2: *Initializing the HM using below given equation and calculating fitness function value of each harmony vector*

$$\begin{aligned} X_i(k) = {} & \text{lowerBound}_k + r \\ & \times \left(\text{upperBound}_k - \text{lowerBound}_k\right) \end{aligned}$$

where $r \in (0,\ 1)$, for $i = 1,\ 2 \ldots \text{HMS};\ k = 1,\ 2,\ 3,\ \ldots d$ *where d is the number of decision variables of the function.*

Step 3: *Sort the HM based on function values.*

Step 4: *Finding the next harmony vector (set of decision variables) to find the solution*

For ($k = 1$ to d)

Generate a random number r between 0 and 1

If r > HMCR:

$$\text{Calculate } X_{\text{new}}(K) = \text{lowerBound}_k + r \times \left(\text{upperBound}_k - \text{lowerBound}_k\right)$$

Else:

$$\textbf{If}\,(\text{second last value of HM} == \text{last value of HM})\;:$$

Find new values of decision variables by subtracting and adding a very small fraction of actual value and find function values with all permutations possible, select the combination of decision variables which gave least function value.

Else:

Assign probabilities to all harmony vectors using below given equation and store them in a roulette_array

$$P_i = \frac{\sum_{i=0}^{\text{HMS}} F_i}{F_i}$$

After getting this array for minimization change all values using formula $(1 - P_i)\big/(N - 1)$ (*N−1 is to normalize the values between 0 and 1*)

Generate a random number r and find the corresponding probability interval in roulette_array and select the $X(K)$ *from HM having the same index value.*

End for

Step 5: *If newly generated harmony has better function value than worst function value in HM. Then, remove the worst harmony vector and insert this new vector in HM such that it remains sorted.*

Step 6: *Repeat step 4 and 5 NOI times.*

Step 7: *The first index of the sorted HM is the solution to the problem*

3 Results

3.1 Experimental Setup

To evaluate the performance of the HSA, code is in Python language, and all the tests are run on a PC with an Intel(R) Core (TM) i5 − 10300H CPU @ 2.50GHz$\big/$8.00 GB(7.87 GB usable) of RAM under Microsoft Windows 11.

3.2 Comparison

In the original HSA, there were many iterations that made no change to the harmony memory since the random generator failed to produce a better value to create a better harmony vector than the best harmony already present in the HM. This problem is being tackled by BRHSA, which assigns fitness proportionate probabilities to each harmony vector of HM which helps in selecting a best value of HM with more probability.

Also, in BRHSA, after a certain number of iterations, the HM values converge, making it difficult for the random function to generate better value, so to overcome this problem, the HM is guided for next ten iterations by finding a better harmony using simple mathematics. This guidance helps HM to find a better solution in less iterations and has been termed as GBRHSA.

In order to understand the best parameters suitable for our experiments, we did a grid search accuracy check which is inspired from sklearn's grid search CV. The key performance indicators used to evaluate them were deviation from actual minima (primary) and time taken to reach minima (secondary). We chose the experiment parameters on a linear scale. For original HSA, the trial parameters used to evaluate are as follows:

(a) Harmony memory size: [10, 50]
(b) HMCR: [0.45, 0.50, 0.60, 0.65]
(c) PAR: [0.05, 0.1, 0.15, 0.2, 0.25]
(d) bandwidth: [0.1, 0.2, 0.3]

For BRHSA and GBRHSA, the experiments were done using the same parameter lists. They are listed below:

(a) Harmony memory size: [10, 50]
(b) HMCR: [0.2, 0.25, 0.3, 0.35, 0.4]

In case of BRHSA and GBRHSA, since we wanted a greater number of iterations to pick the values from within the HM, the value of HMCR was kept lower as compared to the original HSA. The final chosen parameters are discussed in Table 2.

Table 1 shows the benchmark functions used for comparison along with their search domain and global optimum for these functions. The values for HMS and number of Iterations (NOI) were kept as 10 and 100,000 and were constant for all three variants. For the implementation of original HSA, HMCR, PAR, and bw were kept as 0.6, 0.1, and 0.2, respectively. For BRHSA and GBRHSA, the value of HMCR was kept as 0.25 and 0.3.

Tables 3 and 4, Tables 5 and 6, Tables 7 and 8, and Tables 9 and 10 show the different results we got after we applied all three variants for three and five variables on sphere function, Ackley function, Griewank function, and Rosenbrock function, respectively.

Table 1 Test problems

Name of the function	Dimension	Search domain	Global optimum
Sphere function	2, 5	[−5.12, 5.12]	0
Ackley function		[−15,30]	0
Griewank function		[−600,600]	0
Rosenbrock function		[−5,10]	0

Table 2 Parameter values used

Name	HMCR	PAR	Bandwidth	HM size	Number of iterations/evaluations	Total number of runs for each function
Original HSA	0.6	0.1	0.2	10	100,000	30
BRHSA	0.25	NA	NA			
GBRHSA	0.3	NA	NA			

Table 3 Results obtained for sphere function for two variables

Algorithm	Best minima/solution	Average execution time(s)	Mean error for 30 runs
Original HSA	1.13E−08	39.21	2.62E−07
BRHSA	5.50E−09	36.96	5.73E−07
GBRHSA	0.00E−01	41.39	0.00E−01

Table 4 Results obtained for sphere function for five variables

Algorithm	Best minima/solution	Average execution time(s)	Mean error for 30 runs
Original HSA	4.07E−06	38.01	4.44E−05
BRHSA	9.38E−04	50.21	1.11E−02

Table 5 Result obtained for Ackley function for two variables

Algorithm	Best minima/solution	Average execution time(s)	Mean error for 30 runs
Original HSA	9.14E−04	41.05	6.28E−03
BRHSA	1.14E−03	43.99	8.09E−03
GBRHSA	2.13E−14	39.03	2.79E−14

Table 6 Result obtained for Ackley function for five variables

Algorithm	Best minima/solution	Average execution time(s)	Mean error for 30 runs
Original HSA	2.35E−02	41.22	8.74E−02
BRHSA	7.10E−01	34.41	2.17

Table 7 Results obtained for Griewank function for two variables

Algorithm	Best minima/solution	Average execution time(s)	Mean error for 30 runs
Original HSA	5.82E−05	41.34	3.60E−03
BRHSA	1.98E−04	37.56	4.24E−03
GBRHSA	2.22E−16	38.41	8.89E−02

Table 8 Results obtained for Griewank function for five variables

Algorithm	Best minima/solution	Average execution time(s)	Mean error for 30 runs
Original HSA	0.0085	40.27	0.036
BRHSA	0.04	37.42	0.25

Table 9 Results obtained for Rosenbrock function for two variables

Algorithm	Best minima/solution	Average execution time(s)	Mean error for 30 runs
Original HSA	1.33E−04	39.41	3.60E−03
BRHSA	4.31E−06	36.01	4.24E−03
GBRHSA	3.53E−04	36.03	8.89E−02

Table 10 Results obtained for Rosenbrock function for five variables

Algorithm	Best minima/solution	Average execution time(s)	Mean error for 30 runs
Original has	4.52E−04	40.54	1.79E−01
BRHSA	9.08	35.26	80.86

The results obtained by us were also compared with a few other variants mentioned in [17] and the comparison is displayed in Table 11. The first three rows are our implementations with the same conditions and search domains as mention in the paper followed by various others mentioned in [17].

In [17], the individual simulation is repeated 50 times and each simulation performs 50,000 function evaluations for each problem. Parameters like HMCR, PAR, and bandwidth along with the system architecture used for running these simulations are not mentioned in the paper.

4 Conclusions and Future Work

The study presents the results obtained after performing optimization on four different unconstrained optimization problems. A few observations related to GBRHSA are as follows:

Table 11 Comparison with other variants for two variables

Algorithm	Ackley function	Rosenbrock function	Griewank function
Our simple HSA	1.58E−06	1.47E−05	9.19E−04
BRHSA	1.52E−03	1.93E−04	2.71E−05
GBRHSA	2.13E−14	1.15E−02	7.50E−03
Simple HSA	5.22E−05	5.10E−08	1.97E−10
PSF-HS	1.57E−04	3.21E−08	2.39E−09
APF-HS	1.57E−04	3.78E−08	3.46E−09
SGHSA	1.01E−06	0.00E+00	1.17E−13
NSHS	1.25E−06	0.00E+00	2.23E−13
PAHS	1.55E−02	4.77E−03	7.76E−06

1. GBRHSA has good results with a solution closer to the ideal solution because of its nature of guided minimization.
2. It performs exceptionally well for sphere function because of the exponential nature of sphere function.
3. The execution time of GBRHSA is slightly more than BRHSA because of additional math calculation that is used to guide the algorithm but it generally remains lower than the original HSA with sphere function being the only exception.
4. In case of the Griewank function using GBRHSA, although the best value is of the order E−16, the error is observed to be of the order E−02. This can be attributed to high unpredictability of the values that the HM is initialized with and where all the values of HM converge because of the large search domain. This causes a few values to be much better than the others and affects the mean error.
5. One limitation of GBRHSA that could be noticed was its inability to handle problems with higher dimensions. GBRHSA subtracts a fixed value from the variables picked up and generates a combination matrix of all possible combinations which are further evaluated to find the best combination. This method works fine for two variables as there are only four possible combinations, but as the number of dimensions or variables increase, the number of possible combinations also increases and evaluating all of them and finding the best one can be computationally expensive and become an optimization problem in itself.

A few observations related to BRHSA are as follows:

1. The slightest difference in execution time of a few functions in BRHSA is because weighted probabilities have to be assigned every time to the HM.
2. The anomaly in case of Griewank function results can be again justified based on the large search domain and is discussed above in detail.

We see that original HSA and BRHSA are able to converge to minima even for five-dimensions, and hence, we can conclude that they are robust and can be used for higher dimensionality optimization problems by writing a more optimized code.

This paper has a few limitations which can be addressed in the future works. In the GBRHSA, the guidance has been done by decreasing or increasing the current value in the memory by a constant value. A smarter guidance algorithm with a proper stopping criterion will help the algorithm to converge to a more optimum solution and also stop when there is no scope of getting a better solution, thus reducing the number of iterations the optimum solution is found in.

References

1. Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76(2):60–68
2. Lingaraj H (2016) A study on genetic algorithm and its applications. Int J Comput Sci Eng 4:139–143
3. Amjad MK, Butt SI, Kousar R, Ahmad R, Agha MH, Faping Z, Anjum N, Asgher U (2018) Recent research trends in genetic algorithm based flexible job shop scheduling problems. Math Probl Eng 2018(9270802):32. https://doi.org/10.1155/2018/9270802
4. Kumar SR, Singh KD (2021) Nature-inspired optimization algorithms: research direction and survey
5. Suman B (2004) Study of simulated annealing-based algorithms for multiobjective optimization of a constrained problem. Comput Chem Eng 28:1849–1871. https://doi.org/10.1016/j.compchemeng.2004.02.037
6. Dorigo M, Birattari M, Stützle T (2006) Ant colony optimization. Comput Intell Mag IEEE 1:28–39. https://doi.org/10.1109/MCI.2006.329691
7. Pei Y, Wang W, Zhang S (2012) Basic ant colony optimization. In: 2012 International conference on computer science and electronics engineering, pp 665–667. https://doi.org/10.1109/ICCSEE.2012.178
8. Fahad LG, Tahir SF, Shahzad W, Hassan M, Alquhayz H, Hassan R (2020) Ant colony optimization-based streaming feature selection: an application to the medical image diagnosis. Sci Program 2020(1064934):10
9. Karaboga D, Basturk B (2008) On the performance of artificial bee colony (ABC) algorithm. Appl Soft Comput J 8:687–697
10. Sharma A, Sharma A, Choudhary S (2020) A review on artificial bee colony and it's engineering applications. J Crit Rev 7(11)
11. Singh P (2016) A systematic review on artificial bee colony optimization technique. Int J Control Theory Appl 9:5487–5500
12. Askarzadeh A, Rashedi E (2018) Harmony search algorithm: basic concepts and engineering applications. https://doi.org/10.4018/978-1-5225-5643-5.ch001
13. Kim JH (2016) Harmony search algorithm: a unique music-inspired algorithm. Procedia Eng 154:1401–1405. https://doi.org/10.1016/j.proeng.2016.07.510
14. Geem ZW (2009) Music-inspired harmony search algorithm. Stud Comput Intell 191
15. Oliva D, Cuevas E, Pajares G, Zaldivar D, Perez-Cisneros M (2013) Multilevel thresholding segmentation based on harmony search optimization. J Appl Math 2013(575414):24. https://doi.org/10.1155/2013/575414
16. Saka M, Aydogdu I, Hasançebi O, Geem ZW (2011) Harmony search algorithms in structural engineering. https://doi.org/10.1007/978-3-642-20986-4_6
17. Choi YH, Eghdami S, Ngo TT, Chaurasia SN, Kim J-H (2019) Comparison of parameter-setting-free and self-adaptive harmony search

Intelligent Warehouse Automation Using Robotic System

Pranav Dixit, Aniket Nargundkar, Pranjal Suyal, and Rushil Patil

Abstract In the past couple of decades, the applications of industrial robotics in the industries have seen exponential growth due to the advancements in technologies. The concept of Industry 4.0 is the next generation manufacturing system which is a confluence of smart sensors, electric and electronics technologies, and core manufacturing technologies. Warehouse automation plays a critical role to meet today's demanding goals of supply chains, and warehouse automation technologies are required to be unprecedentedly efficient, adaptive, scalable, and fault-tolerant. This paper presents an application of AI-based smart robotics system for intelligence warehouse automation. In the current work, the territory of the warehouse is explored by the robots. AI-based path planning algorithms, viz. rapidly exploring random trees (RRT), RRT*, A* are applied for the path planning of robots in the warehouse. The results show the potential application of robotics systems in the domain of intelligent automation of warehouses.

Keywords A* algorithm · RRT algorithm · RRT* algorithm · Path planning · Warehouse automation

P. Dixit · A. Nargundkar (✉) · P. Suyal · R. Patil
Symbiosis International (Deemed University), Symbiosis Institute of Technology, Lavale, Pune 412115, India
e-mail: aniket.nargundkar@sitpune.edu.in

P. Dixit
e-mail: pranav.dixit.btech2018@sitpune.edu.in

P. Suyal
e-mail: pranjal.suyal.btech2018@sitpune.edu.in

R. Patil
e-mail: rushil.patil.btech2018@sitpune.edu.in

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_34

1 Introduction

The rapid growth of e-commerce industry in the past decade have forced the warehouses worldwide to be equipped with newer and more advanced automation technologies. The techniques such as same day shipment, omni-channel retail business, complex ordering, etc., are adopted to meet the growing demands. To meet the demanding goals of these trends, it is critical that warehouse automation technologies be unprecedentedly efficient, adaptive, scalable, and fault-tolerant. There are two types of warehouse automation, viz. physical automation and digital automation. Physical automation system refers to the application of embedded systems such as robots, whereas digital automation includes the application of cyber-physical systems and ERP-based systems [2, 4].

Recently, robot-based smart automation of the warehouse has gained the attention of researchers due to the latest technological advancements in the field of robotics. Automated guided vehicles (AGVs), automated storage and retrieval systems (AS/RS), "Cobots" or collaborative robots, and robotic arms are some of the prominent examples of robotics systems used for warehouse automation. The AGVs and human–robot system have been proposed by Rey et al. [16]. Koster [12] proposed the architecture for fully automated warehouse with robots, AGVs, AS/RS, and cyber-physical systems (CPS). Poudel [14] described the Kiva system for managing the 100 autonomous robots in the warehouse. A dual-loop optimal hierarchical control scheme for robotic manipulators for warehouse automation has been presented by Prakash et al. [15]. Grey et al. [6] proposed a blockchain-based robotics CPS for intelligent warehouse automation. Belotserkovsky et al. [3] described the multi-robot system based on the visual navigation method. Javaid et al. [10] presented the applications of autonomous robots for warehouse automation in the domain of Industry 4.0.

This paper presents the application of robots using AI-based path planning algorithms. Three widely applied algorithms, viz. RRT, RRT*, and A* are applied [8]. The remainder of the paper as follows: Sect. 2 describes the methodology adopted in the paper. All the three algorithms are explained in this section. Results along with the plots are discussed in Sect. 3. The conclusions and future directions are presented in Sect. 4.

2 Methodology

In a swarm-based robotic-based system, every robot needs to move from one point to another. For achieving this, some techniques which will be able to enable the movement of the robots from start to the goal destination without colliding with each other and avoiding the obstacles in between must be devised. The motion and behaviour of the robot must be planned beforehand so that during the movement, the bots are able to travel without any discrepancy. Motion planning helps to compute the optimal path from the start point to the destination point. Between the two points, there could be many obstacles and obstructions which the bot needs to avoid. There are different kinds of algorithms that can be used to enable motion planning and obstacle detection and avoidance; e.g. probabilistic roadmap method, A* algorithm, RRT, and RRT* method. Additionally, we also need to design the mechanism and the components which are needed to enable the proper motion planning, which include infrared sensors, cameras, and gyroscopes which will detect and send the inputs to the controllers for providing necessary signals to enable proper movement of swarm bots [4].

Path designing lets an autonomous vehicle or a robot realize the shortest and most obstacle-free path from the beginning to the goal state. The trail may be a collection of states (position and orientation) or waypoints. Path planning needs a map of the setting alongside start and goal states as input. The map may be depicted in numerous ways like grid maps, state areas, and topological roadmaps. Path planning techniques embody two major kinds of algorithms for autonomous vehicles. Grid-based search algorithms realize a path based on minimum travel value in an exceedingly grid map which are used for applications like mobile robots in a 2D environment. However, the memory demand to implement grid-based algorithms increases with the number of dimensions [5].

Robotic Path Planning: RRT and RRT* Algorithms

Exploring the optimized version of an orthodox path planning algorithm, robot motion or path planning is one of the most important aspects of robot control. It helps us to plan the movement and the motion of the robot. Our aim is to avoid all obstacles while maintaining a proper path. The path can be predetermined or the robot can dynamically adjust its path according to the environment or application. There are many challenges which are needed to overcome before planning and developing a suitable algorithm for the proper control of robots. There are many path planning algorithms such as rapidly exploring random trees (RRT), RRT*, and A* algorithm.

RRT Method/Algorithm

One of the most commonly used algorithms for robot path planning is rapidly exploring random tree method (RRT). In the RRT method, points are generated randomly in the workspace or the movement area. These points are then connected to the nearest available point or node. Each time a line or a curve is created; it is checked that the node lies outside or away from the obstruction or the obstacle. These points should be joined in such a way that the line joining these points always go around or separate away from the obstacle, else the algorithm will be faulty and the bot may crash into an obstacle. In a similar way, many points and nodes are created till one of the nodes reaches the goal of the endpoint. As soon as the points reach the goal point, the algorithm is completed. In order to generate the above-mentioned points or dots, a simple random number generator can be used. These random dots can be generated anywhere near the closest node which will eventually reach the goal. These random no generators are sufficient enough and satisfy most of the requirements, but in some cases, these may not be fully random and may contain some bias. Depending on the type of application and the simulation time, various methods to connect the nodes can be adopted. The type of graphs that RRT produces is cubic. The newly generated points or nodes are joined with the neighbouring nodes. This method is effective for the most part but sometimes, the structure and pattern created by the RRT method leads to the less efficient path, and the path distances are very long and the waypoints or the path formed is irregular sometimes. These problems can be solved easily by using the RRT* algorithm which will be discussed in the upcoming topics. RRT methods have many advantages which make them a choice for many applications. The main benefit of this method is its speed and implementation. It is very easy to implement and the calculations can be done in very little time [11, 13].

RRT* Algorithm

The RRT* method is another method that is used for the path planning algorithm of the robotic system. This algorithm is similar to the rapidly exploring random tree algorithm but it includes some changes which give it some features which are better than the simple random tree method. The aim of this algorithm is to develop the shortest path between the two points. There are some additions made to the algorithm which makes it more optimized than the other methods. In RRT*, there is a factor called the cost of the vertex which is determined by the distance that every vertex has travelled as compared to its parent vertex. First, the closest node or point is found in the graph and all the vertices at a particular distance or area around the new node are analysed. If any node in the area has a cheaper cost than the previous node, the cheaper node replaces the previous node and becomes the main node. This leads to the formation of a tree-like structure and the nodes continue to branch out from the ends till the goal is found and the most optimal path is found. The basic structure that is created in the RRT* is made keeping in mind the optimization aspect of the algorithm. After the above-explained process of finding out the cost of each node and vertex and replacing them with the cheaper nodes, the vertices are connected to these cheaper nodes. The vertices and nodes are again checked after connecting them, in order to determine if the cost is affected in any way. If there is any increase in cost, the neighbouring node is again connected to the node with the cheapest cost. This process continues until the paths become smooth and optimized. RRT* leads to the creation of straight paths. RRT* is especially useful in cases where the object density is too high and too many obstacles need to be avoided. The graphs formed in RRT* are very different from RRT because the graph is already optimized and does not need more nodes. This leads to shorter path lengths in RRT*. If the destination is changed, RRT* still results in the shortest and the quickest path to all points or destinations in the graph region. Although RRT* is overall a well-optimized and accurate method, it faces some problems such as there is a significant decrease in performance when it comes to time taken. As the nodes are examined and scanned multiple times, it takes nearly eight times longer than the RRT method. Due to this, better hardware is needed to implement RRT* method for quick calculations. A significant amount of the computational power is used in the object detection and avoidance part of the algorithm. Object avoidance is checked; every time a node is joined together or connections are made to ensure that there is no object collision and the surrounding area is free of objects. Doing this in real-time leads to significant processing power and time. But the final result is well-optimized and accurate paths that can be used to guide the robot [1, 9].

A* Algorithm

A* algorithm is a type of informed pathfinding algorithm and is one of the most famous algorithms used to find the shortest path between two points. A* algorithm uses three parameters to find the next correct node $G(n)$; this is the cost required to reach a given node from the previous node. It also takes into account all the previous costs of the nodes visited before ending on the final node. $H(n)$; it is known as the heuristic value of the next cell. The programme estimates a certain cost to reach the next cell as the actual cost cannot be calculated till the time the next cell has been reached and as A* algorithm is an informed pathfinding algorithm, and hence it works on cost estimation and requires an array of information to determine the next path $F(n)$ is the sum of $G(n)$ and $H(n)$;

$$F(n) = G(n) + H(n) \tag{1}$$

To find the first node from an initial position, the A* algorithm searches for all the possible nodes and finds the $F(n)$; for those nodes. The least value of $F(n)$ is selected and the path is formed. Now, when searching for the next path, it takes into consideration all the possible nodes, adds them up there, as well as the previous path's $G(n)$; to the $H(n)$; of the next node to find the next least $F(n)$; and then proceeds to the next node. The A* algorithm continues the same process till it reaches the desired goal. The most optimal path is the one that has the least value of $F(n)$; [7]. Figure 1 shows the nodes in A* algorithm.

The RRT, RRT*, and A* algorithms are coded on the MATLAB platform and are applied for the robot-based system in warehouse automation.

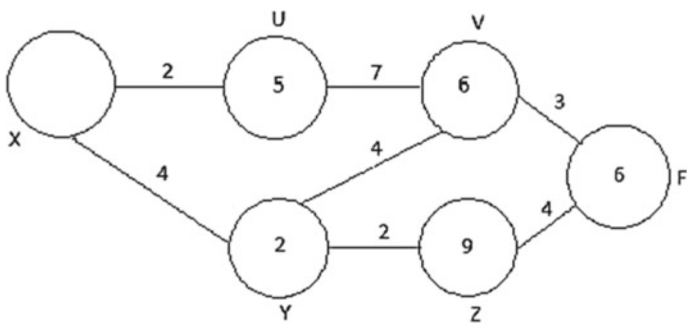


Fig. 1 A* algorithm [7]

3 Results

This section presents the results of RRT, RRT*, and A* algorithms.

RRT Algorithm

The simulation results of the RRT algorithm are shown in Fig. 2. The red dots indicate all the possible nodes that the algorithm has calculated to be in the permissible area. Taking forward, a green line indicates the connection of all the nodes that the RRT algorithm deems the most efficient and the nearest to the ones with lower *G* values and *H* costs.

RRT* Algorithm

The simulation results of the RRT* algorithm are shown in Fig. 3. This algorithm is a more sophisticated approach to the RRT algorithm as it does a more optimized search for all the *G* values, and hence a selected number of nodes/pathways are approached, which leads to a faster computational time to reach the end goal.

A* Algorithm

The simulation results of the A* algorithm are shown in Fig. 4. The red dots indicate the static obstacles in the path of the bot, and each of the blue and the green dots indicates the initial and the final position, respectively. The A* algorithm takes a

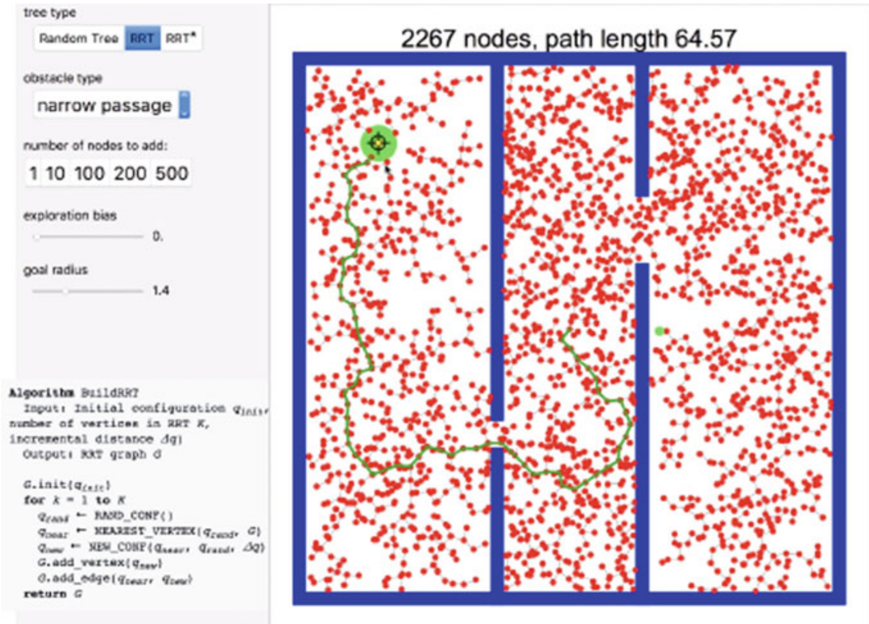


Fig. 2 RRT algorithm

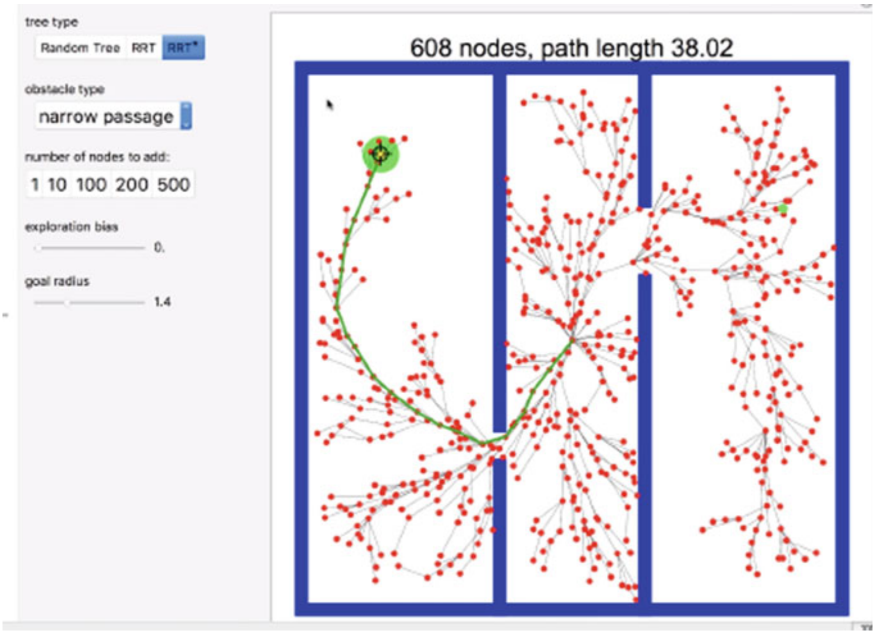
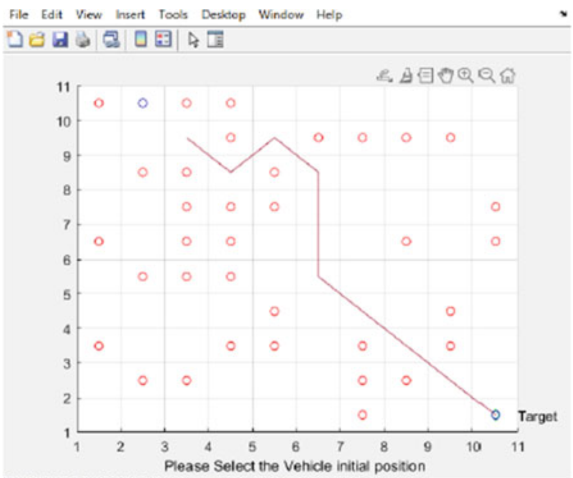


Fig. 3 RRT* algorithm

simple obstacle avoidance approach to plan the shortest path within the mapped area.

The application of the A* algorithm for warehouse automation is shown in Fig. 5. Here, a smart warehouse plan is demonstrated where a bot starts from its initial position of (0, 40) and has to reach its final defined position at a rack at (65, 45) in

Fig. 4 A* algorithm



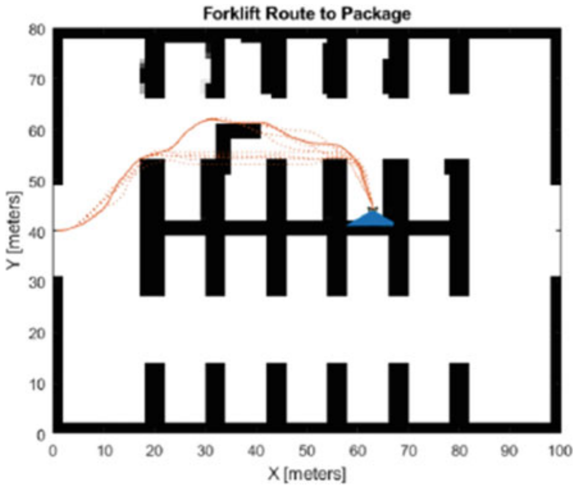


Fig. 5 Application of A* for warehouse automation

the warehouse, at (35, 60) is shown a dynamic obstacle because of which the bot detects it, and in order to avoid, it plans a secondary path (orange line) instead of the original simpler path (dotted orange lines). And this demonstrates the path planning and obstacle avoidance capabilities of bots using the A* algorithm.

4 Conclusions and Future Directions

In recent years, robotics has gained eminence in supply chain, distribution centre, and warehouse management circles and continues to play a significant role in warehouse automation. This paper presents the application of robot-based systems for warehouse automation. Three contemporary AI-based algorithms, viz. RRT, RRT*, and A* are applied for the path planning of a robot inside the warehouse. The robot can scan the warehouse map and avoid the obstacles inside the warehouse using these algorithms. With the successful application of a single robot system for warehouse automation, the current work could be expanded further for developing swarm robotics systems. Also, various nature-inspired evolutionary algorithms can be applied for the path planning of swarm robots inside the warehouse.

References

1. Adiyatov O, Varol HA (2017) A novel RRT*-based algorithm for motion planning in dynamic environments. In: 2017 IEEE international conference on mechatronics and automation (ICMA). IEEE, pp 1416–1421
2. Baker P, Halim Z (2007) An exploration of warehouse automation implementations: cost,

service and flexibility issues. Supply Chain Manag Int J
3. Belotserkovsky A, Lukashevich P, Doganli M, Rabcan J (2021) A concept of a multi-robotic system for warehouse automation. In: 2021 International conference on information and digital technologies (IDT). IEEE, pp 156–161
4. Cai K (2020) Warehouse automation by logistic robotic networks: a cyber-physical control approach. Front Inf Technol Electr Eng 21:693–704
5. Desaraju VR, How JP (2012) Decentralized path planning for multi-agent teams with complex constraints. Auton Robot 32(4):385–403
6. Grey J, Seneviratne O, Godage I (2021) Blockchain-based mechanism for robotic cooperation through incentives: prototype application in warehouse automation. In: 2021 IEEE international conference on blockchain (Blockchain). IEEE, pp 597–604
7. Guruji AK, Agarwal H, Parsediya DK (2016) Time-efficient A* algorithm for robot path planning. Procedia Technol 23:144–149
8. Hamberg R (2012) Automation in warehouse development, p 17Y32. Verriet J (ed) Springer
9. Islam F, Nasir J, Malik U, Ayaz Y, Hasan O (2012) RRT*-smart: rapid convergence implementation of RRT* towards optimal solution. In: 2012 IEEE international conference on mechatronics and automation. IEEE, pp 1651–1656
10. Javaid M, Haleem A, Singh RP, Suman R (2021) Substantial capabilities of robotics in enhancing industry 4.0 implementation. Cogn Robot 1:58–75
11. Karaman S, Walter MR, Perez A, Frazzoli E, Teller S (2011) Anytime motion planning using the RRT. In: 2011 IEEE international conference on robotics and automation. IEEE, pp 1478–1483
12. Koster R (2018) Automated and robotic warehouses. Logistics Transp 38(2):33–40
13. Laber J, Thamma R, Kirby ED (2020) The impact of warehouse automation in amazon's success. Int J Innov Sci Eng Technol 7:63–70
14. Poudel DB (2013) Coordinating hundreds of cooperative, autonomous robots in a warehouse. 27(1–13):26
15. Prakash R, Behera L, Mohan S, Jagannathan S (2020) Dual-loop optimal control of a robot manipulator and its application in warehouse automation. IEEE Trans Autom Sci Eng
16. Rey R, Corzetto M, Cobano JA, Merino L, Caballero F (2019) Human-robot co-working system for warehouse automation. In: 2019 24th IEEE international conference on emerging technologies and factory automation (ETFA). IEEE, pp 578–585

Application of Cohort Intelligence Algorithm for Numerical Integration

Mandar S. Sapre, Anand J. Kulkarni, Ishaan R. Kale, and Milind S. Pande

Abstract In this paper, the application of Cohort Intelligence algorithm for numerical integration of one-dimensional (1D) and two-dimensional (2D) functions are presented and discussed in details. These functions are easy to solve using traditional numerical methods. However, engineering applications from design domain associated with finite element analysis require numerical integration. For such application traditional methods may not suitable; hence, iterative methods are used. For the preliminary stage of the work, the CI is investigated for 1D and 2D integration functions and compared with Gaussian quadrature method, and the results are discussed in details.

Keywords Cohort intelligence · 1D and 2D integration · Finite element analysis · Gaussian quadrature

M. S. Sapre
Symbiosis Institute of Technology, Symbiosis International (Deemed University), Lavale, Pune, MH 412115, India
e-mail: mandar.sapre@sitpune.edu.in

A. J. Kulkarni · I. R. Kale (✉)
Institute of Artificial Intelligence, Dr. Vishwanath Karad MIT World Peace University, Pune, MH 411038, India
e-mail: ishaan.kale@mitwpu.edu.in

A. J. Kulkarni
e-mail: anand.j.kulkarni@mitwpu.edu.in

M. S. Pande
Dr. Vishwanath Karad MIT World Peace University, Pune, MH 411038, India
e-mail: milind.pande@mitwpu.edu.in

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_35

1 Introduction

Engineering applications always deals with system and processes that change. Engineer's duty is to draw various conclusions about the system from an analysis of its mathematical model. To 'Integrate' word suggests to bring close, i.e., to unite or to combine the parts as a whole. One can say integration is to compute the area enclosed by a curve, but an engineer can say integration is the process of calculating integral from a set of tabulated values of specific function. In many engineering applications, integration plays an important role. Quantities like heat and work, cannot be represented by a point on a thermodynamic plane. They are path function and their values depend on the path follow during the process. Hence, they are represented by area under curve. Work is represented by the area under the curve on pressure–volume or P–V plane. Several formulae like Newton Cotes Quadrature [4] and Gauss Legendre Quadrature [13] have been developed in the literature. This work is an attempt to use CI optimization technique for numerical integration.

Cohort Intelligence (CI) is socio-based nature inspired optimization technique Kulkarni et al. [10]. It models the social behavior of leaning candidates inherently having a common goal to achieve the best possible behavior in a cohort. Every candidate in a cohort follow, interact and compete with other candidates to improve their individual behavior which further helps to evolve the behavior of entire cohort. CI has been further modified and implemented to solve different domain problems. It was applied to solve several discrete and missed variables truss structure and mechanical design engineering constrained optimization problems [6, 7]; combinatorial problems from health care and logistic domain, and cross border supply chain [11], knapsack and traveling salesman problems [12]. CI was also, applies to solve the problems from manufacturing domain problems [9, 19]. CI has been applied for finite element mesh smoothing [17]. Variations of CI have been applied for mesh smoothing of hexagonal and pentagonal prisms [16]. A hybrid of GA and CI has been developed and applied for economic optimization of heat exchangers [5].

The paper is organized as follows: In Sect. 2, the steps for implementation of CI algorithm for 1D and 2D numerical integration is presented. Section 3 and 4 represents the solved 1D and 2D functions and its results, respectively. Section 5 investigates the CI for numerical integration functions the result analysis and discussion. The illustration of EFG method is presented in Sect. 6. Section 7 concludes the investigation.

2 CI for Numerical Integration

Since the stiffness matrices $forEFG$ and FEM require numerical integration, the application of CI to numerical integration is investigated. The most straightforward integration is the one-dimensional (1-D) integration with a single variable [21]. The proposed steps using CI are as follows:

Step 1: Define the function $f(x)$ along with the lower bound a and the upper bound b of the integration. Divide the interval $[a,\ b]$ into k sub-intervals $[a_0,\ a_1,\ \ldots,\ a_\mathrm{k}]$. The higher the value of k, better is the solution accuracy and lesser number of iterations are needed.

Step 2: Using the concept of mean value theorem,

$$\int_{ai}^{a_{i+1}} f(x)\mathrm{d}x = (a_{i+1} - a_i)\, f(\varepsilon_i) \tag{1}$$

where $\varepsilon_{i,}$ is the member of sub-interval $\left[a_i,\ a_{i+1}\right]$. The width of each sub-interval is chosen randomly using a random number between 0 and 1.

Step 3: For the number of candidates C ($C = c_1,\ c_2,\ \ldots,\ c_i,\ \ldots,\ c_\mathrm{n}$) such that $c_i = [\varepsilon_1,\ \varepsilon_2,\ \ldots,\ \varepsilon_\mathrm{k}]_i$. The integral for each candidate is computed using the following formula

$$\int_{ai}^{a_{i+1}} f(x)\mathrm{d}x = \sum_{i=0}^{k} f(\varepsilon_i)\mathrm{d}x_i \tag{2}$$

where $\mathrm{d}x_i = (a_{i+1} - a_i) * \mathrm{rand}(c,\ 1)$

Step 4: The integral is evaluated for each candidate. The mean integral is determined as

$$I_{\mathrm{mean}} = \frac{\sum_{i=1}^{n} I_i}{n} \tag{3}$$

Step 5: The objective function is the error given by $F_\mathrm{n}(i) = (I_{\mathrm{mean}} - I_i)^2$

Step 6: Since the aim is to minimize the error, calculate the probability associated with the behavior being followed by every candidate in the cohort using

$$p_c = \frac{1/F_c}{\sum_{c=1}^{N} 1/F_c} \tag{4}$$

Step 7: Using the roulette wheel approach, every candidate selects behavior from the C available choices.

Step 8: Every candidate shrinks/expands the sampling interval of every quality i based on whether the saturation condition is satisfied. The range is reduced by reduction factor r for every iteration.

Step 9: Every candidate forms new behaviors by sampling the qualities within the updated sampling intervals.

Step 10: Continue the process until the convergence criteria is achieved. Same concept is applied to double integrals with few modifications

The steps using CI to solve 2-D integration function are as follows:

Step 1: Define the function $z = f(x, y)$ along with the lower bound a and the upper bound b for both the variables.

Step 2: Intervals of both the variables are divided into k sub-intervals.

Step 3: The integral for each candidate is computed using the following formula

$$\int_a^b \int_p^q f(x, y)\mathrm{d}x\mathrm{d}y = \sum_{i=0}^{k} \sum_{j=0}^{k} f\left(\varepsilon_{xi}, \varepsilon_{yj}\right)\mathrm{d}x_i\mathrm{d}y_j \tag{5}$$

Step 4: The other steps remains the same as in 1-D integration.

The above steps were simulated using MATLAB (R2018a), and the simulations run on the Windows platform using an Intel Core(™) i5-4200U CPU @ 1.60 GHz processor speed and 4 GB of random access memory (RAM). The maximum numbers of learning attempts were 200. These values were chosen based on several experiments. The proposed algorithm was run for 30 times to solve the problems.

3 CI for Integration of One Variable

This problem explains the use of CI for integration of function in one variable. It is defined as follows:

$$f(x) = x^2 \text{and } I = \int_0^1 x^2\mathrm{d}x \tag{6}$$

The results obtained from CI are compared with Gaussian quadrature for two more functions as shown in Table 1. A sample convergence plots for all the considered functions are presented in Fig. 1.

Table 1 The comparison of CI and Gauss quadrature solutions solving 1-D integration functions

Sr No.	Function	Limits	Function value		Computational time (s)	
			Gaussian quadrature	CI	Gaussian quadrature	CI
1	x^2	0, 1	0.3333	0.3333	0.009	5.12
2	$4/\left(1+x^2\right)$	5, 10	0.390908	0.390905	0.006	37.34
3	$\cos(x)$	0, 50	−0.262375	−0.283122	0.002	5.16

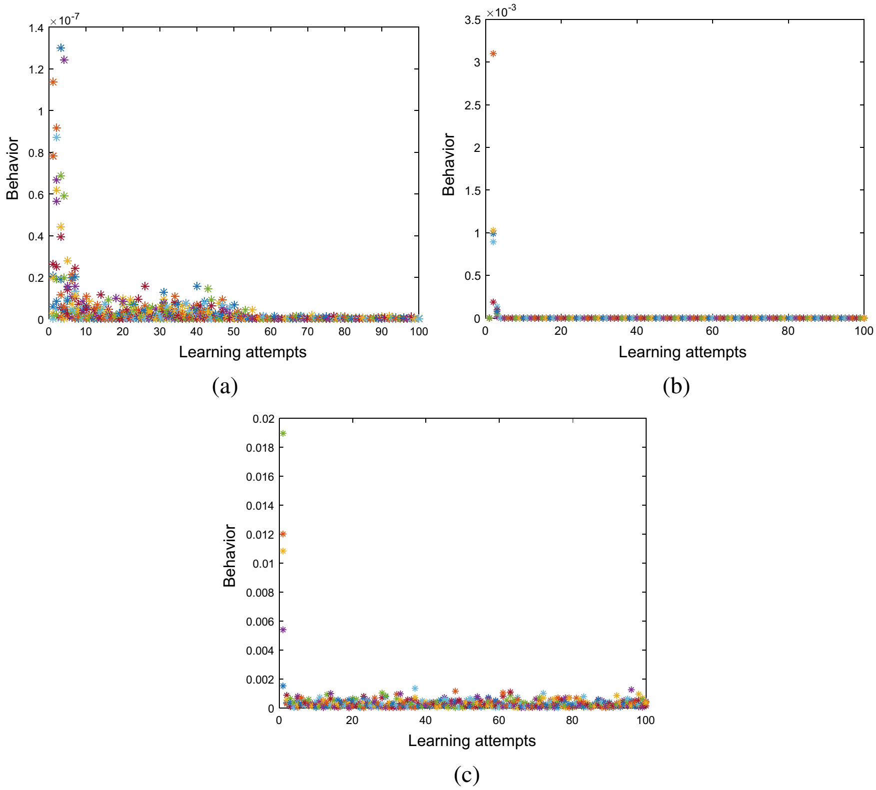


Fig. 1 CI convergence for 1D integration functions **a** x^2; **b** $4/\left(1+x^2\right)$; **c** $\cos(x)$

4 CI for Integration of Two Variables with Constant Limits

The proposed steps for double integration using CI are presented in Sect. 2. The integral is

$$I = \int_{0}^{10}\int_{0}^{10} e^{-x^2-y^2}\,\mathrm{d}x\,\mathrm{d}y \tag{7}$$

For the comparison of results three function are considered. The results are presented in Table 2, and the convergence plots are presented in Fig. 2.

Table 2 The Comparison of CI and Gauss quadrature solutions solving 2D integration functions

Sr No.	Function	Limits	Function value		Computational time (s)	
			Gaussian quadrature	CI	Gaussian quadrature	CI
1	$\int_0^{10}\int_0^{10} e^{-x^2-y^2}\,\mathrm{d}x\mathrm{d}y$	0, 10 and 0, 10	0.7853981634	0.7860	0.0787	36.63
2	$\int_0^{5}\int_0^{5}(x^2+y^2)\,\mathrm{d}x\mathrm{d}y$	0, 5 and 0, 5	416.6666	416.6697	0.0704	13.00
3	$\int_0^{\pi/4}\int_0^{\pi/3}\sin(x)\cos(y)\,\mathrm{d}x\mathrm{d}y$	0, $\pi/4$ and 0, $\pi/3$	0.2536	0.2526	0.0777	17.23

5 Investigation of CI for Integration

CI algorithm is successfully applied to solve 1D and 2D integration functions. The solutions obtained from CI algorithm are compared with exact optimization Gaussian quadrature technique (refer Tables 1 and 2). CI has obtained similar results as compared to Gaussian quadrature. The convergence graphs for all the solved functions are present in Fig. 1 and Table 2. The time required for the integration using CI was high compared to the time taken by the direct method. The method is computationally expensive and slow because more subintervals and iterations are needed for an accurate answer. It balances the accuracy and the computational cost/speed. However, CI could be more suitable when complex integration functions are associated with the real-world applications where the traditional numerical methods may not be applicable.

6 Element Free Galerkin (EFG) Method

EFG is a numerically stabled meshfree method, which makes it easy for comprehension and implementation [14]. EFG is used in many applications like crack analysis and fracture mechanics [2], large-deformation processes [18], metal forming [3], and heat transfer [20]. In this method, the moving least square (MLS) technique is used to construct the shape functions as follows:

$$(x) = p^T(x)A^{-1}(x)B(x) \tag{8}$$

where

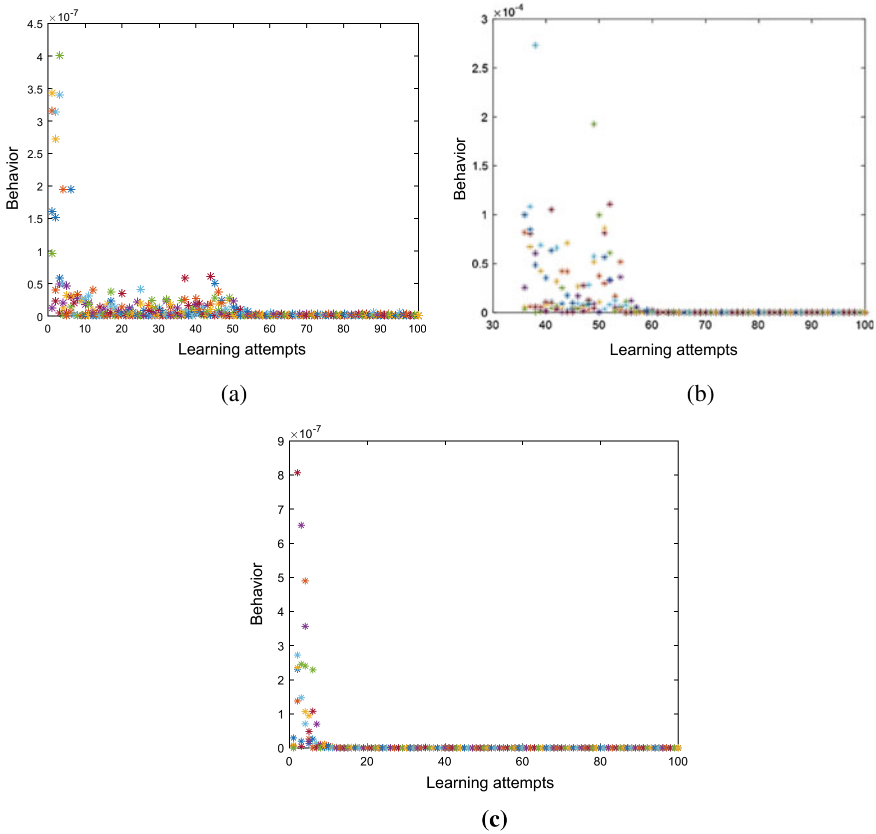


Fig. 2 CI convergence for 2D integration **a** $\int_0^{10}\int_0^{10} e^{-x^2-y^2}\mathrm{d}x\mathrm{d}y$; **b** $\int_0^{5}\int_0^{5}\left(x^2+y^2\right)\mathrm{d}x\mathrm{d}y$; **c** $\int_0^{\pi/4}\int_0^{\pi/3} \sin(x)\cos(y)\mathrm{d}x\mathrm{d}y$

$p^T(x) = [1\ x\ y]$ is the basis function

$$B(x_i) = w(x - x_i)\ p(x_i) \text{ and}$$

$$A(x) = \sum_{I=1}^{N} w(x - x_i)p(x_i)p^T(x_i)$$

Here, $w(x - x_i)$ is the weight function, which is usually a cubic spline function or a Gaussian function, and N denotes the number of nodes whose support includes the point x. The shape functions derived from the MLS approximation method do not satisfy the Kronecker delta condition. Different procedures such as Lagrange multipliers are needed for imposition the essential boundary conditions [15].

The final discrete equation is given by Eqs. 9–11

$$\begin{vmatrix} K & G \\ T & 0 \end{vmatrix} \begin{pmatrix} u \\ \lambda \end{pmatrix} = \begin{pmatrix} f \\ q \end{pmatrix} \tag{9}$$

$$K_{ij} = \iiint B_i^T \mathrm{DB}_j \mathrm{d}v \tag{10}$$

$$G_{i\mathrm{k}} = \iint \Phi_i N_\mathrm{k} \mathrm{d}\Gamma \tag{11}$$

where

K is the global stiffness matrix,
G is the stiffness matrix of essential boundary conditions,
B is the strain matrix for node i,
N is the Lagrange interpolation matrix
D is the stress–strain relationship matrix, and
E and ν are Young's modulus and Poisson's ratio, respectively.

The stiffness matrices K and G require use of integration. The MLS shape functions are not linear and hence the integrand is a complex function. The direct integration is impossible; hence, numerical integration is required. Gaussian quadrature and GA are used in the literature [1]. Numerical integration is an inherent step for analyzes using meshfree methods like EFG [14].

7 Conclusion

It is demonstrated that CI can be applied to 1D and 2D integration functions. The proposed procedure can be applied for the integration of stiffness matrices in EFG. The relative error function was used as the objective function. The present work explores only simple functions, and this procedure could be modified to integrate the matrix functions, and hence, it could be used for EFG. The function in EFG are matrix functions. In the future, variations of CI and ARGA could be used for numerical integration.

References

1. BaniHani SM (2007) On the development and some application of a genetic algorithm based lookup table approach for efficient numerical integration in the method of finite spheres. Rensselaer Polytechnic Institute

2. Belytschko T, Lu YY, Gu L (1995) Crack propagation by element-free Galerkin methods. Eng Fract Mech 51(2):295–315
3. Breitkopf P, Naceur H, Rassineux A, Villon P (2005) Moving least squares response surface approximation: formulation and metal forming applications. Comput Struct 83(17–18):1411–1428
4. Harrold OG (1937) On the expansion of the remainder in the open-type Newton-cotes quadrature formula. Am J Math 59(2):275–289
5. Iyer VH, Mahesh S, Malpani R, Sapre M, Kulkarni AJ (2019) Adaptive range genetic algorithm: a hybrid optimization approach and its application in the design and economic optimization of a shell-and-tube heat exchanger. Eng Appl Artif Intell 85:444–461
6. Kale IR, Kulkarni AJ (2018) Cohort intelligence algorithm for discrete and mixed variable engineering problems. Int J Parallel Emergent Distrib Syst 33(6):627–662
7. Kale IR, Kulkarni AJ (2021a) Cohort intelligence with self-adaptive penalty function approach hybridized with colliding bodies optimization algorithm for discrete and mixed variable constrained problems. Complex Intell Syst 7(3):1565–1596
8. Kale IR, Kulkarni AJ (2021b) Constraint Handling in Cohort Intelligence Algorithm. CRC Press
9. Kale IR, Pachpande MA, Naikwadi SP, Narkhede MN (2022) Optimization of advanced manufacturing processes using socio inspired cohort intelligence algorithm. Int J Simul Multi Design Optim 13:6
10. Kulkarni AJ, Durugkar IP, Kumar M (2013) Cohort intelligence: a self supervised learning behavior. In: 2013 IEEE international conference on systems, man, and cybernetics, pp 1396–1400
11. Kulkarni AJ, Baki MF, Chaouch BA (2016) Application of the cohort-intelligence optimization method to three selected combinatorial optimization problems. Eur J Oper Res 250(2):427–447
12. Kulkarni AJ, Krishnaswamy G, Abraham A (2017) Cohort intelligence: a socio inspired optimization method. Springer International Publishing, Switzerland
13. Lowan AN, Davids N, Levenson A (1942) Table of the zeros of the Legendre polynomials of order 1–16 and the weight coefficients for Gauss' mechanical quadrature formula, vol 18
14. Liu GR (2009) Meshfree methods: moving beyond the finite element method. CRC Press
15. Mukherjee YX, Mukherjee S (1997) On boundary conditions in the element-free Galerkin method. Comput Mech 19(4):264–270
16. Sapre MS, Kulkarni AJ, Chettiar L, Deshpande I, Piprikar B (2021) Mesh smoothing of complex geometry using variations of cohort intelligence algorithm. Evol Intel 14(2):227–242
17. Sapre MS, Kulkarni AJ, Shinde SS (2019) Finite element mesh smoothing using cohort intelligence. In: Proceedings of the 2nd international conference on data engineering and communication technology, pp 469–480
18. Scitovski R, Ungar Š, Jukić D (1998) Approximating surfaces by moving total least squares method. Appl Math Comput 93(2–3):219–232
19. Shastri AS, Nargundkar A, Kulkarni AJ, Sharma KK (2020) Multi-cohort intelligence algorithm for solving advanced manufacturing process problems. Neural Comput Appl 32(18):15055–15075
20. Singh IV, Sandeep K, Prakash R (2003) Heat transfer analysis of two-dimensional fins using meshless element free Galerkin method. Numer Heat Transfer Part A Appl 44(1):73–84
21. Morozenko V, Pleshkova I (2013) Numerical integration by genetic algorithms. Int J Inf Theor Appl 20(3):252–262

Intelligent Systems for Social Welfare II

AI Enabled Energy Consumption Predictor for Smart Buildings

Smrutishikta Das, Tapas Kumar Choudhury, Sanjit Kumar Dash, and Jibitesh Mishra

Abstract Artificial intelligence has been taking the prior position in the technology domain. Unlike other technologies, smart energy management is one of the most likely topics for some specialists in parts of better comprehension of the unpredictable example of energy creation, dissemination and utility line. For the quicker and efficient prior energy demand prediction an AI enabled energy consumption predictor has been developed. A largely varying energy consumption demand within a city is the case where statistical analytics could improve the efficiency of the energy demand prediction. In this paper, data analytics methods have emphasized to aggregate false outliers within widely varying training data. The main focus of appropriate training data selection though statistical analysis could be able to increase the efficiency of the prediction score from 0.90 to 0.96. Various regression models have been implemented over a publicly available data set. Finally, a smart energy consumption predictor could be able to increase overall prediction accuracy up to 6.66%.

Keywords Data science · Energy consumption · Artificial intelligence · Prediction model · Statistical machine learning · Data-driven analysis

1 Introduction

According to the U.S energy estimation of 2019, there would be an increase of 7.2% in energy consumption in India by 2050. As per today's growing need, there would be an obvious straight increase in energy demand in addition to the development of the country in the near future [1, 2]. For the present need, the building energy demand prediction became one of the most desirable research areas for the smart power system and sustainable energy [3–5]. The exact prediction of energy demand per floor area with specific in-house appliances has taken a wider space in the planning of the commercial as well as the residential area planning [6–8]. Not at all like the customary energy had to anticipate technique [9], the AI strategy

S. Das · T. K. Choudhury · S. K. Dash (✉) · J. Mishra
Odisha University of Technology and Research, Bhubaneswar, Odisha, India
e-mail: sanjitkumar303@gmail.com

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_36

improved as a forecast alternative for the adaptable energy utilization design. The artificial intelligence-based methodology has had the option to foresee the qualities which were exceptionally nearer to the real qualities [10]. In the energy creation, supply, and usage chain, there may be a breakdown someplace or may be influenced by manual or specialized blunders. The AI-based indicators have been intended to catch the specific conduct and the example at explicit conditions if there should be an occurrence of building energy utilization expectation [11, 12].

The main motivation of this research work is to predict the commercial building energy consumption demand through various underlying energy demand patterns. According to the utility of the buildings are of in different categories in addition to the floor area, appliances, and seasonal utility. To consider the effect of the seasonal utility criteria and the consumption to floor area ratio a predictor has been implemented to capture the exact energy demand prediction. In this paper, we have implemented a statistical analytics-based predictor along with advanced machine learning prediction algorithms to attain up to 92% of accuracy. From the commercial building energy consumption historical data, we have repeatedly labeled the energy demand ranges in various circumstances to foresee the energy request per floor territory for the particular structure type notwithstanding the occasional energy utilization variety in the structure. Before the planning of the city or and small commercial area through the predictor, we can overview a near-exact energy demand for each building category. According to the utility and floor area, the energy demand is predicted along with the seasonal influence of the cooling and heating devices as major utility devices.

2 Literature Review

Energy demand prediction is mostly related to planning. For commercial or residential planning and energy distribution purposes, the future demand for energy is estimated beforehand. The concept of energy demand has started from the concept of city planning [13]. The energy distribution and the load measurements are the main of this energy demand prediction. On the other hand, the commercial building energy predictors are implemented for the smart utility of the energy for better economic growth [6]. The variation in energy demand, supply, and production is shrinking gradually due to the energy demand predictors. The energy demand predictors could be implemented for the annual energy demand prediction or for the household energy monitoring with the smart meter [9, 14, 15]. For the commercial energy demand prediction, the annual energy consumption data were utilized for the energy demand prediction. Data analysis is the most important part of the supervised machine learning models [16–18]. The whole model performance and the development completely depend on the data only. The feature importance and the behavior of the data also play an important role to decide the desired algorithms [6, 19, 20] for the prediction purpose. The energy demand predictors are influenced by many external features or due to some dependent and independent feature set [1, 6]. Among the influencing features, only the scalable features could be included in the energy demand prediction model. According to the survey, the AI-based predictors are the most recent prediction models where the model output could be able to produce the most accurate predictions [1]. On the other hand, the handling of the data set and the feature analysis could be easier with data science. Data science is not a new thing for the prediction modeling mainly utilized in time series data types. We can go for data science to deal with the outliers and to discover all possible hidden patterns [21, 22]. The pattern or the trend analysis could hike the model performance in most cases. The AI-based predictors are more accurate than any other prediction models with all possible conditions in the pattern or trend evaluation [19, 23]. Besides this, the application of the most appropriate algorithms could also stand alone for better predictions [23]. According to the pattern or trends and the features of the training data, we can pick the most suitable algorithms for the model. The model parameters and the features and the algorithm selection collectively could implement the best performing energy demand predictor.

3 Proposed Model

An artificial intelligence-based machine learning predictor has been introduced to predict energy demand in addition to the independent features. The model has been designed with two subsequent parts. Part one dealt with statistical data analysis and data science and the second part is an application of machine learning algorithms for predictions as shown in Fig. 1. Artificial intelligence belongs to the conditional approaches. Some of the AI-based standardization methods have been applied for the efficient evaluation of the model. According to the data analytics requirements, machine learning algorithms were implemented. Out of the algorithms, the best fit models were selected for further validation and finally for the voting process. For this research work, we have considered a labeled data set of commercial energy consumption data. All the labels are unique and well defined to implement a supervised machine learning model. The problem belongs to a regression-based prediction, where we need to predict the values according to the forecasted values.

According to the problem statements, the energy consumption predictor needs to be implemented with respect to the independent features related to the commercial buildings. From the historical data analysis, the target values could be forecasted through multivariate linear regression. Linear regression is the method to find the best fit line throughout the available data points known as regression line as given in Eq. (1).

$$y_i = \theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + \cdots + \theta_n x_{in} + \in \tag{1}$$

For a train-test data set (x, y), the hypothesis could be defined as to estimate the value of y. Here, the train set has been taken as x and the test set has been taken as y. Here, the θ value stands for the coefficient of the x. To increase the prediction

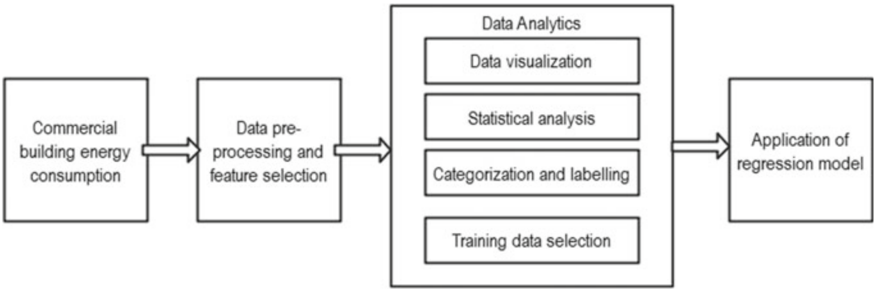


Fig. 1 System model for energy consumption demand prediction

accuracy, we need to minimize the θ values, which is known as the cost function. Mathematically, the cost function of the multivariate linear regression can be written using Eq. (2)

$$J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x)^i - (y^i)\right)^2 \tag{2}$$

The application model uses three regression algorithms namely random forest, gradient boosting, and bagging regressor. The reason behind the selection of these three models supports mainly low variance. Most of the data points were utilized through the fandom forest algorithm where the gradient boosting worked upon the weak learner's models. The bagging regressor algorithm with low variance provides the highest accuracy among these three algorithms.

4 Data Set Analysis and Processing

The data has been taken from the Commercial building energy consumption survey 2012 by the US Energy information administration with 6700 commercial building data (source: https://www.eia.gov/consumption/commercial/data/2012). Energy consumption data for commercial buildings is quite discrete in nature. There exist huge differences between the energy demands for different cases which may not be able to capture by the normal predictor due to huge variations. In the earlier stage of data pre-processing, the range could be defined for the particular building categories and the most appropriate data could be selected. In high dimensional spaces, the subcategories with the data set could be noticed which could be categorized and labeled separately for more accurate predictions. In the commercial building energy data set, the energy demand (MFBTU) with respect to floor area (SQFT) is considered for the data analysis as shown in Fig. 2.

In high dimensional space, we found three major subgroups for the energy consumption demand according to the floor area. The subcategories are divided

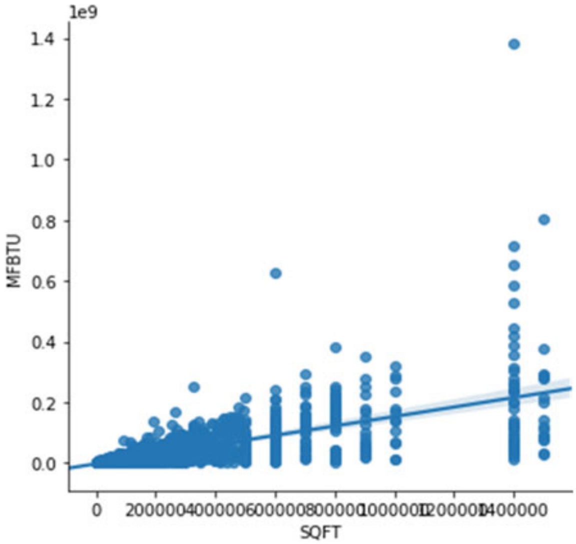


Fig. 2 Data visualization for energy demand according to floor area

and visualized by means of a scatter plot for further verification. Two independent variables are taken as the base and the energy demand to floor area is evaluated accordingly. And finally, we could able to find the true outliers from the major subgroups. With the application of statistical machine learning the parameters and the data set is subdivided, joined, or normalized in terms of data availability and data sufficiency. For any kind of machine learning algorithm, there is a need for sufficient data for each subset for utilization. Otherwise, the model will overfit with less data or could not find maximum data points for further application or there would be no impact of any optimization or evaluation on the model.

The availability of sufficient data for the specific category is the main part of training data selection. Exceptional cases and wide differences in energy demand value are quite common in energy demand predictors. As the building category varies, the energy demand varies in different patterns which seem as if outliers to the predictor. Six features are taken for the prediction model. The square foot area of the building (SQFT), number of floors (NFLOOR), and seasonality were considered as the independent variables where the annual energy expenditure (MFBTU) is calculated as the target outcome. The energy demand also varies according to the seasons, for which two more features are considered for the seasonality. A base temperature of 28 °C has been taken. The energy consumption below the base temperature (CDD65) and above the base temperature (HDD65) is considered as two more features for the seasonality variation.

After the first phase of data cleaning and analysis, three major categories are discovered besides different building categories. A major cluster of buildings shows energy demand with lower to moderate energy demand in Fig. 3a, another cluster showing equally likely energy demand pattern as shown in Fig. 3b, and the third category showing a pattern of different energy demand for the same area in Fig. 3c.

Energy demand values do not follow any specific rules for each described building category as there exist some exceptions and different amounts of energy utilization under the same category. While the analysis took over individual building categories the analysis report suggested different boundary values as shown in Fig. 4a, b, and c. In the second step, the underlying pattern of different parameters are considered to analyze the spreading of data point of energy demand up on corresponding floor area. The number of floors, with respect to categories, is the two major criteria. Among all these categories, the energy demand is mainly depending upon the floor area with respect to building categories. After range estimation, it became easier to distinguish outliers for each case. This concludes the true outlier detection and proper training data selection. The appropriate selected training data is fed to a normal predictor using random forest, gradient boosting, and bagging regressor algorithms.

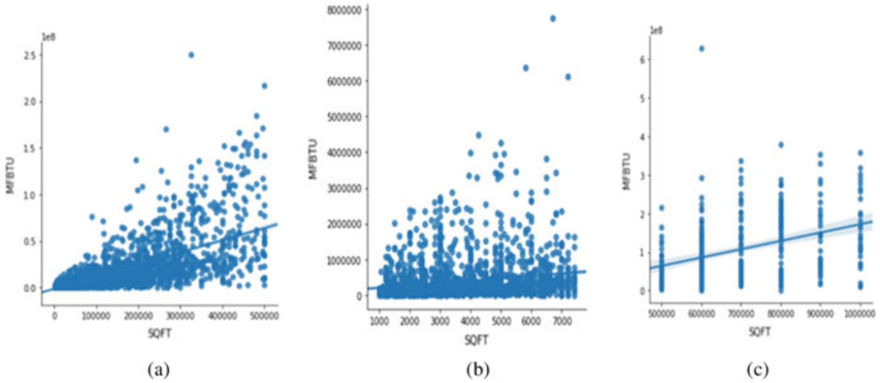


Fig. 3 Subcategories of energy demand within the data set

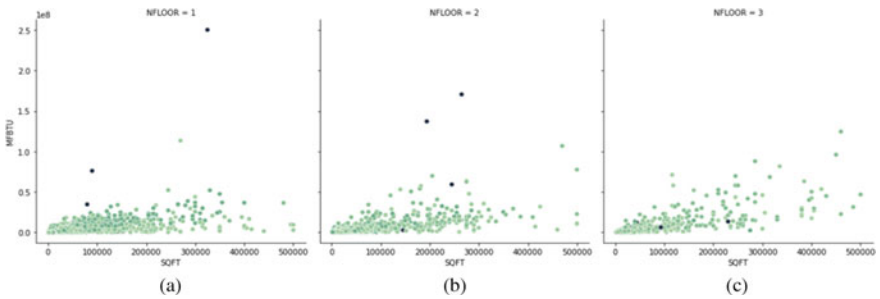


Fig. 4 Energy demand distribution in each category

5 Results and Discussions

The AI-based predictor is implemented to predict the energy consumption demand per floor area with respect to the building category. We have applied three regression models random forest, gradient boosting, and bagging regressor. Initially, a normal predictor was implemented to verify the performance metrics for each algorithm and the efficiency enhanced by the application of data science. The simple predictor contained the train-test splits of 75% training and 25% of testing data and implemented the above said three algorithms for the prediction purpose. The prediction accuracy of random forest algorithms is 0.90, the bagging regressor is 0.88, and the gradient boosting algorithm is 0.92. Among these three regression algorithm, the bagging regressor algorithm performed well for the model. The accuracy of the normal predictor and AI-based predictor are compared with the predicted values for each algorithm which is shown in Table 1.

In the normal prediction model, there is a huge variation in the test data and the predicted values. In most cases, the predicted values are much higher or lower as compared to the original values. These values are optimized through the AI-based predictor and the efficiency of prediction increased on an average of 6.66%. The model performance could be visualized in Fig. 5 showing the variations among the predicted and the actual values for each algorithm.

The final values of the voting regressor are taken as the final model outcome. The proposed model is able to provide up to 93% of accuracy with respect to the square footage area, seasonality, and the building category. The predicted regression line and the actual data are shown in terms of a two-dimensional plot in Fig. 6. The regressor line in boosting regressor fits into the data points in normal predictor is compared with the AI-based predictor. The normal predictor with other optimization algorithms and outlier detection algorithms may or may not conclude with the true outliers. In the case of the statistical analysis-based training data selection method, the predictors got maximum conditions to minimize the variations during the prediction and hence able to provide better efficiency.

Table 1 Comparison between normal predictor and AI-based predictor

Normal predictor				AI predictor			
Algorithm	R^2 score	Test variance	Root mean squared error	R^2 score	Test variance	Root mean squared error	Efficiency enhancement (%)
Random forest	0.90	0.47	48,908,576	0.96	0.73	47,068,168	6.66
Gradient boosting	0.88	0.54	45,183,497	0.84	0.75	45,492,115	3.16
Bagging regressor	0.92	0.52	46,398,645	0.95	0.73	50,608,405	− 4.5

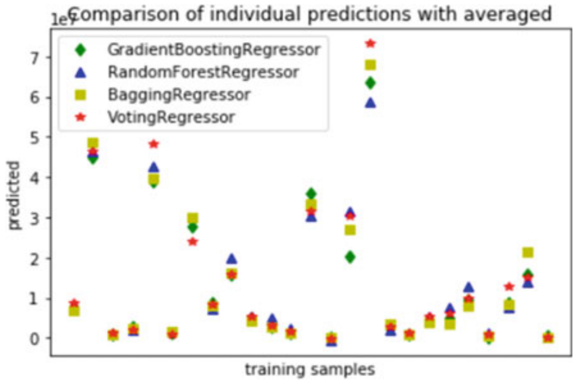


Fig. 5 Variation of predicted values

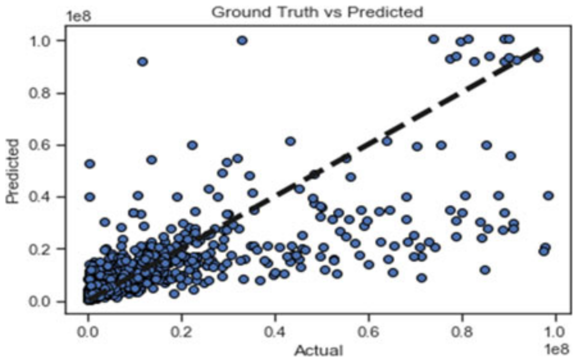


Fig. 6 Predicted regression line among actual data points

6 Conclusion

The prediction model has been developed to predict commercial building energy demand with respect to the floor area, seasonal variation, and the building category. Data science and statistical machine learning have been applied to discover all possible underlying data patterns for each building category and labeled simultaneously. The low variance random forest model could be able to provide around 92% accuracy as compared to other error minimizing algorithms. In this particular data set, we have considered 11 different building categories for which the energy utilization ranges are analyzed to deal with the larger variations within a data set. The data analysis and range estimation method increased the prediction efficiency up to 6.72% as compared to a normal prediction model. Other algorithms could be applied to cover the maximum available subgroups within the data set. Other error minimizing methods could be applied to target the outliers and the errors in the training data. In this research work, we have applied both data science and artificial intelligence for prediction purposes. More statistical analysis would be able to implement a better-performing AI-based predictor.

References

1. Wang B (2017) Building energy performance estimation approach: facade visual information-driven benchmark performance model. Doctoral dissertation, University of Southern California
2. Howard B, Parshall L, Thompson J, Hammer S, Dickinson J, Modi V (2012) Spatial distribution of urban building energy consumption by end use. Energy Build 45:141–151
3. Nielsen HA, Madsen H (2006) Modelling the heat consumption in district heating systems using a grey-box approach. Energy Build 38(1):63–71
4. Boulaire F, Higgins A, Foliente G, McNamara C (2014) Statistical modelling of district-level residential electricity use in NSW, Australia. Sustain Sci 9(1):77–88
5. Kontokosta CE (2012) Predicting building energy efficiency using New York City benchmarking data. In: Proceedings of the 2012 ACEEE summer study on energy efficiency in buildings. American Council for an Energy-Efficient Economy, Washington, DC
6. Zhao HX, Magoulès F (2012) A review on the prediction of building energy consumption. Renew Sustain Energy Rev 16(6):3586–3592
7. Zhao J, Lam KP, Karaguzel OT, Ahmadi S (2012) Design–Build–Operate energy information modeling (DBO-EIM) for green buildings: case study of a net zero energy building
8. Li Z, Han Y, Xu P (2014) Methods for benchmarking building energy consumption against its past or intended performance: an overview. Appl Energy 124:325–334
9. Wei L, Tian W, Silva EA, Choudhary R, Meng Q, Yang S (2015) Comparative study on machine learning for urban building energy analysis. Procedia Eng 121:285–292
10. Liu D, Chen Q (2013) Prediction of building lighting energy consumption based on support vector regression. In: 2013 9th Asian control conference (ASCC). IEEE, pp 1–5
11. Korolija I, Marjanovic-Halburd L, Zhang Y, Hanby VI (2013) UK office buildings archetypal model as methodological approach in development of regression models for predicting building energy consumption from heating and cooling demands. Energy Build 60:152–162
12. Mathew PA, Dunn LN, Sohn MD, Mercado A, Custudio C, Walter T (2015) Big-data for building energy performance: lessons from assembling a very large national database of building energy use. Appl Energy 140:85–93
13. Hapase CR (2019) Melanoma detection in dermoscopy images using a cloud based machine learning application. Doctoral dissertation, Texas A&M University-Kingsville
14. Zeifman M (2014) Smart meter data analytics: prediction of enrollment in residential energy efficiency programs. In: 2014 IEEE International conference on systems, man, and cybernetics (SMC). IEEE, pp 413–416
15. Kuhn M, Johnson K (2013) An introduction to feature selection. In: Applied predictive modeling. Springer, New York, pp 487–519
16. Mehar AM, Gill AQ, Matawie K (2018) Analytical model for residential predicting energy consumption. In: 2018 IEEE 20th Conference on business informatics (CBI), vol 2. IEEE, pp 82–88
17. Fan C, Xiao F, Wang S (2014) Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques. Appl Energy 127:1–10
18. Kuhn M, Johnson K (2013) Applied predictive modeling, vol 26. Springer, New York, pp 13
19. Tso GK, Yau KK (2007) Predicting electricity energy consumption: a comparison of regression analysis, decision tree and neural networks. Energy 32(9):1761–1768
20. Robinson C, Dilkina B, Hubbs J, Zhang W, Guhathakurta S, Brown MA, Pendyala RM (2017) Machine learning approaches for estimating commercial building energy consumption. Appl Energy 208:889–904
21. Das S, Choudhury TK, Mohapatra SK (2020) Data analytics to increase efficiency of the AI based energy consumption predictor. In: 2020 International conference on computational intelligence for smart power system and sustainable energy (CISPSSE). IEEE, pp. 1–4

22. Brown RE, Walter T, Dunn LN, Custodio CY, Mathew PA, Berkeley L (2014) Getting real with energy data: using the buildings performance database to support data-driven analyses and decision-making. In: Proceedings of the ACEEE summer study on energy efficiency in buildings, pp 11–49
23. Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In: IJCAI, vol 14, no. 2, pp 1137–1145

When Sustainable Development Embraces Blockchain: A Systematic Literature Review

Parikshit Joshi, Anshu Singh, Shailendra Kumar, Garima Joshi, Ankit Aggarwal, and Sushil Kumar Gupta

Abstract The non-renewable resources are limited on this earth and will get exhausted if not utilized carefully. Ensuring the availability of these resources for future generations is the key philosophy of sustainable development (SD). To protect and preserve the environment, United Nations has proposed 17 sustainability development goals, which can be attained by coupling nature, human and technology. Among several technological disruptions, blockchain technology (BCT) has the huge potential to verify SD. The current study aims at analysis and synthesis of available studies on implication of BCT for SD, using systematic literature review methodology. Out of 17 SDGs only two goals, *zero hunger (*goal no. 2*)* and *clean water and sanitation* (goal no. 6), were considered for this study. The literature search criteria comprised of automatic and manual search, resulted into 484 studies. After examining inclusion and exclusion criteria, 52 usable studies were obtained. Descriptive analysis of 52 studies was done on the basis of publication year, source of publication, research type and sustainability dimension reported. Synthesis and analysis of these 52 studies addressed the following: implication of BCT for SD, identification

P. Joshi (✉)
Department of Management Studies, Indian Institute of Information Technology Allahabad, Prayagraj, Uttar Pradesh, India
e-mail: f14parikshitj@iima.ac.in

A. Singh
Amity Business School, Amity University Uttar Pradesh, Noida, Uttar Pradesh, India

S. Kumar
Department of Management Studies, Indian Institute of Information Technology Allahabad, Prayagraj, Uttar Pradesh, India

G. Joshi
Management Department, DNS College of Engineering and Technology, Amorha, Uttar Pradesh, India

A. Aggarwal
School of Management, Graphic Era Hill University, Haldwani, Uttarakhand, India

S. K. Gupta
School of Management (UG), Dr. Vishwanath Karad MIT World Peace University, Pune, Maharashtra, India

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_37

of research themes and gaps in available studies and future research direction. It was observed that none of the study has been done on the theme and methodology used in current study.

Keywords Sustainable development · Blockchain · Business practice · Environment · Economic · Society · Triple bottom line

1 Introduction

The process of utilizing available resources in a way that they are available for current and future needs is known as sustainable development (SD) [33]. Responding to the global call of protecting the environment, economy and society (triple bottom line dimensions of SD [11]), the United Nations (UN) proposes a 17 goal agenda known as Sustainable Development Goals (SDGs), to be achieved by 2030 [56].

SDGs can be attained by coupling nature, human and technology [46]. Among several technologies available, blockchain technology (BCT) is the one with features like information transparency, information immutability, security and data privacy [26] and hence has the potential to assist in attainting SDGs [30].

Available literature reveals a growing interest of researchers and practitioners on the theme "application of BCT for SD" in the last few years [38]. Although the interlinkage of BCT and SD has already been established, still the operationalization of SDGs lacks a shared understanding among the stakeholders [46]. Therefore, summarizing the available studies on BCT implication for SD, through a systematic literature review (SLR) approach and identifying future research directions is essential at this juncture.

The current study aims at mapping the available studies addressing the BCT usage for SD through the SLR approach. To meet the research objectives, this study is organized into the following sections including the current section: Sect. 2 outlines the brief overview of SDG no. 2 and 6; Sect. 3 explains the review methodology adopted for the current SLR; Sect. 4 highlights the descriptive analysis of the studies (research papers, articles, books chapters and conference papers) considered for the current SLR; Sect. 5 answers the research questions, and Sect. 6 is the conclusion of this study.

2 Background

Today, around 815 million human beings globally are undernourished and 2 billion are suffering from micronutrient deficiencies [5]. Food adulteration and contamination cost about 420,000 deaths across the globe every year, which include 125,000 children under 5 years age [31]. Sick and ill people due to lack of adequate nutrients are an economic burden to society and nation [36]. Underdeveloped and developing economies are paying heavy price of around \$110 in terms of lost productivity and medical expenses due to unsafe food and badly managed agriculture supply chain [31] annually.

Goal 2 of SDGs, emphasize on "end of hunger, achieving food security and improved nutrition and promote sustainable agriculture" [56]. Food security can be ensured with transparency in transactions, low operational cost and traceability [2, 13] and BCT has all these features.

Water is the foundation of life for centuries [22]. Human civilizations have evolved near natural water resources only,however, today clean and drinking water is not available to masses and several cities of the world are running out of water. Currently, almost 4 billion people across the globe are living in water-scarce regions, out of which 2 billion have limited access to drinking water [25] and 2.7 billion lack access to sanitation [20].

The aim of SDG no. 6 is "to ensure availability and sustainable management of water and sanitation for all" [56]. Despite of scarcity of clean and drinking water worldwide, majority of the people are still using traditional ways of water management. The water crisis demands us to leave the traditional ways and switch to non-conventional approaches, such as technology (BCT)-driven water management systems [51].

The above mention facts and data act as a strong premise for choosing technology (blockchain) enabled solution for addressing food, water and sanitation issues. Hence, in the current study out of 17 SDGs, the focus is only on goal no. 2 (*zero hunger)* and 6 (*clean water and sanitation).*

3 Review Method

3.1 Research Questions

SLR is a scientific approach for analysis, extraction and synthesis of available literature, based on research questions (RQs) [41]. Since the epicenter of the current study is goal no. 2 and 6, therefore, the following RQs are defined for the study:

RQ1: How BCT assist and contribute toward attainment of SGD 2 (zero hunger) and SDG 6 (clean water and sanitation)?

RQ2: What are the gaps in current research and how those gaps can be addressed in the future studies?

To answer the RQs, we have adopted the systematic review process and a standard procedure for locating and selecting the relevant literature is followed [7, 28, 41].

Table 1 Search strings

Search string	No. of studies
"food security" OR "smart agriculture" OR "food supply chain" OR "food traceability" AND "blockchain*" OR "smart contract" OR "digital ledger*"	173
"Water*" OR "sanitation" AND "blockchain*" OR "smart contract" OR "digital ledger*"	305

3.2 Search Strategy

Search for relevant literature was conducted in two stages—*automatic* and *manual* [7]. In automatic search, the studies were located using search strings. For automatic search, Scopus and Web of Science database were used [47, 53]. As these databases cover a wide variety of quality journal publishers like Emerald, Elsevier, Taylor and Francis, Wiley and IEEE Xplore [17], hence there was no need to search individual databases later.

Based on keywords extracted from RQs, the search string used for initial automatic search were—"food security" AND "blockchain *" and "clean water and sanitation" AND "blockchain *". While reviewing the downloaded articles it was realized that few authors have used distributed ledger, digital ledger and smart contract terms instead of blockchain or BCT. Similarly, smart agriculture, food supply chain and food traceability were used in place of food security. Also, water and sanitation were used as separate keywords and not combined anywhere in most of the studies.

Hence, considering all these keywords, the final search strings used for the automatic search are shown in Table 1.

The automatic search was not bounded by any time frame, and the search string was free to locate relevant studies irrespective of their year of publication. The automatic search resulted in a total of 478 studies. 126 studies were found to be common in both the databases, and hence, they were removed; thereafter, total studies left for further analysis were 352.

3.3 Inclusion and Exclusion Criteria

Once the studies were located, they were further reviewed by reading their abstract, introduction and conclusion [34]. Decisions pertaining to study selection or rejection were based on inclusion and exclusion (I-E) criteria shown in Table 2 [8, 34]. The I-E criteria were based on the following parameters:

- Language of the study: since this research paper is in English and all the co-authors are versed with English language only, hence the studies published in English language were considered.
- Source type: call for papers, editorial notes, book review and revisions were not considered for the study.

Table 2 Research type

Evaluation research	11
Opinion paper	4
Philosophical paper	9
Solution proposal	21
Validation research	7

- Full text available: if full text of any study was not available, it was not considered for review in the current study.
- Relevance: only those studies which were focused on the role of BCT for meeting SDGs no. 2 and 6 were considered. Rest all were excluded.
- Study type: only empirical, conceptual and viewpoint studies were considered. Review studies were not considered for this SLR.

When tested for I-E criterion, it was observed that seven studies were published in language other than English; 48 articles were either call for papers or book review or editorial notes; the full text of 47 titles were not available; 161 non-relevant studies and 43 study type review studies were also excluded from the database. After applying I-E criteria total of 46 studies were left.

3.4 Full-Text Scanning and Manual Search

All 46 articles were uploaded on Mendeley. A detailed analysis of all the articles was done by all the authors. While reading the downloaded articles few missing studies were also identified by the authors. Missing studies were those relevant articles that were not located during automatic search but are cited in the studies downloaded by us. Hence, such studies were also downloaded using manual search through "Google Scholar" search engine [7].

As a result of manual search, six new studies were identified and tested for I-E criteria. After adding these studies, the total usable studies for the current SLR reached to 52. These 52 usable studies are referred as *primary studies* from here onwards.

4 Descriptive Analysis

Primary studies are now described based on year of publication, source of publication (journal), research type and sustainability dimension reported.

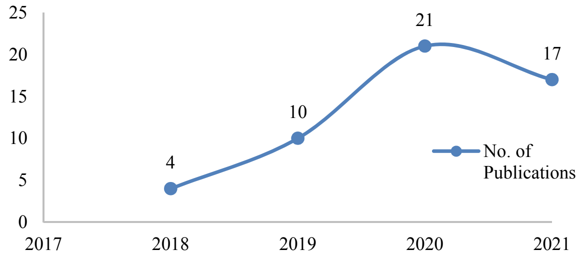


Fig. 1 Year wise distribution of studies

4.1 Publication Year Overview

During article search, no time frame was given by the authors and hence, the search engine was free to locate articles published till August. Article search resulted in 52 primary studies published from 2018 to 2021.

The growth in terms of number of published articles on "BCT application for attaining SDG no 2 and 6" from 2018 to 2021 is as follows: in 2018 total of four articles were published; in 2019 total articles were ten, which almost doubled in 2020 with 21 publications and in 2021 up to August 30, total published studies were 17. The year-wise distribution of publication is shown in Fig. 1. Increased publication trend indicates that the theme is grabbing the attention of researchers and resulted in continuous growth in terms of number of publications each year.

4.2 Research Type

Primary studies were categorized into the following research types as suggested by [40]:

- Evaluation research: studies comprised of techniques that were implemented and used in practice. Evaluation led to getting a better understanding of how the technique was implemented and brought into practice (solution implementation) and what were the consequences of the implementation in terms of benefits and drawbacks (implementation evaluation).
- Opinion paper: studies covering personal opinion of researcher related to categorization of a certain technique as good or bad and suggestions on how things should have been done.
- Philosophical paper: research articles under this category visualized a paradigm change in the way existing things were looked upon by structuring and restructuring the field based on its taxonomy or conceptual framework or further propositions.
- Solution proposal: studies proposing a solution for a problem. It could be either a novel solution or a significant extension of an existing technique. The solution entails the potential benefits and their applicability.

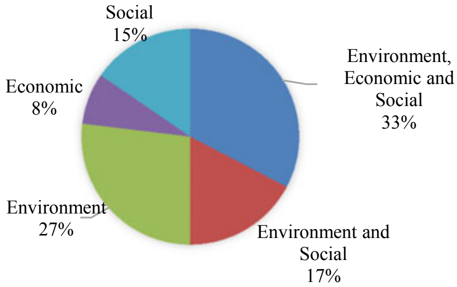


Fig. 2 Sustainability dimension reported in primary studies

- Validation research: under this category, novel techniques were investigated which were never implemented and were not in practice.

Distribution of primary studies, based on research type is shown in Table 2.

4.3 Sustainability Dimension Reported

The distribution of primary studies based on triple bottom line dimensions [11] indicated that majority of the studies (33%) have considered all three dimensions of the triple bottom line, followed by 27% studies reporting a single dimension "environment" only. A combination of "environment" and "social" has been considered by 17% of studies. Only "economic" dimension has been used in 8% of studies, whereas "social" dimension has been referred in 15% of studies. The distribution of primary studies on the basis of sustainability dimension is shown in Fig. 2.

5 Research Question Result

RQ1: How BCT assist and contribute toward attainment of SGD 2 (zero hunger) and SDG 6 (clean water and sanitation)?

The RQ1 is answered in the following phases—i. an introduction to BCT and its key capabilities and ii. application of BCT in attaining SDG 2 and 6.

Blockchain Technology—An overview

Developed in 2008 by Satoshi Nakamoto, a pseudonym [10], the blockchain is a network of blocks containing data. Each succeeding block is attached with the previous block forming a chain of information [23]. Each block has its own timestamp and is linked with the previous block through a cryptographic hash [48]. Blocks once added to blockchain network cannot be deleted [48]. Data once entered into the blocks are irreversible and not possible to edit or delete [37]. Data is validated before placing into the block [23]. For manipulating or editing data entered into a

Table 3 Key features of blockchain technology

Key feature	Description
Information transparency	The entire history of transactions is visible to every single user connected to the blockchain network [3, 18]
Trust	Data exchange between nodes* requires cryptographic proof instead of third-party authentication or mutual trust [4, 16] * Node can be any electronic device that maintains copies of decentralized distributed ledgers and keeps the entire chain functioning
Decentralization	There is no server or mainframe system which controls the entire blockchain. Hence, no one (individual or organization) can claim the ownership of the chain and trust between the parties gets improved as information transparency is attained [12, 42, 58]
Privacy	Using public-key cryptography, blockchain ensures privacy and the available content is accessible to the authorized users only [14, 15]
Data immutability	Data once entered into the blockchain network cannot be edited, modified or deleted [19]
Security	Data in blockchain is stored in encrypted form (using asymmetric cryptographic principles) which safeguards it from external and internal attacks. Also, the transmission of data within the blockchain is protected by cryptographic algorithms, which upsurge the security of data during transmissions between the nodes [43, 55]

block one need to access all previous blocks attached to it, as all the blocks in a chain are related with a hash. Hence, hacking a block or manipulating data in the block is almost impossible (Table 3).

The following are the key capabilities of BCT

Smart contract A smart contract is a decentralized automatically executable computer program code that remains stored on the blockchain and runs autonomously after fulfilling certain pre-defined conditions and verifies the contract between two parties [21, 27]. The correct code automatically validates the data legally without human intervention [21].

Distributed Ledger Technology (DLT) Blockchain is the distributed ledger system that records the data in real time into different nodes using a cryptographic algorithm that ensures the security and privacy of data [57].

Application of Blockchain Technology for attaining SDG No. 2 (Zero hunger)

According to UN the SDG no. 2 says "End hunger, achieve food security and improved nutrition and promote sustainable agriculture" [56]. The traditional food production and distribution system have several challenges and operational inefficiencies like poor beneficiary identification, non-uniform transaction records and lack of integration among stakeholders [49]. Today consumers have become more concerned about the quality and safety of food they consume, and hence, they demand real-time updated information on food items [29].

One of the ways to overcome barriers to food production and distribution and provide real-time information to consumers is through smart agriculture. Smart agriculture is BCT enabled agricultural practice which includes agri-food supply chain, agriculture logistics, food availability, food safety, food traceability, fresh food delivery and circular agriculture.

Agri-food supply chain moves from farmer (manufacturer of agricultural product) to end consumer. All the parties involved in supply chain becomes blockchain network members and each member can feed their transactions in the system and stakeholders can check required information at any point of time [2]. BCT is often used in association with other technologies like remote sensing technology and artificial intelligence (AI), to collect data at each stage of the agri-food supply chain [6]. For instance, customized sensor networks tracking location, time, temperature, and humidity levels report data to a blockchain, where the data is verified and stored [6].

DLT (a key capability of BCT) can be helpful in developing a secured food trading system with integrity and transparency in agriculture data. Data transparency refers to the property of blockchain which facilitates all the stakeholders to view the data stored in various blocks of the entire chain; if any change in ownership of the physical asset is reported then blockchain can provide information of present and past owners this feature is known as data integrity. Also, BCT reduces the number of intermediaries in the agriculture supply chain and minimizes delayed payments and high transaction lead times [13].

BCT enabled agricultural reforms that can bring “digital agricultural democratization” (DAD) lead to freedom of information (upstream and downstream) in agri-food supply chain, improve agricultural production efficiency and, ultimately, form a society where quality and nutritious food is served to all [9].

The DAD framework is an amalgamation of BCT and the circular agricultural [9]. The blockchain network automatically collects and uploads data through various types of smart devices, which expands the information set that can be used for sharing. This can solve problems such as asymmetric information, unreliable third-party institutions and poor traceability of organic food [9].

To meet the aim of zero hunger in developing nations, like India, often schemes like public distribution system (PDS) are implemented by the Government. PDS a program run by government to provide basic food commodities at subsidized prices to economically backward sections of the society [39]. Aims at providing nutritious and unadulterated food in required quantity to the needy. The entire food supply chain comprises several stakeholders from procurement of food commodities to end consumers. In traditional PDS, due to non-transparency and humanized food quality checks (which can be falsified easily), several pitfalls were observed [39]. A BCT enabled system of PDS can be helpful in ensuring the quality and quantity of food commodities at each stage. All the data get stored in the Blockchain and hence become immutable and secure [39]. Also, all the stakeholders are connected through a smart contract and only an authorized person/party can access the information [50].

Application of Blockchain Technology for attaining SDG No. 6 (Clean water and sanitation)

Limited water resources, exponential growth of water demands due to increasing world population and impact of climate change has increased pressure on water sources and their management [44]. There are two major challenges to be faced by the world in coming years—non-availability of clean and drinking water, and unavailability of water.

Clean water can be defined as—sufficient water available for at least 12 h per day which is free from fecal indicator bacteria arsenic and fluoride [22]. As per UN projections, by 2025 almost 1.8 billion people around the world will be living in no water zone [44]. A water management strategy for handling water-related issues like water scarcity, lack of water quality and water sanitation issues has become a need of the hour [25]. Clean water and sanitation can be made available to the masses through water security, smart water conservation and water resource management [25].

Water security is the process to protect sustainable access to water for humans against water contamination and water-related disasters [25] by developing suitable infrastructure for harvesting, storing, treating and transporting water for agriculture, commercial and domestic purposes [20].

Based on the re-useable nature of water, it can be categorized as—consumptive and non-consumptive. A consumptive use converts water into a form that cannot be reused, for instance, a portion of irrigation water (in agriculture) gets evaporated and cannot be reused; contrary after non-consumptive use, water can be recycled and reused, like flush water in washroom can be recycled and used for drinking and other purposes [20]. With the amalgamation of BCT and artificial intelligence, a system for water security can be developed for non-consumptive water usage [25]. BCT could lengthen the lives of water assets, reduce water leaks and lower water expenditure and loss [25].

Merely restoration and re-usage of water is not going to work, as only 0.5% of earth's water is usable [25] that demands a mechanism to be developed to predict future water consumption and water management [32]. Blockchain-based framework for smart water saving and distribution can be developed,houses (water consuming bodies can be treated as the nodes of the blockchain network, and the computation takes place at the edge of the network and provides instant information regarding water consumption[52]. The technology enables houses can predict the daily water requirement and a BCT and ICT enabled system installed in houses for real-time monitoring of water supply provide input to the authorities like Municipal Corporation [44].

Rainwater harvesting through smart grid is also gaining popularity. Instead of only consuming the available waters, now households can install a rainwater harvesting system and can become prosumers. A smart water grid is a BCT enabled water network that is shared by multiple diverse users (domestic or irrigation or industrial), who can either produce or consume water [44].

Health and hygiene have to be maintained at individual and society level. BCT-driven sanitation management system, commonly known as "smart sanitation", monitor the level of chemicals and biological compounds by analyzing biomarkers found in sewage [45]. Biosensors can be used for monitoring chemical and biological properties of sewage and real-time data can be stored in nodes of blockchain [45]. Biosensors are the devices that capture the biological signal, related to DNA, RNA, protein/enzymes, sugar level in the body and convert it into a detectable electrical signal [1]. The data stored in blockchain can be accessed by individuals or community health department, and if found any health issues, then necessary actions can be taken immediately.

Toilets embedded with biosensors and attached to a blockchain network are becoming popular now a days, such toilets are known as "smart toilets" [45].

Despite being the most versatile technological disruption, the BCT has several limitations also. Excess power consumption and hazardous impact on the climate are a few of its limitations [34]. Since the BCT is at an early stage, and we can expect that the developers and users of the technology will definitely overcome the shortcoming so in the current paper, we have focused only on the beneficial aspects of the technology.

RQ2: What are the gaps in current research and how those gaps can be addressed by future studies?

The gaps in available studies are accumulated under research type, research themes and BCT adoption barriers.

Mapping research type and research themes

Synthesis of primary studies resulted in six themes, which were mapped with research types. The details of mapping are as:

Under *agro-food supply chain* theme, 19 research papers were positioned. In this, theme evaluation research and solution proposal received good submission while opinion papers, philosophical papers and validation research received moderate responses.

Food quality and safety is the theme that has clustered 13 primary studies, with evaluation research and solution proposal having good submission while opinion paper, philosophical paper and validation research have moderate response.

Majority of the studies under *food traceability* theme were solution proposal types followed by philosophical papers, hence leaving a scope for validation research and opinion paper type.

Under *rural waste management* theme, we have located a single philosophical paper and a strong gap and opportunity is there in other research types.

Under *smart communities* theme, a dearth of literature was observed as only three primary studies are grouped under this theme. The available studies falls under

Table 4 Mapping of research type and research theme

Research theme	Research type					
	Evaluation research	Opinion paper	Philosophical paper	Solution proposal	Validation research	Total
Agro-food supply chain	5	2	3	6	3	19
Food quality and safety	5	1	2	3	2	13
Food traceability	1	0	2	6	0	9
Rural waste management	0	0	1	0	0	1
Smart communities	0	1	0	1	1	3
Water consumption management	0	0	1	5	1	7
Total	11	4	9	21	7	52

opinion paper, solution proposal and validation research. Since the smart community theme addresses the sub-themes like smart cities and smart toilet, hence future researchers are advised to focus on this theme and develop solution proposal and evaluation research.

Under *water consumption management* theme, majority of the studies were solution proposal, whereas no study is evaluation or opinion-based. Hence, future researchers are advised to conduct evaluation-based studies under this theme.

A summary of mapping of primary studies based on research type and research theme is shown in Table 4. Findings of the Table 4 reveals that validation research type needs to be explored more by future researchers. Also, the under explored research themes are rural waste management and smart communities.

Blockchain adoption barriers

Analysis of primary studies highlighted the following challenges in implementing BCT: infrastructure performance and scalability, standardization and interoperability, data security and privacy, legal and regulatory issues in the smart contract [48], lack of commercially available BCT solutions [24] and cost of BCT adoption and implementation [35].

DLT and smart contract are key characteristics of BCT which are vital for ensuring SDG no. 2 and 6. However, when it comes to practical implementation, permission design and transaction capacity, data accessibility for DLT and the consensus algorithm's selection become the challenge [13]. Similarly, in smart contact, the challenge is to identify and fix the bugs due to the irreversible nature of blockchain [13].

Agriculture systems in developing countries have been characterized by a low technology level. The principal reasons are the high cost of technologies and the difficulties to integrate the systems with the dynamics of the current business world [35].

The greatest challenge faced by bodies (Government or Non-Government) with respect to water and sanitation is the lack of data and a lack of methodologies and systems to enable data collection and management [22]. Poor maintenance of water supply systems is a looming problem in many low- and high-income countries [22]. Also developing smart toilets can be a costly affair for underdeveloped and developing nations.

6 Conclusion

This review study is an attempt to summarize the studies published in the domain of BCT application for SD. To gain an in-depth understanding of the topic, two RQs were developed. The SLR methodology was adopted to answer the RQs. The studies obtained as a result of automatic and manual literature search were passed through I-E criteria, and a total of 52 useful studies were obtained, which are termed as primary studies. A descriptive analysis of primary studies was conducted, and then, RQs were answered.

Based on in-depth analysis of primary studies, it can be said that although the trend of publication is very optimistic and research publications are gaining momentum, still there is a vast opportunity to compare and contrast different areas of knowledge to gain an insight into the possibilities which is at present very scattered [14].

The infancy of BCT is the main limitation of the study, as it is also reflected in scarcity of available literature on application of BCT for SD [54].

References

1. Ali J, Najeeb J, Asim Ali M, Farhan Aslam M, Raza A (2017) Biosensors: their fundamentals, designs, types and most recent impactful applications: a review. J Biosens Bioelectron 08(01):1–9. https://doi.org/10.4172/2155-6210.1000235
2. Antonucci F, Figorilli S, Costa C, Pallottino F, Raso L, Menesatti P (2019) A review on blockchain applications in the agri-food sector. J Sci Food Agric 99(14):6129–6138. https://doi.org/10.1002/jsfa.9912
3. Astill J, Dara RA, Campbell M, Farber JM, Fraser EDG, Sharif S, Yada RY (2019) Transparency in food supply chains: a review of enabling technology solutions. Trends Food Sci Technol 91:240–247. https://doi.org/10.1016/j.tifs.2019.07.024
4. Bai C, Sarkis J (2020) A supply chain transparency and sustainability technology appraisal model for blockchain technology. Int J Prod Res 58(7):2142–2162. https://doi.org/10.1080/00207543.2019.1708989
5. Blesh J, Hoey L, Jones AD, Friedmann H, Perfecto I (2019) Development pathways toward "zero hunger." World Dev 118:1–14. https://doi.org/10.1016/j.worlddev.2019.02.004

6. Bumblauskas D, Mann A, Dugan B, Rittmer J (2020) A blockchain use case in food distribution: do you know where your food has been? Int J Inf Manage 52. https://doi.org/10.1016/j.ijinfomgt.2019.09.004
7. Busalim AH, Hussin ARC (2016) Understanding social commerce: a systematic literature review and directions for further research. Int J Inf Manage 36(6):1075–1088. https://doi.org/10.1016/j.ijinfomgt.2016.06.005
8. Casino F, Dasaklis TK, Patsakis C (2019) A systematic literature review of blockchain-based applications: current status, classification and open issues. Telematics Inform 36:55–81. https://doi.org/10.1016/j.tele.2018.11.006
9. Chen Y, Li Y, Li C (2020) Electronic agriculture, blockchain and digital agricultural democratization: origin, theory and application. J Clean Prod 268. https://doi.org/10.1016/j.jclepro.2020.122071
10. Cole R, Stevenson M, Aitken J (2019) Blockchain technology: implications for operations and supply chain management. Supply Chain Manage Int J 24(4):469–483. https://doi.org/10.1108/SCM-09-2018-0309
11. Elkington J (1997) Cannibals with forks: the triple bottom line of 21st century business. Capstone Publishing Limited, Oxford
12. Erol I, Ar IM, Ozdemir AI, Peker I, Asgary A, Medeni IT, Medeni T (2021) Assessing the feasibility of blockchain technology in industries: evidence from Turkey. J Enterp Inf Manag 34(3):746–769. https://doi.org/10.1108/JEIM-09-2019-0309
13. Feng H, Wang X, Duan Y, Zhang J, Zhang X (2020) Applying blockchain technology to improve agri-food traceability: a review of development methods, benefits and challenges. J Clean Prod 260. https://doi.org/10.1016/j.jclepro.2020.121031
14. Fernandez-Vazquez S, Rosillo R, De La Fuente D, Priore P (2019) Blockchain in FinTech: a mapping study. Sustainability 11(22). https://doi.org/10.3390/su11226366
15. Fraga-Lamas P, Fernandez-Carames TM (2019) A review on blockchain technologies for an advanced and cyber-resilient automotive industry. IEEE Access 7:17578–17598. https://doi.org/10.1109/ACCESS.2019.2895302
16. França ASL, Amato Neto J, Gonçalves RF, Almeida CMVB (2020) Proposing the use of blockchain to improve the solid waste management in small municipalities. J Clean Prod 244. https://doi.org/10.1016/j.jclepro.2019.118529
17. Franciosi C, Voisin A, Miranda S, Riemma S, Iung B (2020) Measuring maintenance impacts on sustainability of manufacturing industries: from a systematic literature review to a framework proposal. J Clean Prod 260. https://doi.org/10.1016/j.jclepro.2020.121065
18. Fu B, Shu Z, Liu X (2018) Blockchain enhanced emission trading framework in fashion apparel manufacturing industry. Sustainability 10(4). https://doi.org/10.3390/su10041105
19. Ghobakhloo M (2018) The future of manufacturing industry: a strategic roadmap toward Industry 4.0. J Manuf Technol Manage 29(6):910–936. https://doi.org/10.1108/JMTM-02-2018-0057
20. Grant SB, Saphores JD, Feldman DL, Hamilton AJ, Fletcher TD, Cook PLM et al (2012) Taking the "waste" out of "wastewater" for human water security and ecosystem sustainability. Science 337(6095):681–686. https://doi.org/10.1126/science.1216852
21. Grover P, Kar AK, Janssen M, Ilavarasan PV (2019) Perceived usefulness, ease of use and user acceptance of blockchain technology for digital transactions–insights from user-generated content on Twitter. Enterp Inf Syst 13(6):771–800. https://doi.org/10.1080/17517575.2019.1599446
22. Guppy L, Mehta P, Qadir M (2019) Sustainable development goal 6: two gaps in the race for indicators. Sustain Sci 14(2):501–513. https://doi.org/10.1007/s11625-018-0649-z
23. Heinrich M, Scotti F, Booker A, Fitzgerald M, Kum KY, Löbel K (2019) Unblocking high-value botanical value chains: is there a role for blockchain systems? Front Pharmacol 10:1–8. https://doi.org/10.3389/fphar.2019.00396
24. Hughes L, Dwivedi YK, Misra SK, Rana NP, Raghavan V, Akella V (2019) Blockchain research, practice and policy: applications, benefits, limitations, emerging research themes and research agenda. Int J Inf Manage 49:114–129. https://doi.org/10.1016/j.ijinfomgt.2019.02.005

25. Iyer LS, Giri SV (2020) Harnessing technology for mitigating water woes in the city of Bengaluru. J Phys Conf Ser 1427(1). https://doi.org/10.1088/1742-6596/1427/1/012004
26. Kim JS, Shin N (2019) The impact of blockchain technology application on supply chain partnership and performance. Sustainability 11(21). https://doi.org/10.3390/su11216181
27. Kimani D, Adams K, Attah-Boakye R, Ullah S, Frecknall-Hughes J, Kim J (2020) Blockchain, business and the fourth industrial revolution: whence, whither, wherefore and how? Technol Forecast Soc Chang 161. https://doi.org/10.1016/j.techfore.2020.120254
28. Kitchenham B, Pearl Brereton O, Budgen D, Turner M, Bailey J, Linkman S (2009) Systematic literature reviews in software engineering—a systematic literature review. Inf Softw Technol 51(1):7–15. https://doi.org/10.1016/j.infsof.2008.09.009
29. Kittipanya-ngam P, Tan KH (2020) A framework for food supply chain digitalization: lessons from Thailand. Prod Plann Control 31(2–3):158–172. https://doi.org/10.1080/09537287.2019.1631462
30. Kshetri N (2018) 1 Blockchain's roles in meeting key supply chain management objectives. Int J Inf Manage 39:80–89. https://doi.org/10.1016/j.ijinfomgt.2017.12.005
31. Kshetri N (2019) Blockchain and the economics of food safety. IT Prof 21(3):63–66. https://doi.org/10.1109/MITP.2019.2906761
32. Li H, Chen X, Guo Z, Xu J, Shen Y, Gao X (2021) Data-driven peer-to-peer blockchain framework for water consumption management. Peer-to-Peer Netw Appl. https://doi.org/10.1007/s12083-021-01121-6
33. Lindsey TC (2011) Sustainable principles: common values for achieving sustainability. J Clean Prod 19(5):561–565. https://doi.org/10.1016/j.jclepro.2010.10.014
34. Lund EH, Jaccheri L, Li J, Cico O, Bai X (2019) Blockchain and sustainability: a systematic mapping study. In: 2019 IEEE/ACM 2nd international workshop on emerging trends in software engineering for blockchain (WETSEB), pp 16–23. https://doi.org/10.1109/WETSEB.2019.00009
35. Morales MLV, Elkader MAA (2020) Logistics 4.0 technologies in agriculture systems: potential impacts in the SDG. In: International association for management of technology, pp 976–989
36. Murray CJL, Vos T, Lozano R, Naghavi M, Flaxman AD, Michaud C et al (2012) Disability-adjusted life years (DALYs) for 291 diseases and injuries in 21 regions, 1990–2010: a systematic analysis for the global burden of disease study 2010. Lancet 380(9859):2197–2223. https://doi.org/10.1016/S0140-6736(12)61689-4
37. Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system. Retrieved from https://direct.mit.edu/artl/article/23/4/552-557/2870
38. Nikolakis W, John L, Krishnan H (2018) How blockchain can shape sustainable global value chains: an evidence, verifiability, and enforceability (EVE) framework. Sustainability 10(11). https://doi.org/10.3390/su10113926
39. Pawar RS, Sonje SA, Shukla S (2021) Food subsidy distribution system through Blockchain technology: a value focused thinking approach for prototype development. Inf Technol Dev 27(3):470–498. https://doi.org/10.1080/02681102.2020.1841714
40. Petersen K, Feldt R, Mujtaba S, Mattsson M (2008) Systematic mapping studies in software engineering. In: Proceedings of the 12th international conference on evaluation and assessment in software engineering (EASE'08), vol 17. BCS Learn-Ing and Development Ltd., Italy, pp 68–77
41. Petticrew M, Roberts H (2006) Systematic reviews in the social sciences: a practical guide. In: Systematic reviews in the social sciences: a practical guide. https://doi.org/10.1002/9780470754887
42. Prashar D, Jha N, Jha S, Lee Y, Joshi GP (2020) Blockchain-based traceability and visibility for agricultural products: a decentralized way of ensuring food safety in India. Sustainability 12(8). https://doi.org/10.3390/su12083497
43. Queiroz MM, Telles R, Bonilla SH (2019) Blockchain and supply chain management integration: a systematic review of the literature. Supply Chain Manag 25(2):241–254. https://doi.org/10.1108/SCM-03-2018-0143

44. Ramsey E, Pesantez J, Fasaee MAK, DiCarlo M, Monroe J, Berglund EZ (2020) A smart water grid for micro-trading rainwater: hydraulic feasibility analysis. Water 12(11). https://doi.org/10.3390/w12113075
45. Rary E, Anderson SM, Philbrick BD, Suresh T, Burton J (2020) Smart sanitation—biosensors as a public health tool in sanitation infrastructure. Int J Environ Res Public Health 17(14). https://doi.org/10.3390/ijerph17145146
46. Sachs JD, Schmidt-Traub G, Mazzucato M, Messner D, Nakicenovic N, Rockström J (2019) Six Transformations to achieve the Sustainable Development Goals. Nat Sustain 2(9):805–814. https://doi.org/10.1038/s41893-019-0352-9
47. Schanes K, Dobernig K, Gözet B (2018) Food waste matters—a systematic review of household food waste practices and their policy implications. J Clean Prod 182:978–991. https://doi.org/10.1016/j.jclepro.2018.02.030
48. Shen C, Pena-Mora F (2018) Blockchain for cities—a systematic literature review. IEEE Access 6:76787–76819. https://doi.org/10.1109/ACCESS.2018.2880744
49. Singh SK, Jenamani M, Dasgupta D, Das S (2021) A conceptual model for Indian public distribution system using consortium blockchain with on-chain and off-chain trusted data. Inf Technol Dev 27(3):499–523. https://doi.org/10.1080/02681102.2020.1847024
50. Son NM, Nguyen TL, Huong PT, Hien LT (2021) Novel system using blockchain for origin traceability of agricultural products. Sens Mater 33(2). https://doi.org/10.18494/SAM.2021.2490
51. Sriyono E (2020) Digitizing water management: toward the innovative use of blockchain technologies to address sustainability. Cogent Eng 7(1). https://doi.org/10.1080/23311916.2020.1769366
52. Thakur T, Mehra A, Hassija V, Chamola V, Srinivas R, Gupta KK, Singh AP (2021) Smart water conservation through a machine learning and blockchain-enabled decentralized edge computing network. Appl Soft Comput 106. https://doi.org/10.1016/j.asoc.2021.107274
53. Thomé AMT, Scavarda LF, Scavarda AJ (2016) Conducting systematic literature review in operations management. Prod Plann Control 27(5):408–420. https://doi.org/10.1080/09537287.2015.1129464
54. Tiron-Tudor A, Deliu D, Farcane N, Dontu A (2021) Managing change with and through blockchain in accountancy organizations: a systematic literature review. J Organ Chang Manag 34(2):477–506. https://doi.org/10.1108/JOCM-10-2020-0302
55. Tiwari P, Ilavarasan PV, Punia S (2021) Content analysis of literature on big data in smart cities. Benchmarking Int J 28(5):1837–1857. https://doi.org/10.1108/BIJ-12-2018-0442
56. UN (2015) Transforming our world: the 2030 agenda for sustainable development. United Nations. https://doi.org/10.1163/157180910X12665776638740
57. Vujičić S, Hasanspahić N, Car M, Čampara L (2020) Distributed ledger technology as a tool for environmental sustainability in the shipping industry. J Mar Sci Eng 8(5):366. https://doi.org/10.3390/jmse8050366
58. Wang Q, Su M (2020) Integrating blockchain technology into the energy sector—from theory of blockchain to research and application of energy blockchain. Comput Sci Rev 37. https://doi.org/10.1016/j.cosrev.2020.100275

KNN-Based Decision Model for Device Argumentation in Ambient Assisted Living

G. S. Madhan Kumar, S. P. Shiva Prakash, and Kirill Krinkin

Abstract Aging population has increased in the past few decades with the rise in social and economic challenges. In particular, people wish to live an independent life without giving most of the burden to caretakers. This is where the Ambient Assisted Living (AAL) has gained momentum that allows elderly people to lead an independent and safe lifestyle by performing their daily activities. Such an AAL environment which consists of heterogeneous sensors and actuators helps in monitoring the daily activities carried out by elderly people throughout the day. These smart objects interact among themselves by using higher-level information generated as a result of processing the sensor values on chips to decide their own course of actions toward their goals. Thus, these smart objects which are now speaking and hearing objects ("things") interact through argumentation. These objects could communicate and argue with each other about a particular activity in progress and to find the understanding of the present state and take decisions and act accordingly. Thus argumentation as a common basis for interactions among these smart objects, this paper focuses on implementing a KNN-based decision model for device argumentation about an activity identification and to take actions accordingly. The proposed solution shows that KNN gives an accuracy of 72.28%, Precision of 85%, Recall of 42.3% and F1-score of 38.40% for classifying ambiguous device argumentation in AAL.

Keywords Ambient Assisted Living · Device argumentation · Decision model · Machine learning · Activity

G. S. Madhan Kumar · S. P. Shiva Prakash (✉)
JSS Science and Technology University Mysuru, Mysuru, India
e-mail: shivasp@jssstuniv.in

K. Krinkin
Saint Petersburg Electrotechnical University "LETI", Saint Petersburg, Russia
e-mail: kirill@krinkin.com

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_38

1 Introduction

Ambient Assisted Living (AAL) creates an environment using the communication and information technologies in which the person performs daily routine activities and enables them to stay active and live independently. AAL intends to improve elderly people's quality of life and through the application of technology to support their daily requirements in their future. The AAL environment consists of heterogeneous smart objects that capture values continuously to monitor the activities of the person who resides in it. The higher-level information generated by processing the sensor values on chips are used by these smart objects to interact among themselves and to decide their own course of actions toward the goals. These smart objects argue with each other for the occurrence of a particular activity in progress and to find the perceived state of the affairs and, to act and take decision accordingly. Thus argumentation as a common basis for interactions among these smart objects, this paper focuses on implementing a KNN-based decision model for device argumentation during an activity identification, and thus, implements an smart ecosystem that enables these smart objects to interact and support decision making whenever required as shown in Fig. 1.

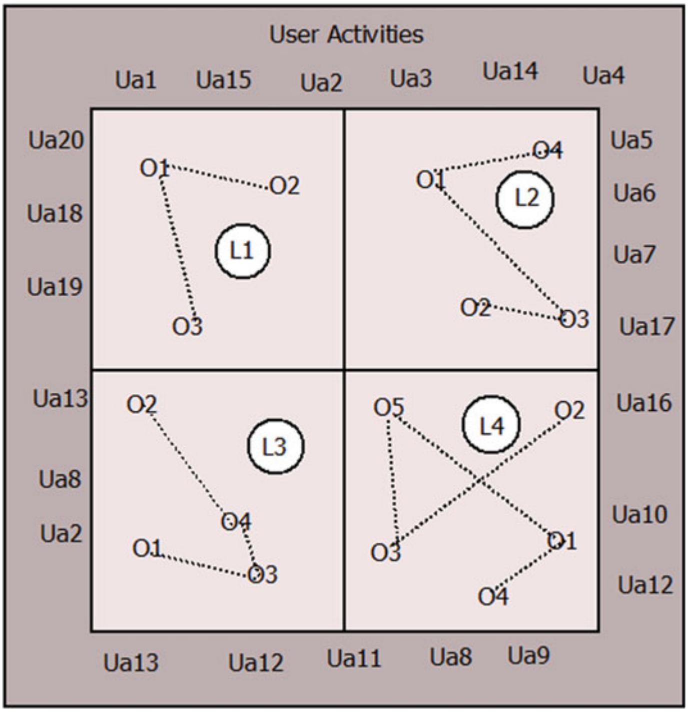


Fig. 1 Ambient Assisted Living environment

The rest of the paper is organized as follows: The related works carried out are presented Sect. 2. Section 3 describes problem statement. The system model is presented in Sect. 4. Section 5 discusses the proposed model design. The results obtained and its discussions are presented in Sect. 6. Section 7 discusses the conclusion and future work.

2 Related Works

Parsons and Green [1] proposed a qualitative probabilistic networking system of argumentation that captures the reasoning about expected utilities of actions and the propagation of cooperation between actions. It extends the previous work of systems argumentation with qualitative probabilities reasoning. Jung et al. [2] presented a computational distributed constraint satisfaction problem (DCSP) model for investigating negotiation via argumentation (NVA) to study argumentation properties, and to formulate. Carneiro et al. [3] developed a CBR system prototype that uses machine learning capabilities to exploit the services of users in an Ambient Assisted Living environment, i.e., to dynamically evolve as the user's interaction with the environment changes its habits or routines. Ding et al. [4] proposed a review on sensor technology, strengths and limitations of different sensor technologies and opportunities and challenges in smart homes. Rashidi and Mihailidis [5] summarize the tools of Ambient Assisted Living (AAL) used for older adults based on the ambient intelligence paradigm and also look at current and future challenges. Sernani et al. [6] proposed the possibility of using 3D virtual environments similar to robotics with 3D simulators. Liu et al. [7] proposed a simulator for indoor human movement to facilitate developments and validations of Ambient Assisted Living (AAL) environment. Li et al. [8] explored many AAL research on older adults. To achieve better facilitation to old adults and aging society, the studies showed cognitive aspects of AAL. Rahim Mohammad Forkan et al. [9] developed a Hidden Markov model-based approach for detecting abnormalities in daily activities to identify irregularity in routine behaviors and to predict vital signs changes in the future, an smoothing technique was also developed. Brunmeir et al. [10] proposed a decision making engine that observes a person to learn and categorize human behavior and also replicate the person's actions. The altering behavioral patterns are recognized and adapted to actions for changing situations. The aim is to enhance the life quality of elderly persons or persons with special needs to automate certain activities. Calvaresi et al. [11] proposed a overview of the AAL domain by comprehensive systematic analysis over 10 years of relevant literature that focuses on the needs of the stakeholders and bridging the gap of existing reviews. Lippi et al. [12] worked on building interacting through argumentation Speaking Objects—"things" to show how human dialog naturally fit in SIoT to cooperate and coordinate. Lippi et al. [13] proposed smart sensors and actuator's vision that is capability of understanding and reporting about the happening and about the possibly that can happen. Sharma et al. [14] focused on real-time continuous annotation of emotions exhibited while watching various videos that were used to generate a Continuously Annotated Signals of Emotion (CASE) dataset. To generate the dataset, an interface was developed that was a novel, intuitive joystick-based annotation that simultaneously captured valence and arousal. ECG, BVP, EMG (3x), GSR (or EDA), respiration, and skin temperature sensors were used to parallelly capture eight high-quality, synchronized physiological values. 15 males and 15 females participated in generating the physiological and annotation data that form the dataset. Based on annotation and physiological data, the emotions induced were validity. Alsulami et al. [15], in total, reviewed, identified and discussed 14 factors that played an essential role in AAL adoption in Saudi Arabia. These factors help for the identification of the main dimensions and their characteristics and responsibilities in AAL adaption. For better planning and understanding, the work provided a ground to reach a better stage in adapting AAL. Chen et al. [16] presented a sensor-based human activity recognition survey using deep learning methods. Also introduces the sensory data of multi-modality for evaluation in different challenge tasks. Banjarey et al. [17] proposes a survey on some recent research papers to find different approaches used for analyzing human activities. Bouchabou et al. [18] presented recent human activity recognition algorithms, works, challenges and taxonomy in a smart home through ambient sensors. Vimarlund et al. [19] proposed a method for identifying services for older adults in AAL living environments to diminish social isolation and loneliness during COVID-19 pandemic. It also examined the challenges and needs of seniors who reside in AAL. Qureshi et al. [20] used the wearable technologies for early detect of cardiovascular diseases for the treatment of patients and, finally, using biomedical data originated from wearable devices and other medical databases, an deep learning-based applications is proposed.

3 Problem Statement

Device argumentation among smart objects in the AAL environment occurs during an activity identification. These arguments can lead to ambiguous situations for identifying the activity that occurred. Thus there is a need to build a decision support system for the device argumentation between smart speaking objects during identification of the activity in AAL environment.

4 System Model

Consider a system having $S = \{s_1, s_2, \ldots, s_{14}\}$ sensors attached to different objects $O = \{o_1, o_2, \ldots, o_n\}$ located in various location $L = \{l_1, l_2, \ldots, l_n\}$ that captures activities $A = \{a_1, a_2, \ldots, a_m\}$ within a IoT environment. Also let $S_e = \{S_{e1}, S_{e2}, \ldots, S_{e10}\}$ be the sensors attached to the user's body with respect to activity to capture $E = \{e_1, e_2, e_3, e_4\}$ emotions. Table 1 gives the various system model variables and its descriptions.

Let U_a be an activity that has occurred at time t_o in location l of L. O_{ne} be the nearby objects during activity occurrence U_a. Let $O_a \subseteq O_{ne}$ be objects that identifies the activity occurrence of U_a and $O_{na} \subseteq O_{ne}$ be objects that identifies different

Table 1 System model variables and its descriptions

Variables	Descriptions
S	Set of sensors
O	Set of objects
L	Various locations
A	Various activities performed by user
S_e	Set of sensors capturing emotions
E	Emotions exhibited by user
U_a	Activity at time t_o
O_{ne}	Nearby sensors
O_a	Objects that identifies the activity occurrence of u_a at time t_o
O_{na}	Objects that identifies different activity occurrence at time t_o
O_{sp}	Supporting objects
E	Various emotions
A_g	Argument objective function
SM	System model

activity occurrence at the same time t_o. Hence, the system model SM is

$$\begin{aligned} \mathrm{SM} &= \sum_{1}^{n} O^n(U_a, t_o)|l_n) \\ &= \lim_{1\to m} f(A^m \cdot E((O_a \subseteq O_{ne}) + (O_{na} \subseteq O_{ne}))) \end{aligned} \tag{1}$$

4.1 Problem Formulation

During an activity occurrence of U_a at time t_o in location $l \exists L$, O_{ne} nearby objects captures the sensor values. $O_a \subseteq O_{ne}$ objects infer the activity U_a has occurred at time t_o and $O_{na} \subseteq O_{ne}$ objects infer an different activity has occurred at the same time t_o resulting in device argumentation among O_a and O_{na} for identification of activity U_a at time t_o. Hence the objective function of argument A_g is,

$$A_g = \lim_{1\to m} f(A^m \cdot E((O_a \subseteq O_{ne}) + (O_{na} \subseteq O_{ne}))) \tag{2}$$

subjected to

$$A_m = f(O, S) \tag{3}$$

and

$$E = f(S_e, U_a) \tag{4}$$

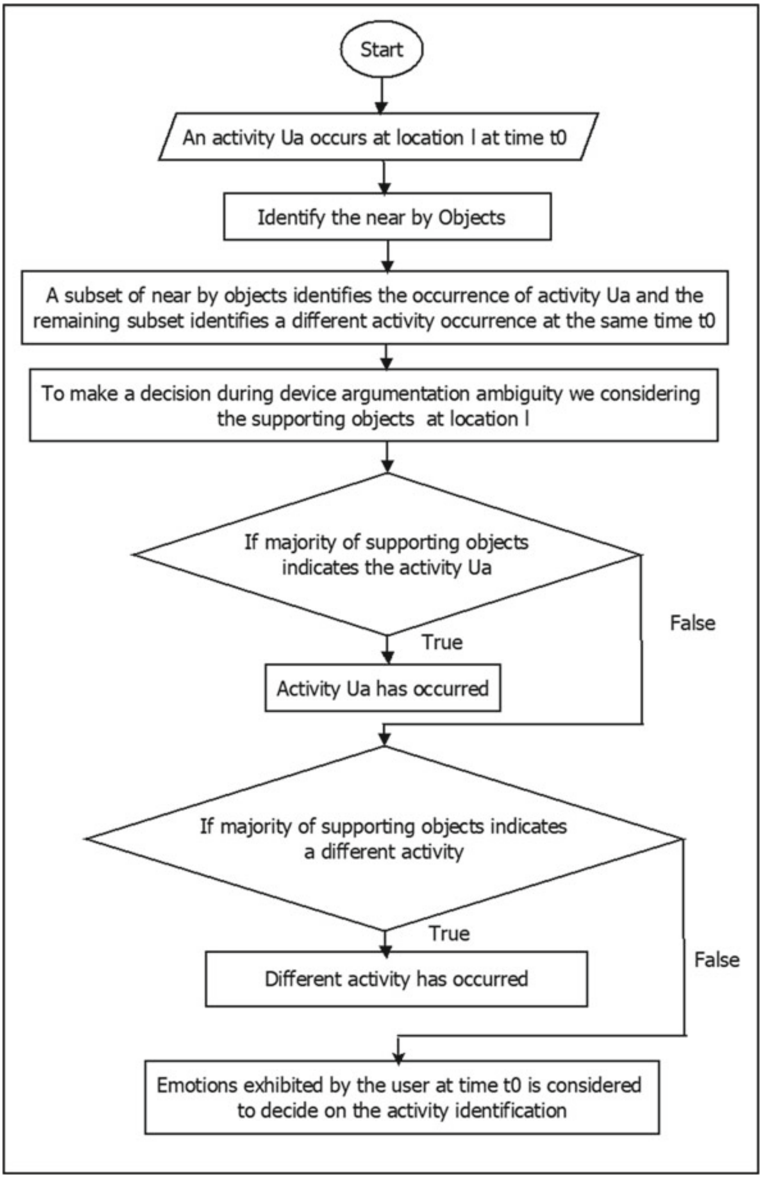


Fig. 2 Flowchart for the proposed method

where activities are identified based on the objects O and the sensor S attached to each objects, and the emotions exhibited by the users during activity occurrence respectively.

Table 2 Dataset features and its descriptions

Features	Descriptions
Objects	Set of objects in AAL environment
Sensors	Set of sensors attached to the objects
Value	Sensor values
Location	Various location within AAL environment
Status	Sensor values status—high, mild or low
Valence	Set of sensors capturing emotions
Arousal	Emotions exhibited by user
ecg	Electrocardiogram sensor
bvp	Blood volume pulse detection sensor
gsr	Galvanic skin response sensor
rsp	Respiration sensor
skt	Skin temperature sensor
emg_zygo	Electromyography-zygomaticus
emg_coru	Electromyography-corrugator
emg_trap	Electromyography-trapezius sensors
emotion	Emotion exhibited by the user
Activity	Activities performed by the user (target)

5 Proposed Model Design

This section presents the proposed decision model for device argumentation among O_a and O_{na} for identification of activity U_a at time t_o. To make a decision during device argumentation ambiguity, we consider O_{sp} supporting objects in the location $l \exists L$ at time t_o. If the majority of O_{sp} indicates the activity as U_a, then it is considered as activity U_a has occurred at time t_o. Else a different activity has occurred at time t_o. If there are an equal number of supporting sensors for occurrence of activity u_a and a different activity at same time t_o, we consider the emotion exhibited at time t_o to decide about the activity. The proposed method uses k-Nearest Neighbor (KNN) for decision making. Figure 2 shows the flowchart of the proposed method for device argumentation.

6 Results and Discussions

The results obtained are presented and analyzed in this section.

The experiment was implemented using Anaconda Jupyter version 3 with RAM size of 16 GB and processor with 2.6 Hz Speed. The dataset used for the proposed work is prepared by combining the dataset extracted from the simulator which is

Table 3 Performance metric of proposed method

Model	Performance metric	Score (%)
KNN	Accuracy	72.28
	Precision	85.00
	Recall	42.3
	F1-Score	38.40

a multi-classification dataset and the CASE [11] dataset. The dataset has a total of 16 features and 1 labeled class. Table 2 gives the dataset features and target. In this work, k-Nearest Neighbor (KNN) machine learning algorithm-based decision model for device argumentation was considered among the smart speaking objects in AAL environment for resolving ambiguous situations for identification of the activity occurred. Performance metrics like Accuracy, Precision, Recall and F1-Score are used as measurement for the result evaluation. Table 3 gives the performance metric of the proposed method.

The proposed method uses KNN-based decision model for device argumentation among O_a and O_{na} for identification of activity U_a at time t_o. To make a decision during device argumentation ambiguity, O_{sp} supporting objects in the location $l \exists L$ at time t_o are considered. If the majority of O_{sp} indicates the activity as U_a, then it is considered as activity U_a has occurred at time t_o. Else a different activity has occurred at time t_o. If there are an equal number of supporting sensors for occurrence of activity u_a and a different activity at same time t_o, the emotion exhibited by the user at time t_o to decide about the activity are considered.

Figure 3 shows the performance metric of our proposed method to produce an optimal decision model. Figures show that the proposed method gives an accuracy of **72.28%**, Precision score of **85%**, Recall score of **42.3%** and F1-Score of **38.4%**.

7 Conclusion and Future Work

AAL, consisting of heterogeneous smart objects, intends to improve elderly people's life quality by continuously capturing sensor values to monitor the activities of the person who resides in it. These heterogeneous smart objects argue with each other for the occurrence of a particular activity in progress and to identify the state of the affairs to make decision accordingly. This work has proposed a KNN-based decision model for device argumentation among the smart speaking objects in AAL environment during ambiguous situations, for identification of the activity that occurred, using machine learning methods. The obtained results give an Accuracy of **72.28%** and Precision of **85%**. In the future, various other artificial intelligence and machine learning methods along with reasoning models can be proposed for device argumentation among the smart speaking objects in the AAL environment.

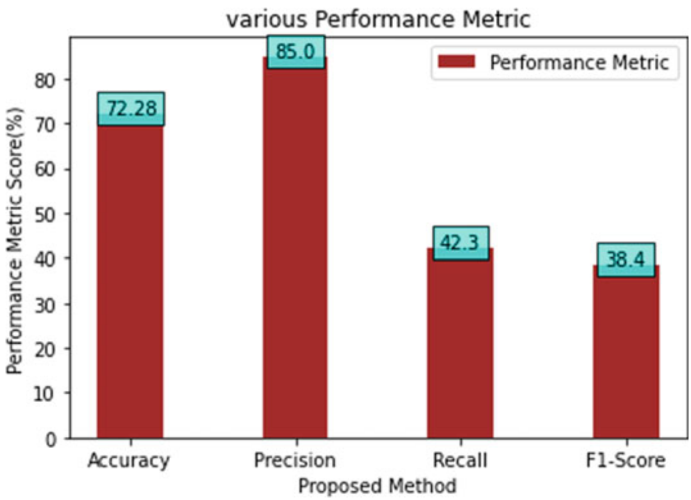


Fig. 3 Performance metric of proposed method

Acknowledgements This work was carried out under the "Development program of ETU "LETI" within the framework of the program of strategic academic leadership" Priority-2030 No. 075-15-2021-1318 on 29 Sept 2021.

References

1. Parsons S, Green S (1999) Argumentation and qualitative decision making. In: European conference on symbolic and quantitative approaches to reasoning and uncertainty. Springer, pp 328–339. https://doi.org/10.1007/3-540-48747-6_30
2. Jung H, Tambe M, Kulkarni S (2001) Argumentation as distributed constraint satisfaction: applications and results. In: Proceedings of the fifth international conference on autonomous agents, pp 324–331
3. Carneiro D, Novais P, Costa R, Neves J (2009) Case-based reasoning decision making in ambient assisted living. In: Proceedings of the international work-conference on artificial neural networks. Springer, pp 788–795
4. Ding D, Cooper RA, Pasquina PF, Fici-Pasquina L (2011) Sensor technology for smart homes. Maturita 69(2):131–136
5. Rashidi P, Mihailidis A (2012) A survey on ambient-assisted living tools for older adults. IEEE J Biomed Health Inform 17(3):579–590
6. Sernani P, Claudi A, Calvaresi P, Accattoli D, Tofani R, Dragoni AF (2014) Using 3D simulators for the ambient assisted living. In: Proceedings of the 3rd international workshop on artificial intelligence and assistive medicine, pp 16–20
7. Liu W, Shoji Y, Shinkuma R (2015) An indoor-movement simulator for ambient assisted living systems. IEEE Globecom workshops (GC Wkshps)
8. Li R, Lu B, McDonald-Maier KD (2015) Cognitive assisted living ambient system: a survey. Digit Commun Netw 1(4):229–252

9. Forkan AR, Khalil I, Tari Z, Foufou S, Bouras A (2015) A context-aware approach for long-term behavioural change detection and abnormality prediction in ambient assisted living. Pattern Recogn 48(3):628–641
10. Brunmeir D, Kropf J (2015) Decision making in ambient assisted living environments based on uncertain and fuzzy data
11. Calvaresi D, Cesarini D, Sernani P, Marinoni M, Dragoni AF, Sturm A (2017) Exploring the ambient assisted living domain: a systematic review. J Ambient Intell Hum Comput 8(2):239–257
12. Lippi M, Mamei M, Mariani S, Zambonelli F (2017) An argumentation-based perspective over the social IoT. IEEE Internet Things J 5(4):2537–2547
13. Lippi M, Mamei M, Mariani S, Zambonelli F (2017) Coordinating distributed speaking objects. In: IEEE 37th international conference on distributed computing systems (ICDCS). IEEE, pp 1949–1960. https://doi.org/10.1109/ICDCS.2017.282
14. Sharma K, Castellini C, van den Broek EL, Albu-Schaeffer A, Schwenker F (2019) A dataset of continuous affect annotations and physiological signals for emotion analysis. Sci Data 6(1):1–13
15. Alsulami MH, Alsaqer MS, Atkins AS (2020) Decision-making framework for using ambient assisted living. Int J Pervas Comput Commun
16. Chen K, Zhang D, Yao L, Guo B, Yu Z, Liu Y (2021) Deep learning for sensor-based human activity recognition: overview, challenges and opportunities. ACM Comput Surv (CSUR) 54(4):1–40
17. Banjarey K, Sahu SP, Dewangan DK (2021) A survey on human activity recognition using sensors and deep learning methods. In: Fifth international conference on computing methodologies and communication (ICCMC 2021), pp 1610–1617
18. Bouchabou D, Nguyen SM, Lohr C, LeDuc B, Kanellos I (2021) A survey of human activity recognition in smart homes based on IoT sensors algorithms: taxonomies, challenges, and opportunities with deep learning. Sensors 21(18):6037
19. Vimarlund V, Borycki EM, Kushniruk AW, Avenberg K (2021) Ambient assisted living: identifying new challenges and needs for digital technologies and service innovation. Yearbook Med Inform 30(01):141–149
20. Qureshi MA, Qureshi KN, Jeon G, Piccialli F (2021) Deep learning-based ambient assisted living for self-management of cardiovascular conditions. Neural Comput Appl 1–9

Intelligent Systems in Finance and Stock Market

Factors Influencing Behavioural Intentions Towards Investment in Cryptocurrency: A Study on Generation Z Female of India

Garima Joshi, Prabodh Narayan Gour, Pravesh Soti, Ankit Aggarwal, Harshwardhan Singh, and Sushil Kumar Gupta

Abstract Drawn upon the premise of technology acceptance model (TAM), innovation diffusion theory (IDT) and unified theory of acceptance and use of technology (UTAUT), the current study aims at identifying and prioritizing the reasons behind the behavioural intentions to invest in cryptocurrency among Generation *Z* females of India. Theoretical model developed for the study hypothesized the following precursors of BI-perceived usefulness, perceived ease of use, trust, price value, hedonic motivation, facilitating conditions. Validated scales were used for measuring the study constructs and data was collected through Google form from 276 female students pursuing B.Tech. or equivalent degree from Institute of National Importance of India. Using structural modelling approach theory, model was tested and it was found that perceived usefulness has highest impact on BI, whereas price value has a negative impact.

Keywords Cryptocurrency · Behavioural intentions · Perceived usefulness · Trust · Price value · Hedonic motivation · Facilitating conditions

G. Joshi (✉)
Management Department, DNS College of Engineering and Technology, Amorha, Uttar Pradesh, India
e-mail: f18garimaj@iima.ac.in

P. N. Gour
Department of Management and Commerce, Khandelwal College, Bareilly, Uttar Pradesh, India

P. Soti
Department of Management Studies, Krupanidhi Group of Institutions, Bangalore, Karnataka 560066, India

A. Aggarwal
Graphic Era (Deemed to be University), Dehradun, Uttarakhand, India

H. Singh
DNS College of Engineering and Technology, Amorha, Uttar Pradesh, India

S. K. Gupta
School of Management (UG), Dr. Vishwanath Karad MIT World Peace University, Pune, Maharashtra, India

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_39

1 Introduction

Investment can be defined as 'an asset acquired for the purpose of wealth building.' To understand the future of investment market, it is essential to predict the behavioural intentions of investors towards several types of investment. In past few years, two unique phenomena, related to investment behaviours, have drawn our attention:

i. rise in women investors
ii. significant growth in cryptocurrency investment
iii. increasing proportion of young investors.

Although coronavirus disease (COVID-19) outbreak has resulted into several harmful outcomes, however, it has also resulted into a significant rise in female investors [11]. Based on this premise, we believe that in future also we will observe a significant growth in female investors. Among several types of investment, the cryptocurrency has gained a quick edge over others in terms of rate of adoption.

Cryptocurrency is a blockchain technology-based digital currency which can be mined or purchased from cryptocurrency exchanges and is nearly impossible to counterfeit [8]. The global adoption rate of cryptocurrency has increased from 11.2% in October 2021 to 15.5% in December 2021 [23] with India ranking second in the world in the list of 20 countries with highest cryptocurrency adoption rate [18]. Increasing adoption rate of blockchain technology and significant role of digital currency in industrial revolution 4.0 promises a bright future of cryptocurrency [3]. In order to deal in cryptocurrency, a technical know-how of it is essential. Since, it requires a thorough understanding of disruptive technologies like blockchain, we believe that Generation *Z* people will be more comfortable in dealing with it. As, with a huge access to information, Generation *Z* individuals are more tech-savvy than any other generation [22].

The contribution of Generation *Z* individuals in the world population is around 27% with majority of them residing in India and China [22]. Today, a major chunk of Generation *Z* citizens of India has reached to college level and soon they will start earning. Hence, we found them a suitable population of interest for this study.

Based on above premise, we believe that it is high time to understand the factors influencing behavioural intention (BI) of Generation *Z* individuals to invest in cryptocurrency. To understand the same, we run a literature search on Scopus database and found that the theme is drawing the attention of researchers. However, most of the studies have not segregated the population or have focused on millennials only [2, 9, 13, 15, 16, 20, 21, 25]. Only one published study has considered Generation *Z* population [4].

In continuation to the above discussion and keeping in view the substantial growth in women investors after COVID-19 outbreak, we decided to draw our sample from Generation *Z* females who are pursuing B.Tech. or equivalent degree courses. Hence, our study becomes unique, as none of the published study has addressed the BI of Generation *Z* females.

Being a potential investor of cryptocurrency, which usually requires more funds than investing in any other stock or securities, one must be able to earn a handsome salary after their college education. To address this issue, we have targeted students at Institute of National Importance (INI) (like IIT and IIIT), where average salary package of UG programme is Rs. 6,000,000 INR.

Therefore, the current study aims at identifying and prioritizing the reasons behind the behavioural intentions to invest in cryptocurrency among Generation *Z* female of India, with special reference to students pursuing B.Tech. or equivalent course from Institute of National Importance.

2 Theoretical Framework

Exponential growth in investment in cryptocurrency reveals that it has become an integral part of modern digital world, and in near future, it will be on the portfolio of the majority of the investors [9]. Hence, it is high time to understand the factors influencing the BI of potential (Generation *Z*) investors towards cryptocurrency. In order to meet the research objectives, we have considered three most popular and renowned theories—technology acceptance model (TAM) proposed by Davis [6], innovation diffusion theory (IDT) propounded by Rogers [19] and unified theory of acceptance and use of technology (UTAUT) given by Venkatesh et al. [24].

TAM was originally developed to determine the user acceptance of computers with the help of perceived usefulness (PU) and perceived ease of use (PEoU). PU refers to a strong feeling in an individual regarding enhancing the performance by using an emerging technology [6]. PU is one of the factors which has a significant effect on the adoption of new technology [9]. The extent to which a person believe that using of a new technology will require less or no effort is referred to as PEoU [6]. PU is more significant tool to predict the adoption of new technology, whereas PEoU has been found helpful in reducing cognitive effort in using a technology and hence adopting it [4].

With the help of constructs like 'compatibility' and 'observability', IDT explains how, why and with what rate a new technology spreads. Compatibility refers to the degree to which a technological innovation is considered to be coherent with the end-users needs, potential usage, ease of use and comfort; whereas observability refers to the extent to which the outcomes of newly invented technology are evident to other people [14]. Empirical researches prove that compatibility and observability have the potential to explain PEOU [4, 14]. Based on above discussion, the following hypothesis are proposed:

H_1: Observability has a significant impact on perceived ease of use among Generation*Z* females.

H_2: Compatibility has significant impact on perceived ease of use among Generation *Z* females.

H_3: Perceived ease of use influence behavioural intention of Generation Z females towards investment in cryptocurrency.
H_4: Perceived usefulenss positively influence behavioural intention of Generation Z females towards investment in cryptocurrency.

Among several theories and models available to understand technology acceptance, the UTAUT has the potential to explain around 70% of the variance in BI to utilize a technology [24]. The revised version of UTAUT comprises the following constructs—hedonic motivation (the sense of enjoyment or happiness obtained by using new technology) and price value (cognitive trade-off between the price of the technology and the derived benefits from it) [24]. To understand the cryptocurrency adoption behaviour Abbasi et al. [1] extended UTAUT model by adding two more constructs—trust and facilitating conditions—to the original model. Trust is a subjective disposition, which refers to the degree to which an individual believes that his investment in cryptocurrency is safe and secure. Facilitating conditions refer to the perception of an individual regarding the availability of adequate infrastructure for investment in cryptocurrency. Therefore, based on the above arguments, the following set of hypothesis are proposed:

H_5: Trust influence behavioural intention of Generation Z female towards investment in cryptocurrency.
H_6: Price value influence behavioural intention of Generation Z female towards investment in cryptocurrency.
H_7: Hedonic motivation influences behavioural intention of Generation Z female towards investment in cryptocurrency.
H_8: Facilitating conditions influence behavioural intention of Generation Z females towards investment in cryptocurrency.

Based on the above discussion, the following theory model has been proposed for the study (Fig. 1).

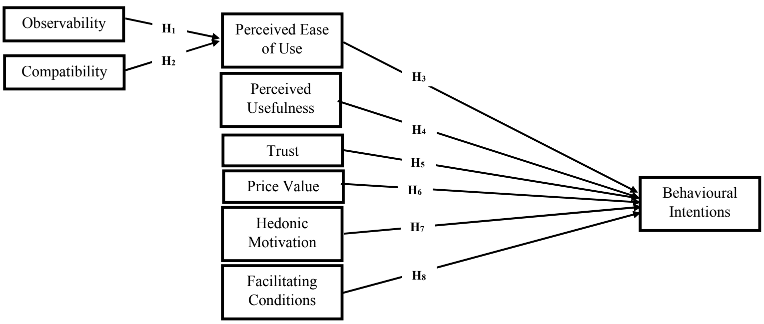


Fig. 1 Proposed research framework

3 Methodology

This section of the paper addresses the details of the population of interest from which data is collected and the research instrument used for data collection. Hence, it has been divided into two sections—population and sample, and measures.

3.1 Population and Sample

The data was collected from the female students pursuing B.Tech. or equivalent degree from INI, India. Due to COVID-19 restrictions, we contacted students online via e-mail, Cisco Webex, MS Teams and Google Meet to make them understand the research objectives, and after their due consent, the structured questionnaire in Google form was shared with them. Purposive sampling was used to establish the contact with few respondents and then through snowball sampling further respondents were approached. Total 311 students were contacted, out of which only 276 participated in the survey.

3.2 Measures

In order to collect the empirical data for this research, we have developed a structured questionnaire, which has two sections—*A* and *B*. Section *A* of the questionnaire comprises demographic details of the respondents like name, age, affiliating university/institution, whereas Section *B* comprises different scales used for measuring the study constructs. We have adopted validated scales for measuring the study constructs. The details of scale items and their source are given in Table 1.

Table 1 List of constructs and no. of measurement item

Construct	No. of items	References
Observability (OB)	4 items coded as OB1, OB2, OB3 and OB4	[4]
Compatibility (COM)	3 items coded as COM1, COM2 and COM3	[4]
Perceived usefulness (PU)	3 items coded as PU1, PU2 and PU3	[4, 6]
Perceived ease of use (PEOU)	3 items coded as PEOU1, PEOU2 and PEOU3	[4, 6]
Trust (TR)	3 items coded as TR1, TR2 and TR3	[1]
Price value (PV)	3 items coded as PV1, PV2 and PV3	[1, 24]
Hedonic motivation (HM)	3 items coded as HM1, HM2 and HM3	[1, 24]
Facilitating conditions (FC)	3 items coded as FC1, FC2 and FC3	[1]
Behavioural intention (BI)	2 items coded as BI1 and BI2	[6]

For measuring the study items, Likert scale varying from 1 (= strongly disagree) to 5 (= strongly agree) was used.

4 Data Analysis

4.1 Common Method Bias

Common method bias can be a serious concern in our study as the responses were collected from single category of respondents [5]. Harman's one factor test was conducted to test the data set for common method bias, the results reveal that single factor has explained a total variance of 19.99%, which was less then threshold 50% [5]. Hence, the data set is free from common method bias.

4.2 Measurement Model

All the items of research model were tested for item total correlation and three items OB3, OB4 and PEoU2 were found showing poor correlation values, hence removed from further analysis [10]. Confirmatory factor analysis (CFA) was then applied on remaining items and obtained factor loadings were used for computing composite reliability (CR) and average variance extracted (AVE) (result shown in Table 2). The model fit indices of CFA (CMIN/DF = 1.385, RMSEA = 0.037, CFI = 0.951, NFI = 0.849, TLI = 0.937) were in acceptable limit [12].

The obtained value of CR (> 0.70) and AVE (> 0.50) was above their threshold values; hence, the research model displays the convergent validity [7, 17]. Also, the square root of all obtained AVE was greater than the intercorrelation of constructs, ensuring the discriminant validity of the scale [10].

4.3 Structural Model

The model fit indices (CMIN/DF = 2.741, RMSEA = 0.060, CFI = 943, NFI = 0.859, TLI = 0.918) obtained as a result of structural equation modelling (SEM), performed on AMOS-20, were in acceptable limits [12]. The results of structural model are discussed in the next section.

Table 2 Measurement model result

Construct	Item code	Factor loading	CR	AVE
BI	BI1	0.84	0.865	0.689
	BI2	0.602		
OB	OB1	0.737	0.849	0.659
	OB2	0.659		
PEOU	PEOU1	0.762	0.85	0.662
	PEOU3	0.637		
COM	COM1	0.632	0.756	0.51
	COM2	0.726		
	COM3	0.777		
TR	TR1	0.731	0.788	0.555
	TR2	0.702		
	TR3	0.798		
PV	PV1	0.844	0.779	0.543
	PV2	0.709		
	PV3	0.643		
HM	HM1	0.845	0.886	0.722
	HM2	0.916		
	HM3	0.783		
FC	FC1	0.745	0.754	0.506
	FC2	0.729		
	FC3	0.657		
PU	PU1	0.799	0.797	0.568
	PU2	0.772		
	PU3	0.686		

**CR* Composite Reliability, *AVE* Average Variance Explained

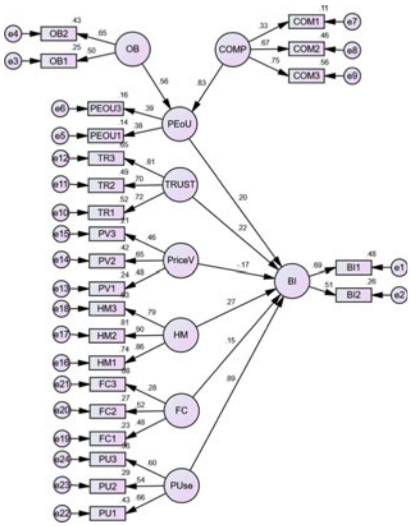
5 Discussion

This study contributed to the domain of knowledge by identifying the factors determining BI to adopted cryptocurrency among Generation *Z* females of India. Findings of SEM (shown in Fig. 2 and Table 3) reveal that, PU has the highest impact on behavioural intentions to purchase cryptocurrency, followed by HM and trust. Although, the impact of PEoU on BI is positive but obtained loading value is relatively low (0.197). However, the *p*-value supports the poor but significant impact of PEOU on BI. Since Generation *Z* individuals are more tech-savvy than any other generation people, hence, for them, the ease of using a technology does not matter much, and this could be the possible reason behind the poor loading of PEOU on BI. Antecedents of PEOU, OB and COM have significantly determined the value of

PEOU, hence, the findings are in accordance with Bhardwaj and Deka [4]. Results of hypothesis testing, given in Table 3, reveals that H_1, H_2, H_3, H_4, H_6, H_7 and H_8 are accepted, whereas test results failed to accept H_5.

PV shows a negative impact on BI, the possible reason could be most of the individuals belonging to Generation Z have not started earning yet. Hopefully with some other population, like Millennials or a mixed population, we can expect a significant impact of PV on BI.

The findings of the study prove that the behaviour of Generation Z individuals, towards investment in cryptocurrency, is driven by the confluence of personal and technical factors. This generation is more concerned about the enjoyment achieved



*OB- Observability, COMP-Compatibility, PEoU-Perceived ease of use, PriceV- Price Value, HM-Hedonic Motivation, FC-Facilitating conditions, PUse – Perceived usefulness, BI-Behavioural intentions

Fig. 2 Path diagram

Table 3 Result of hypothesis testing

Hypothesis	Relationship			Loading	S.E	t-value	p-value	Result
$H1$	PEOU	←	OB	0.561	0.152	2.916	0.004	Supported
$H2$	PEOU	←	COM	0.828	0.269	3.455	0.000	Supported
$H3$	BI	←	PEOU	0.197	0.105	2.284	0.022	Supported
$H4$	BI	←	TRUST	0.221	0.052	2.991	0.003	Supported
$H5$	BI	←	PV	-0.17	0.093	-1.894	0.058	Insignificant
$H6$	BI	←	HM	0.27	0.036	3.895	0.000	Supported
$H7$	BI	←	FC	0.146	0.14	1.376	0.009	Supported
$H8$	BI	←	PU	0.888	0.107	7.524	0.000	Supported

in investment in cryptocurrency rather than the price value attained through it. They believe in self-performance enhancement due to cryptocurrency adoption rather than the ease of using it.

6 Limitations of the Study and Future Research Directions

Even though the current study is the first empirical attempt to establish a connect between three different theories to understand the investment behaviour of Generation *Z* female, still it fails to address few aspects, like:

i. Only female participants belonging to Generation *Z* and pursuing degree course of INI were considered in this study. However, this specific segment might lead to *social desirability bias*, hence, we suggest that in future, a mixed population of male and female both along with students of public and private institutions can be considered to develop a better and in-depth understanding of the study constructs and their association.
ii. Since, the study sample was from India, hence, the study faced a major limitation in terms of *generalization of the findings*. The findings of the study may not be applicable to Generation *Z* population of other nations. Hence, it is recommended that in future researchers can consider a cross-national survey to validate the findings of this research.
iii. The epicentre of the study was BI to adopt cryptocurrency, however, BI towards hesitation in cryptocurrency adoption was not considered in this study. Hesitate to invest in cryptocurrency can be a potential theme and future researchers can think about developing and validating a scale to understand and measure *crypto-hesitancy behaviour*.

7 Conclusion

With an aim to understand the BI of Generation *Z* female towards investment in cryptocurrency, the current study has significantly contributed towards the domain of knowledge by combined three established theories—TAM, IDT and UTAUT. Earlies BI studies had focused on either single theory or a combination of two, like TAM and IDT combined by Bhardwaj and Deka [4].

This study is unique in two aspects—i. it addresses the acceptance and utilization of cryptocurrency along with the reasons and rate with which it will spread among Generation *Z* female, and hence, the study findings can be useful for position the product (or service) provided by the cryptocurrency trading companies in India; ii. it is the first attempt to predict the BI towards cryptocurrency investment through a combined effort of PEoU, PU, Trust, PV, HM and FC.

Findings of the study reveal that, being ‘digital native’ and ‘tech-savvy’, the Generation *Z* individuals enjoy and find pleasure in cryptocurrency adoption, also they believe that their performance will be enhanced by adopting cryptocurrency. The findings of the study can be helpful for consultants and organizations, dealing in cryptocurrencies to design promotional strategies. The potential consumers of digital currencies can also use the findings to strengthen their attitudes and subjective norms before making any investment decisions.

References

1. Abbasi GA, Tiew LY, Tang J, Goh YN, Thurasamy R (2021) The adoption of cryptocurrency as a disruptive force: deep learning-based dual stage structural equation modelling and artificial neural network analysis. PLOS ONE 16(3). https://doi.org/10.1371/journal.pone.0247582
2. Anser MK, Zaigham GHK, Imran Rasheed M, Pitafi AH, Iqbal J, Luqman A (2020) Social media usage and individuals' intentions toward adopting bitcoin: the role of the theory of planned behavior and perceived risk. Int J Commun Syst 33(17):1–16. https://doi.org/10.1002/dac.4590
3. Arli D, van Esch P, Bakpayev M, Laurence A (2021) Do consumers really trust cryptocurrencies? Mark Intell Plan 39(1):74–90. https://doi.org/10.1108/MIP-01-2020-0036
4. Bhardwaj S, Deka S (2021) Behavioural intention towards investment in cryptocurrency: an integration of Rogers' diffusion of innovation theory and the technology acceptance model. Forum Sci Oeconomia 4(9):137–159. https://doi.org/10.23762/FSO_VOL9_NO4_7
5. Cabral C, Dhar LR (2019) Green competencies: construct development and measurement validation. J Clean Prod 235:887–900. https://doi.org/10.1016/j.jclepro.2019.07.014
6. Davis FD (1989) Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Q Manag Inform Syst 13(3):319–339. https://doi.org/10.2307/249008
7. Fornell C, Larcker DF (1981) Evaluating structural equation models with unobservable variables and measurement error. J Mark Res 18(1):39. https://doi.org/10.2307/3151312
8. Frankenfield J (2011) Cryptocurrency. https://www.investopedia.com/terms/c/cryptocurrency.asp
9. Gupta S, Gupta S, Mathew M, Sama HR (2021) Prioritizing intentions behind investment in cryptocurrency: a fuzzy analytical framework. J Econ Stud 48(8):1442–1459. https://doi.org/10.1108/JES-06-2020-0285
10. Hair JF, Black WC, Babin BJ, Anderson RE (2014) Multivariate data analysis. Pearson Education. https://doi.org/10.1002/9781118895238.ch8
11. HT Brand Studio (2021) The rise of female investors in the post-pandemic era-Hindustan Times. https://www.hindustantimes.com/business/the-rise-of-female-investors-in-the-post-pandemic-era-101637933271258.html
12. Hu LT, Bentler PM (1999) Cutoff criteria for fit indexes in covariance structure analysis: conventional criteria versus new alternatives. Struct Equ Model 6(1):1–55. https://doi.org/10.1080/10705519909540118
13. Kim M (2021) A psychological approach to Bitcoin usage behavior in the era of COVID-19: focusing on the role of attitudes toward money. J Retail Consum Serv 62. https://doi.org/10.1016/j.jretconser.2021.102606
14. Lou ATF, Li EY (2017) Integrating innovation diffusion theory and the technology acceptance model: the adoption of blockchain technology from business managers' perspective. In: Proceedings of the international conference on electronic business (ICEB), pp 299–302
15. Mnif E, Lacombe I, Jarboui A (2021) Users' perception toward Bitcoin Green with big data analytics. Soc Bus Rev 16(4):592–615. https://doi.org/10.1108/SBR-02-2021-0016
16. Nadeem MA, Liu Z, Pitafi AH, Younis A, Xu Y (2020) Investigating the repurchase intention of Bitcoin: empirical evidence from China. Data Technol Appl 54(5):625–642. https://doi.org/10.1108/DTA-10-2019-0182

17. Nunnally JC, Bernstein IH (1994) Psychometric theory, 3rd edn. McGraw-Hill, Inc. https://books.google.com/books?id=_6R_f3G58JsC&pgis=1
18. Ramasubramania S (2021) India ranks second in Chainalysis's 2021 global crypto adoption index. The Hindu. https://www.thehindu.com/sci-tech/technology/india-ranks-second-in-chainalysiss-2021-global-crypto-adoption-index/article36057776.ece
19. Rogers EM (1983) Diffusion of innovations, 3rd edn. The Free Press. https://doi.org/10.4324/9780203710753-35
20. Soomro BA, Shah N, Abdelwahed NAA (2022) Intention to adopt cryptocurrency: a robust contribution of trust and the theory of planned behavior. J Econom Adm Sci. https://doi.org/10.1108/jeas-10-2021-0204
21. Ter Ji-Xi J, Salamzadeh Y, Teoh AP (2021) Behavioral intention to use cryptocurrency in Malaysia: an empirical study. Bottom Line 34(2):170–197. https://doi.org/10.1108/BL-08-2020-0053
22. Thangavel P, Pathak P, Chandra B (2021) Millennials and Generation Z: a generational cohort analysis of Indian consumers. Benchmarking Int J 28(7):2157–2177. https://doi.org/10.1108/BIJ-01-2020-0050
23. TOI (2022) Crypto adoption trends in South Africa and world in 2022: Finder's cryptocurrency adoption index. Times of India. https://timesofindia.indiatimes.com/business/cryptocurrency/blockchain/crypto-adoption-trends-in-south-africa-and-world-in-2022-finders-cryptocurrency-adoption-index/articleshow/88994672.cms
24. Venkatesh V, Thong JYL, Xu X (2012) Consumer acceptance and use of information technology: extending the unified theory of acceptance and use of technology. MIS Q 36(1):157–178
25. Yeong YC, Kalid KS, Sugathan SK (2019) Cryptocurrency acceptance: a case of Malaysia. Int J Eng Adv Technol 8(5C):28–38. https://doi.org/10.35940/ijeat.E1004.0585C19

Freight Cost Prediction Using Machine Learning Algorithms

Pranav Kulkarni, Ishan Gala, and Aniket Nargundkar

Abstract With the advent of the digital age, it is certainly noticeable that the world has indeed become a smaller place and better connected due to progressing technology and innovations that occur every day. In the virtual world, the distance factor has certainly been negated due to the Internet phenomenon, however in the physical world, distance plays an extremely important role in any kind of business decisions that a company might need to take, and the cost related to the distance is also one of the most prominent factors that influence any decision. This paper focuses on predicting the freight cost of a cargo, which is dependent on a variety of variables such as location, the weight of the cargo, etc. Four widely applied ML models are adopted, viz. K nearest neighbors regressor, Random Forest regressor, XGBoost regressor, and LightGBM regressor. Using the most prominent and highly influential features out of all, the comparative analysis of the machine learning models is presented.

Keywords Freight cost · XGBoost · Random forest · Machine learning · K nearest neighbor · LightGBM

1 Introduction

The expense of carrying a certain cargo from one point to another is known as freight cost. The cost is determined by the type of cargo, the mode of transportation (truck, ship, train, or aircraft), the weight of the cargo, and the distance to the delivery location. Due to the surge in demand for goods as consumers prefer to spend their

P. Kulkarni (✉) · I. Gala
Marathwada Mitramandal's College of Engineering, Pune 411052, India
e-mail: pranav.kulkarni1337@gmail.com

I. Gala
e-mail: ishangala16@gmail.com

A. Nargundkar
Symbiosis International (Deemed University), Symbiosis Institute of Technology, Lavale, Pune 412115, India
e-mail: aniket.nargundkar@sitpune.edu.in

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_40

money on goods rather than services during pandemic lockdowns and restrictions, there is a rise in freight costs. As a result, businesses must forecast freight costs ahead of time and make transportation decisions accordingly. The cost charged to a shipper (the person or company who provides products for shipment) or consignee (the person or company to whom commodities are transported) for the transportation of goods is influenced by a number of factors. Mode of transportation, weight, size, distance, collection and delivery places, and the actual products being shipped are all aspects to consider. All of these factors play separate but interconnected roles in determining the cost at which the freight will be transported.

In this paper, the problem is solved using four different machine learning algorithms, viz. K nearest neighbors regressor, Random Forest regressor, XGBoost regressor, and LightGBM regressor. These four are different regression-based algorithms. We have compared the results obtained from all four algorithms [1, 2]. Recently, various contemporary AI-based techniques are applied for freight cost prediction [3–5].

Various machine learning approaches have been applied before, at different airports and seaports where there is a huge demand for forecasting data related to freights. Milenković et al. [6] used time series data from January 2010 to December 2016 and combined ANN with simulated annealing and genetic algorithms to predict container flow for the port of Barcelona. Sulistyowati et al. [7] used time series data to develop hybrid models that combine statistical and ML (ANN and SVM) models. Zhao et al. [8] developed a hybrid heuristic algorithm that incorporates ANN into Monte-Carlo simulation and a genetic algorithm to select sea-rail container routes with the goal of minimizing total transportation costs. Moscoso-López et al. [9] created a two-stage Bayesian regularization neural network (BRNN) to provide a reliable prediction of fresh freight weight on Ro-Ro (roll-on/roll-off) vessels transportation. Most of the approaches to freight cost analysis and prediction are generally based on different neural network architectures, however, this research has been conducted to find a different approach to the same using traditional machine learning and ensemble learning techniques.

The remainder of the paper is as follows: Sect. 2 describes the methodology adopted in the current work. Section 3 provides the results obtained and comparison of solutions. A conclusion along with future direction is discussed in Sect. 4.

2 Methodology

The methodology followed in this paper includes data preprocessing and applying machine learning algorithms to the dataset. We have used the SCMS Delivery History Dataset which contains transactional information from the distribution of HIV/AIDS supplies to PEPFAR-supported countries. The data is collected by USAID, which collects, manages, and publishes the SCMS procurement data on an almost quarterly basis.

Data Preprocessing

We started with basic data exploration and data cleaning operations such as:

1. Converting the columns into respective data types.
2. Checking for null values and calculating the percentage of null values.
3. Visualize the null values for better clarity.
4. Understanding the numerical features.
5. Checking for outliers in the dataset.
6. Handling missing values and NaN values.
7. Dropping categorical variables with too many classes.
8. Converting categorical variables into ordinal variables using label encoding.
9. Useful feature extraction.
10. Dropping columns that have many missing values and are of least priority.
11. Handling outliers using inter-quartile range.
12. Splitting the dataset into train set and test set.

Visualization of important features of the dataset before data preprocessing

See Fig. 1.

Visualization of important features of the dataset after data preprocessing

Figures 1 and 2 highlight the box plots (quartile ranges) of the data before and after the application of preprocessing techniques. It is evident that the data prior to preprocessing contains a large number of outliers, which affect the accuracy of the

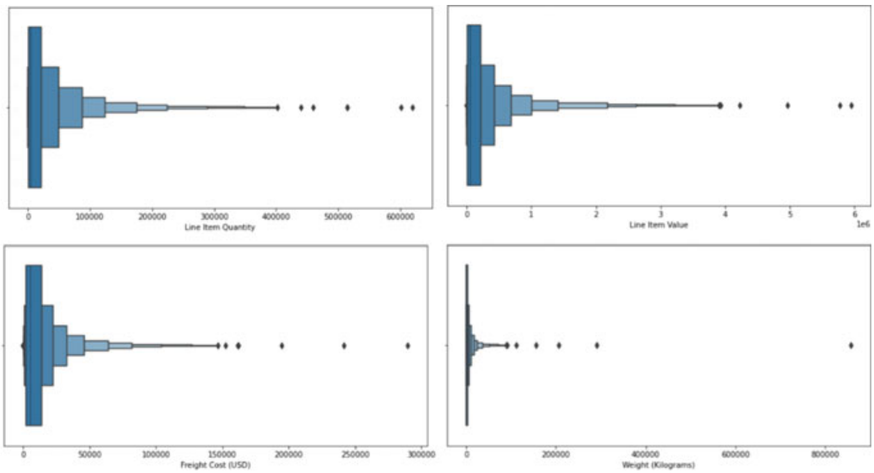


Fig. 1 Unprocessed data box plot

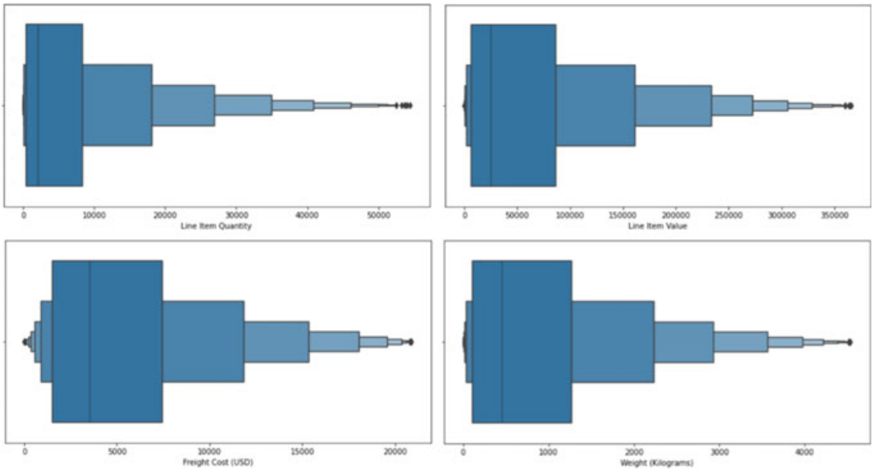


Fig. 2 Processed data box plot

model and hence need to be dealt with, which is what we have done as a part of the preprocessing.

Figure 3 describes the *F*-scores of each feature present in the dataset relative to the cost of the actual freight. The *F*-score reflects the discriminative strength of each characteristic on its own when related to the actual feature that is to be predicted.

Figure 4 explains the correlational coefficients of every feature relative to the freight cost, expressed as a floating-point value between 1 and − 1, where 1 indicates absolute relation and − 1 indicates no relation whatsoever. After application of data

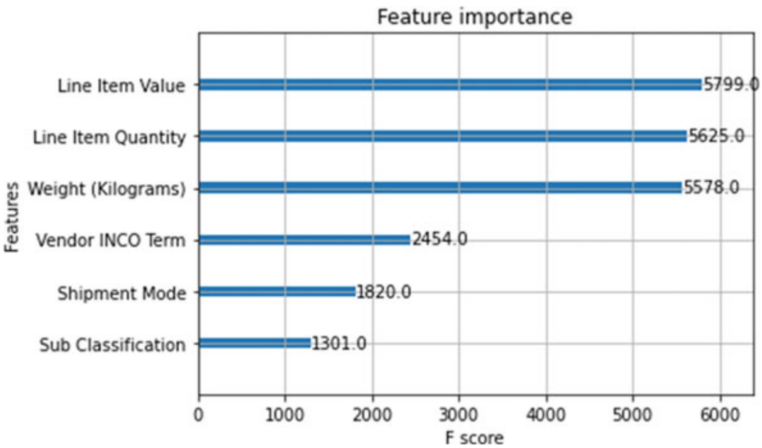


Fig. 3 F-scores of each feature relative to cost

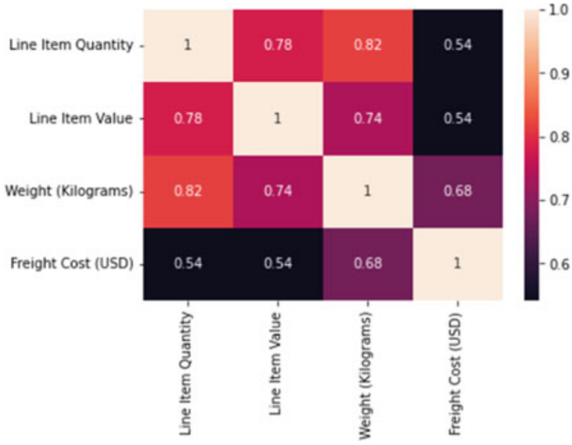


Fig. 4 Correlation heatmap of features

preprocessing techniques on the dataset, the correlational values obtained have been used for feature selection.

Training the Models

In this step, we train the models using the training data. Four different machine learning algorithms were used, which are described below.

1. **K Nearest Neighbors Regressor**

KNN regression is a non-parametric method that uses averaging data in the same neighborhood to approximate the relationship between independent variables and continuous outcomes [10].

2. **Random Forest Regressor**

Random Forest regression is a supervised learning algorithm that performs regression using the ensemble learning method. The ensemble learning method combines predictions from multiple machine learning algorithms to produce a more accurate prediction than a single model [11].

In Random Forests, each tree in the ensemble is constructed using a sample selected from the training set with replacement. Furthermore, the optimal split is selected from all input features or a random subset of size max features when dividing each node during tree construction.

The goal of these two sources of randomness is to reduce the forest estimator's variance. Individual decision trees do, in fact, have a lot of diversity and are prone to overfitting. Forests with injected randomness produce decision trees with decoupling prediction errors. Some mistakes can be eliminated by averaging their predictions. By merging distinct trees, Random Forests reduce volatility, sometimes at the expense of a modest increase in bias. In practice, the variance reduction is often significant, hence yielding an overall better model.

3. **XGBoost Regressor**

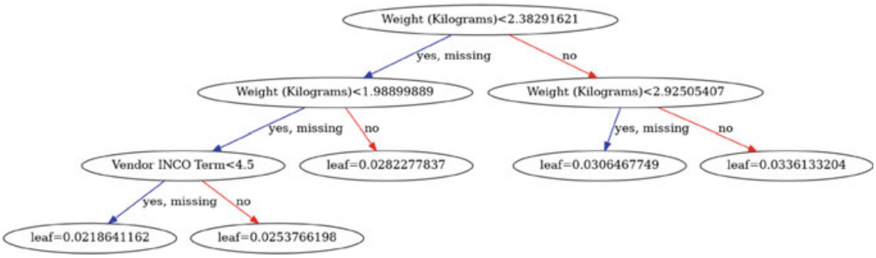


Fig. 5 XGBoost plot tree

Extreme gradient boosting (XGBoost) is an efficient and effective implementation of the gradient boosting algorithm. Predicting a numerical value, such as a dollar amount or a height, is the goal of regression predictive modeling problems. For regression predictive modeling, XGBoost can be used directly [12].

XGBoost and gradient boosting machines (GBMs) are both ensemble tree approaches that use the gradient descent architecture to boost weak learners (CARTs in general). XGBoost, on the other hand, enhances the fundamental GBM framework through system optimization and algorithmic improvements.

Figure 5 is the actual representation of the tree plot of the XGBoost algorithm for the dataset under consideration.

4. **LightGBM Regressor**

LightGBM is a decision tree-based gradient boosting framework that enhances model efficiency while minimizing memory utilization. It uses two innovative strategies to overcome the limitations of the histogram-based algorithm used in all gradient boosting decision tree (GBDT) frameworks: gradient-based one side sampling and exclusive feature bundling (EFB). The GOSS and EFB approaches combine to generate the LightGBM algorithm's features [13].

3 Results

For the evaluation of the machine learning algorithms on the freight cost prediction analysis, the following metrics were considered:

Accuracy—The accuracy of the algorithm under testing was obtained by running it on the validation data in the dataset, and then the accuracy on the validation dataset was considered.

MAE—MAE stands for ‘Mean Absolute Error’, a common metric used for evaluation of machine learning algorithms, and the closer it is to zero, the better the algorithm is at predicting unknown quantities.

The comparative analysis of the four algorithms used is given below:

1. **KNN Regression Algorithm**

For the KNN regression algorithm, to find the best configuration of the model, some hyperparameters (in the case of this algorithm, the number of neighbors) are tried out from a list of options, i.e., a range of 1–50 neighbors is considered and the model is constructed and evaluated for every one of those configurations, and the best configuration is selected, on which a maximum accuracy of 63.025% on the validation dataset for a configuration of 18 neighbors (i.e., the algorithm is run by tuning the value of neighbors to 18).

The KNN regression algorithm gives a Mean Absolute Error of 0.209, which is comparably quite decent and is better than the decision tree algorithm.

2. **Random Forest Regression Algorithm**

For the Random Forest regression algorithm, a similar configuration searching method is implemented with the help of the GridSearchCV module from Scikit-Learn, which enables one to get the optimized hyperparameter values for any estimator by taking the range of hyperparameter values as input and comparing the output to the actual data and then providing the optimal hyperparameters.

When this algorithm is evaluated against the validation data, an accuracy measure of 63.780% is obtained, which is better than the KNN algorithm. Also, the MAE obtained for the Random Forest regression model is 0.204, which is also lesser than the KNN algorithm, and hence, it is evident that the Random Forest model performs better on this data in comparison with the KNN model.

3. **Extreme Gradient Boosting Algorithm (XGBoost)**

XGBoost is considered conventionally one of the best algorithms for tabular data regression problems, which is why it was taken under consideration for the comparative analysis.

XGBoost also has some tunable hyperparameters, and after tweaking out the numbers, an optimal configuration of 1000 n-estimators, a maximum depth of 5, and a learning rate of 0.01 provided the optimal results, in which the accuracy obtained was 63.9238%, better than both the KNN model and the Random Forest model and the Mean Absolute Error obtained was 0.2052, better than the KNN regression algorithm.

4. **Light Gradient Boosting Algorithm (LGBM)**

Light gradient boosting is another method commonly used for the evaluation of regression problems in machine learning and works similarly to XGBoost barring

Table 1 Mean absolute error and accuracy of algorithms

Models	MAE	Accuracy (%)
KNN	0.2090	63.025
RF	0.2048	63.780
XGBoost	0.2051	63.924
LGBM	0.2035	64.278

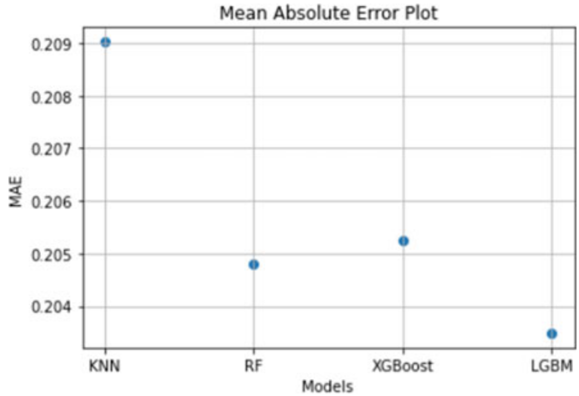


Fig. 6 Scatter plot of mean absolute error

some differences in the growth of the learning trees, XGBoost applies level-wise tree growth or horizontal growth, while LGBM uses leaf-wise tree growth or vertical growth.

LGBM has similar tunable hyperparameters as compared to XGBoost, and after similar tweaking, an optimal configuration of 1000 n-estimators, a maximum depth of 5, and a learning rate of 0.01 resulted in the optimal solution, in which the accuracy was 64.2781%, better than all the prior algorithms considered and the Mean Absolute Error obtained was 0.203, significantly better than all previous algorithms (Table 1).

Figure 6 shows the Mean Absolute Error of each model used. On the vertical axis, we have the absolute error values, and on the horizontal axis, we have the models used for the analysis.

4 Conclusions and Future Directions

Logistics play a critical role in today's complex and closed-loop supply chain because of the globalization and complex supply chain models. Freight cost is one of the most important financial indicators of supply chain efficiency. Forecasting freight costs ahead of time helps the decision maker to make optimal transportation decisions. Over the years, researchers have studied and applied various AI-based techniques for freight cost prediction. In the current work, two widely used and contemporary machine learning algorithms, viz. KNN and Random Forest regressors, are applied for the freight cost prediction. Furthermore, two ensemble learning techniques such as XGBoost and LGBM have also been applied. The results show that ensemble learning techniques outperform contemporary machine learning algorithms by 1.253% which has a significant impact at the scale of freight costs. It can be concluded that the best performance for the purpose of freight cost prediction using this data is certainly provided by LGBM followed by XGBoost. This is evident from the highest accuracy output by LGBM on the validation dataset as well as the lowest Mean Absolute Error as compared to the other algorithms. This demonstrates the applicability of contemporary machine learning models and ensemble learning techniques for freight cost prediction. In the near future, authors intend to apply stacked neural networks and perform feature engineering for complex real-world supply chain big data.

References

1. Murphy PR, Hall PK (1995) The relative importance of cost and service in freight transportation choice before and after deregulation: an update. Transp J 30–38
2. Sert E, Hedayatifar L, Rigg RA, Akhavan A, Buchel O, Saadi DE, Bar-Yam Y (2020) Freight time and cost optimization in complex logistics networks. Complexity
3. Ćalić J, Šelmić M, Macura D, Nikolić M (2019) Fuzzy logic application in green transport-prediction of freight train energy consumption. In: Proceedings of the 4th logistics international conference, Belgrade, Republic of Serbia, pp 35–44
4. Zhao Y, Ioannou PA, Dessouky MM (2018) Dynamic multimodal freight routing using a co-simulation optimization approach. IEEE Trans Intell Transp Syst 20(7):2657–2667
5. Wang CY, Zhu AD (2018) A novel GA-BP based bidding prediction algorithm for contract logistics of road freight transportation. In: Proceedings of 2018 international conference on education reform and management science (ERMS2018). Advanced Science and Industry Research Center: Science and Engineering Research Center, pp 6
6. Milenković M, Milosavljevic N, Bojović N, Val S (2017) Container flow forecasting through neural networks based on metaheuristics. Oper Res 1–33
7. Sulistyowati R, Kuswanto H, Astuti ET (2018) Hybrid forecasting model to predict air passenger and cargo in Indonesia. In: international conference on information and communications technology, pp 442–447
8. Zhao Y, Liu R, Zhang X, Whiteing A (2018) A chance-constrained stochastic approach to intermodal container routing problems. PLoS ONE 13(2):e0192275
9. Moscoso-López JA, Turias I, Jiménez-Come MJ, Ruiz-Aguilar JJ, Cerbán MDM (2019) A two-stage forecasting approach for short-term intermodal freight prediction. Int Trans Oper Res 26(2):642–666
10. Almanjahie IM, Aissiri KA, Laksaci A, Chikr Elmezouar Z (2020) The k nearest neighbors smoothing of the relative-error regression with functional regressor. Commun Stat Theory Methods 1–14
11. Cootes TF, Ionita MC, Lindner C, Sauer P (2012) Robust and accurate shape model fitting using random forest regression voting. In: European conference on computer vision. Springer, Heidelberg, pp 278–291
12. Chen T, He T, Benesty M, Khotilovich V, Tang Y, Cho H (2015) Xgboost: extreme gradient boosting. R package version 0.4–2 1(4):1–4
13. Kadiyala A, Kumar A (2018) Applications of python to evaluate the performance of decision tree-based boosting algorithms. Environ Prog Sustain Energy 37(2):618–623

An Ontological Framework for Risk Mitigation in Stock Market

Ambrish Kumar Mishra, Shweta Anand, Narayan C. Debnath, and Archana Patel

Abstract The stock market is used to be very popular from the eighteenth century, and the craze of the stock market is increasing with the high rate across the globe. Though people are putting their hard-earned money into the stock market to earn a high profit, yet few of them do research before investing or taking a trade, which results in a major loss, and sometimes whole capital vanishes out. The stock market depends on various factors like price, volume, decay value, etc. these factors can be easily identified by the indicators. In the literature, various indicators are available that can be used to mitigate the risk in the stock market while investing or trading. However, none indicator can provide the full assurance of protecting the capital when applied alone. Therefore, it is required either to develop a new indicator or to propose a framework that consists of a set of those indicators that can mitigate the investor's risk as compared to any of the available indicators. The aim of this paper is to propose an ontology-based framework that assimilates the features of three indicators, namely, Bollinger band, Fibonacci, and Heikin-Ashi, and map them with an ontology in the form of classes, properties, instances. The obtained results depict that the proposed framework achieves better results than available indicators.

Keywords Ontology · Risk mitigation · Indicators · Stock market · Bollinger band · Fibonacci · Heikin-Ashi

A. K. Mishra · S. Anand
School of Management, Gautam Buddha University, Greater Noida, Uttar Pradesh, India
e-mail: shweta.anand@gbu.ac.in

N. C. Debnath · A. Patel (✉)
Department of Software Engineering, Eastern International University, Binh Duong, Vietnam
e-mail: archana.patel@eiu.edu.vn

N. C. Debnath
e-mail: narayan.debnath@eiu.edu.vn

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_41

1 Introduction

A financial market is a place where financial assists like shares, bonds, currencies, commodities, derivatives, etc. are traded. The financial market is one of the strong pillars of the economy. By financial market capital is being accumulated that can be used for the production of goods and providing services. The financial market also provides a platform for cross-border fund transfer or investment among the countries. So, this is a win–win situation for the countries. Generally, the financial market has two types, namely, money market and capital market, whereas the capital market is divided into two categories, namely, primary market and secondary market. The stock market comes under the primary market. To meet the long-term goals and day-to-day expenses, companies need additional funds in a large amount. This fund either directly maybe collected from the public or may be taken as debt. To get the fund from the public, companies need to sell their shares to the public for the required amount. This buying and selling (trading) may be done at a particular place named the stock market. This buying and selling are done between promoters (owner of the company or who started the company) and investors (people who buy shares for better return).

At present stock market can be called a barometer of the economy as it is a business of business. According to [1] stock market reflects the general condition of the economy, i.e., atmosphere of business. The important functions of the stock market are:

- Trading of market securities: buying and selling of market securities
- Fund Collection: This motivates investors to save more and invest more
- Assets Creation: By mobilizing the fund to the economy, the stock market can provide capital to the business which can create assets for the economy in long run.
- Protecting the interest of investors: stock market is being regulated by certain rules and regulations. Thus, it ensures the safety of funds.
- Indicators of the financial health of companies: The stock market collects information from listed and non-listed companies of the stock market that may assist the investors in making their investment decisions.

The stock market can provide a high return in a short time. Investors can also get tax benefits along with ownership in the company, and their rights are also protected by the regulatory body. On the other side, due to volatility and risk, the exact prediction of return is difficult.

To mitigate the risk in stock market, various indicators have been developed by many analysts and researchers. However, none indicator (alone) can provide the full assurance of protecting the capital. Therefore, it is required either to develop a new indicator or propose a framework that consists a set of those indicators that can mitigate the investors' risk as compared to any of the available indicator. The aim of this paper is to propose an ontology-based framework for the risk mitigation in stock market. The ontology is used for the storage and access of highly volatile and large data [2] on real-time basis by interconnecting all the stock markets across the globe. Ontology is a semantic model used for knowledge representation, and it consists set of classes, properties, instances, the relationship between the classes, and axioms. Nowadays, all areas are using ontologies for the representation of data and inferencing over the stored data. In this paper, we use ontology for two purposes, namely, storage of data that comes from the different sources and stored the extracted features of the three indicators (Bollinger bands, Fibonacci retracement, and Heikin-Ashi). The rest of the paper is organized as follows: Sect. 2 provides a consolidated overview of the existing indicators that are used to mitigate the risk. Section 3 shows the proposed framework and algorithm. Section 4 focuses on the results and discussion. The Sect. 5 concludes this paper.

2 Literature

Investors invest the money in the share market based on various parameters like fundamental analysis, technical analysis, and price action strategy, as well as different indicators. The financial health of any company is observed by fundamental analysis, which includes the balance sheet, profit and loss statement, and financial statement. The market trend and future projection may be predicated by technical analysis. The characteristic of security price movement is shown by price action. The risk can be mitigated to the minimum level if the investor catches price action at an early stage by the studies of charts. Generally, two types of charts are used for the analysis, named line chart and bar chart (candle chart). The indicator indicates the anticipated future moves that are often used for the day-to-day decision on entry or exit from any stock. For example, if the relative strength indicator (RSI) [3] of any stock is less than or equal to 25%, it indicates the buying opportunity as the stock price is very low and our risk margin increases, whereas if the RSI of any stock is greater than or equal to 75%, it indicates the selling opportunity. However, the indications of RSI are not perfect for every time. Therefore, many indicators have been available in the literature that helps in the mitigation of the risk.

Lauguico et al. [4] have proposed a fuzzy logic-based trading algorithm for the stock market by using Bollinger bands. The proposed algorithm undergoes some certain trading strategies like, technical indicators are used to trigger the strength of signals (buy, hold, and sell), stock price data shows the opening and closing prices that are used by Bollinger bands, the membership functions were classified based on the input parameters used by the traders. The system is implemented using MATLAB and NI LabVIEW. Sethi et al. [5] have predicted the future of the current market trend by using Fibonacci retracement. They stated that Fibonacci retracement offers the current time frame of entry and exit points in 70% of the cases through the Golden ratio that is intended to offer a good structural design of the stock market. The hypothesis of this indicator is based on the technical analysis to forecast the stock prices in the near future. According to them, Fibonacci retracement can be calculated either manually or using automated tools by applying on the two points (high and low) on the graph. To achieve the better reliability, this indicator is applied in conjunction with other fundamental tools to predict both points (the entry and exit points) in the stock market. Madbouly et al. [6] have proposed a model that accurately predict the stock trend. This model combines the cloud model, fuzzy time series and Heikin-Ashi candlestick and predict Japanese stock market trend. The cloud model handles the ambiguity and uncertainty in the Japanese candlestick definitions (called qualitative information) and actual stock prices (quantitative data). The model constructed dynamic weighted fuzzy logical relationships based on the membership functions that aims to predict the next open and close prices of the stock as well as the high and low values. Heikin-Ashi Candlestick is constructed by calculating averages of the previous and current period prices. They have provided imperial evaluation that state the proposed model has high forecasting accuracy.

Hassen et al. [7] have proposed an application of cloud model qualitative forecasting for stock market trends. They have used extended fuzzy sets because the existing methods based on fuzzy reasoning techniques can not describe the data comprehensively. The proposed approach followed the cloud model with fuzzy time series and Heikin-Ashi. The experimental results proved the feasibility and high forecasting accuracy of the model. Trivedi and Kyal [8] have provided an overview of effective trading in financial markets using technical analysis. They have covered the following topics: introduction to technical analysis, basic principles of technical analysis, Classical reversal and continuation patterns, Candlesticks patterns and their use in trading strategies (Reversal patterns, Continuation patterns), Moving averages and their use for trading strategies, Momentum indicators and stochastics, Volatility and volume indicators, Elliott Wave Principles, Time cycles, Introduction to backtesting and algorithmic trading. Sundarakamatchi and Gajanand [9] have proposed a recommendation engine for stock market trading based on technical analysis and strategies. This recommendation engine is proposed with an aim to shift the way one trades by removing some of the biases inherent in the manual process. The proposed model consists of three stages: (a) obtain input data: this stage has three tasks, namely, fetch price data of stocks for the last n days, determine support level and resistance level for the stocks, obtain input from the decision-maker for a buffer around support and resistance levels (b) analyze the data: categorize the stocks considering decision-maker preferences, compare current prices of the stocks with their support and resistance levels, set the conditions for inferences on the buy (or) sell, generate a list of stocks for consideration, and (c) display the results: publish results in a consumable format. The proposed recommendation engine has encouraging results. Chen [10] has provided a book entitled an essentials of technical analysis for financial markets.

The book has the following key points (a) it explains how to navigate the markets successfully, including the top techniques for entries, exits, and risk management (b) straightforward descriptions of proven technical trading methods and strategies, and (c) filled with technical analysis insights, charts, and examples.

3 Proposed Work

We have proposed an ontological framework to mitigate the risk and generate greater returns. Figure 1 shows the working of the proposed framework, which starts from the technical analysis of any stock or index. This framework is an attempt to directly correlate the technical analysis with the help of chart pattern, volume, and price action with the help of some useful indicators. Thus, this framework is able to indicate the buying or selling opportunity more accurately as compared to any existing single indicator.

All the stocks of the companies are stored in the stock exchange database (we have taken BSE and NSE). The first task is to find out the appropriate stocks from the database based on the technical analysis and delivery-based volume of the stock. This technical analysis includes only stocks having two parameters, namely, (a) delivery-based volume greater than 50% for the last five consecutive days (b) chart of stock should follow any particular pattern stored in the database of ontological risk mitigation indicator (ORMI). The ORMI database contains various patterns like

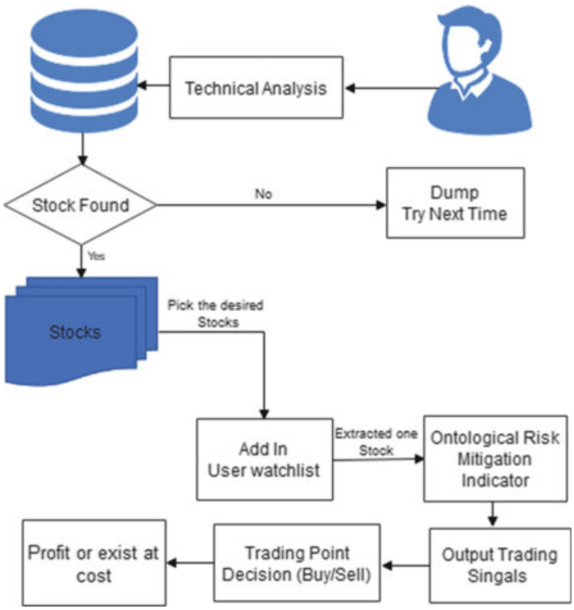


Fig. 1 Ontological risk mitigation framework

parallel channel pattern, ascending triangle pattern, cup and holder pattern, head and shoulder pattern, double top and double bottom pattern, etc. If the user did not find desired stocks, dump the process and try next time. However, if the user has any stock as per the above-mentioned criteria, then add it to the watch list. Now, pick one tested stock from the watchlist and run ORMI on the tested stock. The ORMI provides output trading signals from which users find trading point decisions (buying and selling). Now, the user either can get profit or exist at cost.

Ontological Risk Mitigation Indicator (ORMI) Algorithm

ORMI Database ← Different type of patterns.
    int i, m, n,
    m ← Previous 20 trading days
    n ← Upcoming trading days
    x ← Trough value of last trading day
    y ← Crest value of last trading day
    Golden ration ← 1.618
    $for(i \leftarrow 1, i \le n; i++)$
        MLi ← Average value of closing price of last m days
        ULi ← MLi + 2 × last m days standard deviation of prices
        LLi ← MLi − 2 × last n days standard deviation of prices
        Plot MLi, ULi, LLi
    $for(j \leftarrow 1, j++,$ until SLj >= 0.25)
        SLj ← $\frac{y}{golden\,ratio}$
        y ← SLj
      Plot SLj
    $for(k \leftarrow 1, k \le n; k++)$
            Candle_Openk = $\frac{1}{2}$ (Open of previous bar + Close of previous bar)
            Candle_Closek = $\frac{1}{4}$ (Open + High + Low + Close)
            Candle_Highk ← max (High, Open, Close)
            Candle_Lowk ← max (Low, Open, Close)
          Plot candle with values Candle_Openk, Candle_Closek, Candle_Highk, Candle_Lowk

Ontology model (C, P, I) ← features of Bollinger bands, Fibonacci retracement, and Heikin-Ashi

The proposed ORMI algorithm is taken the features of Bollinger bands, Fibonacci retracement, and Heikin-Ashi and stores all the features in the ontology for efficient search and query as the stock market is a place where numerous data is generated every second. This data is highly volatile that users need to access in a microsecond. This data is used in not only real-time but also historically. This is the place where ontology can resolve the problem of storage and access of this highly volatile and large data on real basis by interconnecting all the world market. Ontology stores the extracted features of Bollinger bands, Fibonacci retracement, and Heikin-Ashi in the form of Classes (C), Properties (P), and Instances (I). The ORMI algorithm starts to work when any of the patterns (that are stored in the database) follow by the chart. The proposed algorithm works better because it considers essential aspects of three indicators, namely Bollinger bands, Fibonacci retracement, and Heikin-Ashi altogether, that helps to detect the swing at an earlier stage and exist when the trend starts reversing. Hence two indications (i.e., earlier entry: when the trend is about to be positive, and earlier exit: when the trend is about to be negative) simultaneously triggers that mitigates the risk by providing better returns.

4 Results and Discussion

The proposed indicator is implemented in Java programming language, and ontology is developed by the protégé tool [11]. To test the proposed ORMI, we have taken the chart pattern of TCS share for the period of July–December 2021, i.e., for the last two quarters of the year 2021 (available: https://bit.ly/3IhFViz). Candlestick chart is used for the Bollinger band and Fibonacci retracement indicators to generate the points (entry and exit) and levels respectively, whereas the Heikin-Ashi is used for investment or trading duration. Figure 2a–d is show candlestick chart pattern of TCS share for the period of July to December 2021.

In Fig. 2a, the chart is prepared without using any indicator. So, it is difficult to predict the next move of the share. Hence, no trading or investment decision can be made. Figure 2b shows the two parallel channels that are depicted by the shaded region. The parallel chart is a kind of pattern based on technical analysis. The proposed ontological indicator works when any pattern is followed by a chart of any stock. Figure 2c shows the Bollinger band indicator applies to the above chart. This indicator indicates the entry and exit point for trading and investment. The indication triggers when the band is squeezed, and any candle crosses the middle line of the Bollinger band for given the closing below or above the middle line of the Bollinger band. The former two point (shown in blue) indicates the buying opportunity for trading and investment; and the latter two point (shown in pick) indicates the selling opportunity for trading and investment. Though the figure provides the trigger for entry and exit nevertheless, it is not able to predict about some certain level for the buying and selling opportunity, and it is also not able to predict the holding period of particular trade or investment. Figure 2d shows the Fibonacci indicator applied on the above chart. This indicator provides certain levels (38.2, 50, and 61.8%) for the triggers indicating buying or selling opportunities for trading and investment decision. These levels indicate that retracement can be seen in the price of the stock before it moves in a certain direction. With the help of these levels, traders or investors can be cautious if the price of the stock is near to these levels. Therefore, these points can be used for the profit booking or can be used as a stop loss as well as taking a new trade. However, since these points are too near and hard to predict the exact move of the stock, this is also not able to indicate the entry or exit points. Figure 2e shows the chart containing Heikin-Ashi. The Heikin-Ashi strongly indicates the positive or negative trend of the stock that results in deciding the buying or selling opportunity to the trader and investor. This indicator triggers a buying indication if 20 EMA (red

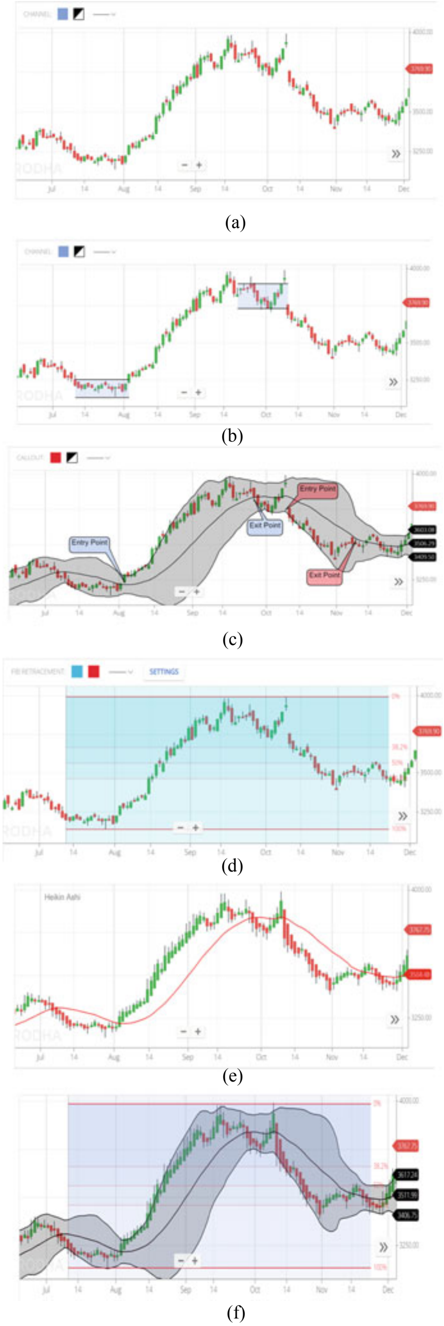


Fig. 2 **a** Stock, **b** stock selection based on technical analysis, **c** Bollinger band, **d** Fibonacci retracement, **e** Heikin-Ashi, **f** ORMI

line in Fig. 2e) is crossed by any green candle, and the next green candle gives its closing above to the previous green candle. The selling indication is triggered if 20 EMA line is crossed by any red candle and the closing of the next red candle is below the previous red candle. The trader or investor should have to keep its position until the Heikin-Ashi color is changed. However, this indicator does not provide any particular level for trading and investment decision. Figure 2f shows the proposed ontological indicator, which stored the features of these three indicators and pattern information in the ontology. The detailed comparative analysis and findings presented in the below-mentioned Table 1 show that the proposed indicator works better than the existing indicators.

Table 1 Comparison among Bollinger band, Fibonacci, Heikin-Ashi and ORMI

S.No.	Characteristics	Existing indicators	Explanation	Proposed indicator (ORMI)	Advantage of ORMI
1	Entry point	Bollinger band	Entry triggered on 3rd August as shown in Fig. 2c	Entry triggered on 3rd August, which is the earliest entry among these three indicators	The early entry provides an opportunity for getting the return for one more extra day, i.e., of 3rd August
		Fibonacci	Since a chart can have many different troughs or crests hence unable to predict an entry point as depicted in Fig. 2d		
		Heikin-Ashi	Entry triggered on 4th August as shown in Fig. 2e		
2	Exit point	Bollinger band	Exit triggered on 28th September as shown in Fig. 2c	Exit triggered on 27th September, which is earliest entry among these three indicators	Early exit restricts users from lossing the return on an extra day, i.e., of 28th September, which means users have mitigated their risk
		Fibonacci	Since a chart can have many different troughs or crests hence unable to predict an exit point as depicted in Fig. 2d		

(continued)

Table 1 (continued)

S.No.	Characteristics	Existing indicators	Explanation	Proposed indicator (ORMI)	Advantage of ORMI
		Heikin-Ashi	Exit triggered on 27th September as shown in Fig. 2e		
3	Duration of trading or investment	Bollinger band	Longer duration and less return as compared to ORMI due to losing money for one extra day	Less duration and higher return as compared to other indicators	Due to the short duration and higher return, it enables users to park their funds based on their choice in other investment or trading decisions for one more extra day to generate more profit
		Fibonacci	Since it is used for levels hence unable to predict the duration of trading or investment		
		Heikin-Ashi	Longer duration and less return than ORMI due to delayed entry by one day		
4	Levels	Bollinger band	No particular level is indicated except the entry or exit point by the middle band	Levels are provided to be used as a precautionary point for trading or investment decision	Due to the levels provided user can be more careful about the trend reversal. To mitigate the risk user can set the stop loss or may initiate the next opportunity for these levels
		Fibonacci	This indicator provides certain levels (38.2%, 50%, and 61.8%) for the triggers indicating buying or selling opportunities for trading and investment decision		
		Heikin-Ashi	No particular level is indicated except the entry or exit point by the 20 EMA		

5 Conclusion

The paper proposed an ontological risk mitigation framework that contains the features of three indicators (Bollinger band, Fibonacci retracement indicators, and Heikin-Ashi), pattern, and volume information stored in the ontology in the form of classes, properties, and instances. The results show that the proposed framework works better to generate high returns and mitigate the risk than existing indicators. Future work will be focused on creating an advanced ORMI that will automatically generate the investing or trading triggers by accessing the real-time data via interlinking the global markets.

References

1. Business management ideas. https://www.businessmanagementideas.com/essays/stock-exchange-essays/essay-on-stock-exchange-top-8-essays-business-management/11832
2. Mishra AK, Patel A, Jain S (2021) Impact of covid-19 outbreak on performance of indian banking sector. In: CEUR workshop proceedings
3. Rudik NI (2013) The encyclopedia of the indicator RSI (relative strength index). Corp Governance Int J Bus Soc
4. Lauguico S, Concepcion II R, Alejandrino J, Macasaet D, Tobias RR, Bandala A, Dadios E (2019) A fuzzy logic-based stock market trading algorithm using bollinger bands. In: 2019 IEEE 11th international conference on humanoid, nanotechnology, information technology, communication and control, environment, and management (HNICEM), pp 1–6
5. Sethi N, Bhateja N, Singh J, Mor P (2020) Fibonacci retracement in stock market. In: 5th International conference on next generation computing technologies (NGCT-2019)
6. Madbouly MM, Elkholy M, Gharib YM, Darwish SM (2020) Predicting stock market trends for japanese candlestick using cloud model. In: Joint European-US workshop on applications of invariance in computer vision. Springer, Cham, pp 628–645
7. Hassen OA, Darwish SM, Abu NA, Abidin ZZ (2020) Application of cloud model in qualitative forecasting for stock market trends. Entropy 22(9):991
8. Trivedi SR, Kyal AH (2020) Effective trading in financial markets using technical analysis. Routledge, India
9. Sundarakamatchi S, Gajanand MS (2021) Recommendation engine for stock market trading. Computational management. Springer, Cham, pp 499–516
10. Chen J (2010) Essentials of technical analysis for financial markets. Wiley
11. Patel A, Debnath NC, Mishra AK, Jain S (2021) Covid19-IBO: a Covid-19 impact on Indian banking ontology along with an efficient schema matching approach. N Gener Comput 39(3):647–676

Stock Price Prediction Using GRU, SimpleRNN and LSTM

Anjali A. Shejul, Aashay Chaudhari, Bharti A. Dixit, and B. Muni Lavanya

Abstract In today's era, stock prediction has become one of the dominant real world application. Most of the times, scientists attempted to establish a direct connection between information macroeconomic factors and stock returns; however, with the revelation of nonlinear slants in financial exchange record returns, there has been a significant shift in the scientists' focus toward the nonlinear expectation of stock returns. Even though various articles on nonlinear measurable presenting of stock returns have appeared since then, the huge demand is the nonlinear model which is specified before the estimation is performed. Predicting stock value is a difficult task that necessitates a solid algorithmic structure to calculate returns. Because stock prices are volatile and depend on the market up and down, forecasting stock prices becomes difficult. It has never been easy to invest in a portfolio of assets; the abnormalities of the financial market prevent simple models from accurately predicting future asset values. Machine learning, which is teaching computers to execute activities that would ordinarily need human intelligence, is the current scientific study hot topic. This paper explores gated recurrent units (GRU), simple recurrent neural network (Simple RNN) and long short term memory (LSTM) models for stock price prediction.

Keywords Stock prediction · Machine learning · Neural networks · LSTM · GRU · SimpleRNN

A. A. Shejul (✉) · A. Chaudhari · B. A. Dixit
School of CET, MIT World Peace University—Pune, Kothrud, Pune, Maharashtra, India
e-mail: anjali.shejul@mitwpu.edu.in

B. M. Lavanya
Department of CSE, JNTUA COE, Pulivendula, Andhra Pradesh, India

A. J. Kulkarni et al. (eds.), *Intelligent Systems and Applications*, Lecture Notes in Electrical Engineering 959, https://doi.org/10.1007/978-981-19-6581-4_42

1 Introduction

To buy and sell shares in a public corporation, stock exchange is used. The volatile nature of the stock market necessitates a great deal of analysis based on historical data [1]. To predict stock trends, historical time series data is used. The statistical examination of stock data plays vital role in stock price forecasting processes.

In this work to predict the stock prices of a specific share, three different algorithms are applied on training datasets. This model considers company's old share price and share's historical data. Date, Closing, Opening, Volume, High and Low are all characteristics of shares. This model applies temporal graphical analysis to predict a periodical share price. In this research paper, performance of GRU [2], SimpleRNN [3] and LSTM [2–5] is compared in terms of maximum accuracy in quantitative trading.

Long short term memory (LSTM) network is basically a recurrent neural network that can solve linear problems. LSTM is a deep learning technique. To learn very lengthy sequences, long term memory (LSTM) units are required. Because LSTMs address the evanescent gradient issue, they are more benign than other deep learning algorithms such as RNN or regular feedforward.

2 Objective

- To forecast the closing price of any share for next day and for specific period.
- To analyze impact of the number of epochs and overfitting on stock prediction results while training the model.
- To analyze the stock prediction results obtained with LSTM, GRU and SimpleRNN.

3 Literature Survey

The stock market prediction has sparked a lot of attention. It is necessary to build ways for gradually more accurate measurement. All of the procedures documented on the backslide have their own ideal circumstances and hurdles [6]. Straight backslide models are consistently fitted using the least squares approach, although they can be fitted in a variety of ways, such as by lowering the "non-appearance of fit" in another standard, or by minimizing a disabled form of the least squares setback task. Again, the least squares approach can be used to fit nonlinear models.

To forecast IBM daily stock returns in [7], author used neural networks. This study depicts various machine learning algorithms and ANN methods used for forecasting.

In paper [8], author used LSTM networks that can be utilized in the technique of predicting share values. A hybrid model with the combination of LSTM and GRU is proposed in [6]. First level predictor are generated by feeding input to LSTM

Table 1 Relevant observations [6]

	Single layer	Two layers	Three layer
LSTM	0.001	0.018	0.003
GRU	0.003	0.001	0.003

network. GRU layer will accept the output of the LSTM to get a final prediction. Author collected 66 years data and performed the experiment. Performance of the hybrid models compared to other LSTM models is better (Table 1).

4 Methodology

4.1 LSTM

LSTM networks are well suited to classify, process and predict based on time series data, as there may be an unknown length of time between significant events in the timeline. LSTMs have been established to address the extinct gradient problem that can be encountered when training traditional RNNs.

4.2 SimpleRNN

As traditional networks are not capable to handle sequential and time series data, recurrent neural network (RNN) comes to rescue. RNN is variation of artificial neural networks having capability to handle sequential and time series data. Also RNN is adapted to hold historical data. Due to memory component in RNN, previous state information can be stored to identify relationship among states. Feedback nature of RNN is helpful in producing temporal dynamic behavior [9, 10] (Fig. 1).

4.3 GRU

Kyunghyun Cho et al. invented gated recurrent units (GRUs). As name suggests, GRU has gated mechanism in RNN. Equivalence of GRU and LSTM is both has forget gate. GRU has less parameters compared to LSTM. For most of the application, performance of GRU is same as that of LSTM. For smaller and less frequent datasets, GRUs exhibit better performance (Fig. 2).

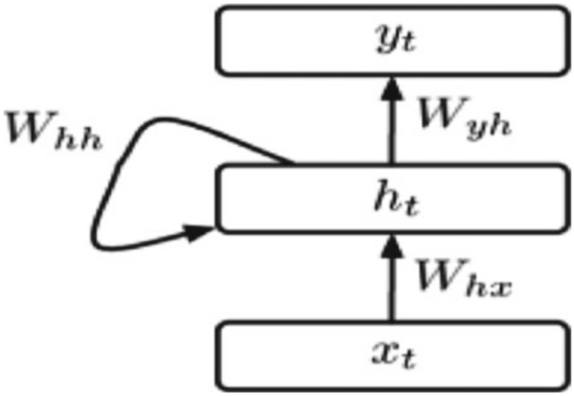


Fig.1 Block diagram of SimpleRNN

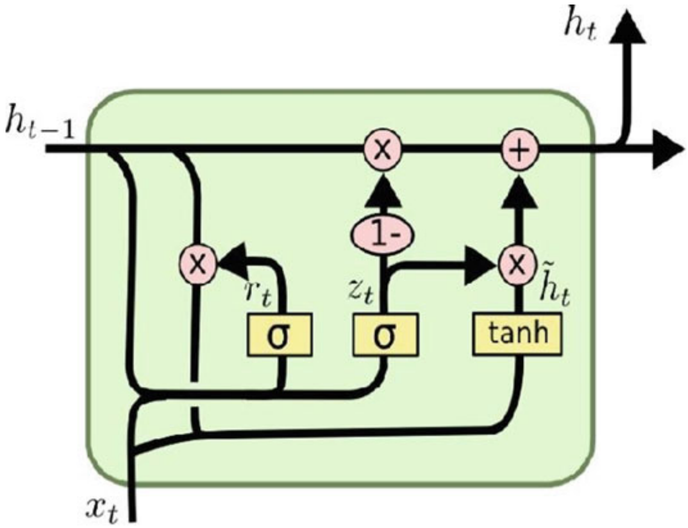


Fig. 2 Block diagram of GRU

5 Results

Increase in number of epochs increases closing price prediction accuracy of LSTM; however, too much increase in epochs also leads to overfitting ultimately impacting prediction accuracy. Below table gives the number of epochs and respective RMSE (error) obtained (Table 2).

For these observations, security under consideration is the NIFTY50 index having value 11,438 which is utilized for making these observations (Figs. 3 and 4).

Table 2 RMSE error obtained for different epoch

No. of years	Epoch	% Error
10	1	2.39
	3	1.78
	5	1.93
	9	0.95
	11	1.30
	15	1.48

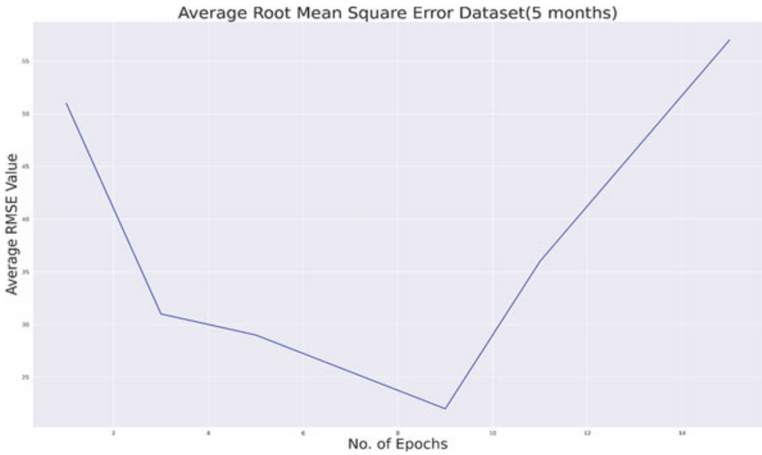


Fig. 3 Plot of RMSE values and number of epochs for 5 months

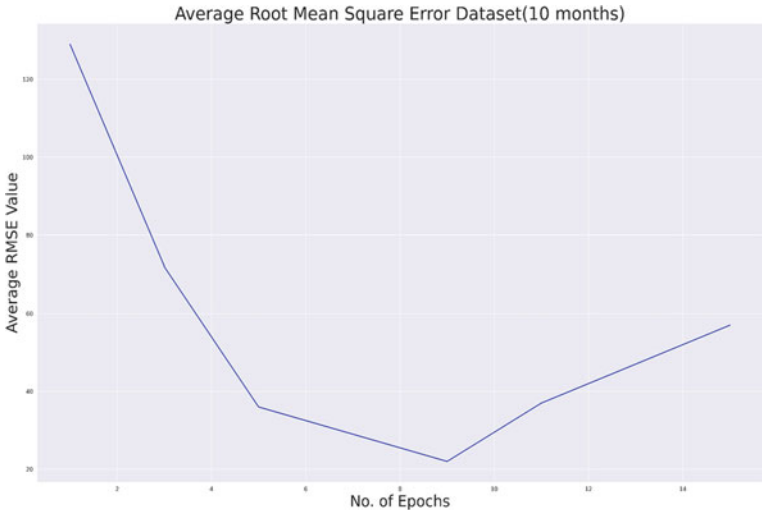


Fig. 4 Plot of RMSE values and number of epochs for 10 months

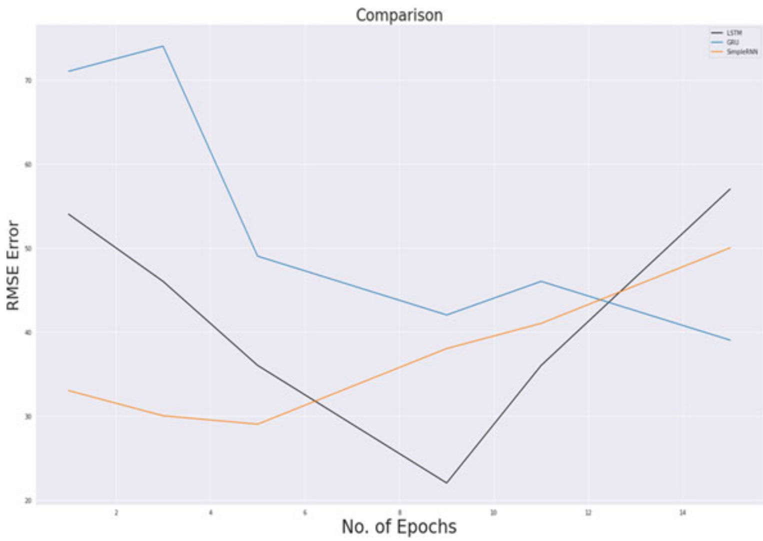


Fig. 5 Comparison between LSTM, GRU and SimpleRNN

Table 3 Evaluation of LSTM against GRU and SimpleRNN: (dataset-5 months)

Epoch	LSTM	GRU	SimpleRNN
1	54	71	33
3	46	74	30
5	36	49	29
9	22	42	38
11	36	46	41
15	57	39	50

Comparison made on basis of RMSE values

After trying different epochs at epoch 9, more accuracy is derived using LSTM In this case, learning rate, optimization algorithm and batch size are kept constant (Fig. 5; Table 3).

6 Conclusion

After performing parameter optimization, our neural networks achieved minimum average error of 0.9% using LSTM. This shows that the LSTM neural networks can give general idea of tomorrow's stock closing price position based on its training and the data supplied to it. To understand market movement in the future, this model can serve as efficient tool.

References

1. Leung CKS, MacKinnon RK, Wang Y (2014) A machine learning approach for stock price prediction. In: Proceedings of the 18th international database engineering and applications symposium
2. 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), pp 1597–1600. https://doi.org/10.1109/MWSCAS.2017.8053243
3. Prabowo YD, Warnars HLHS, Budiharto W, Kistijantoro AI, Heryadi Y, Lukas (2018) LSTM And SimpleRnn comparison in the problem of sequence to sequence on conversation data using Bahasa Indonesia. In: 2018 Indonesian association for pattern recognition international conference (INAPR), pp 51–56. https://doi.org/10.1109/INAPR.2018.8627029
4. Bensalah N, Ayad H, Adib A, Ibn El Farouk A (2021) LSTM vs. GRU for Arabic machine translation. In: Proceedings of the 12th international conference on soft computing and pattern recognition (SoCPaR 2020), pp 156–165. https://doi.org/10.1007/978-3-030-73689-7_16
5. Zhao Z et al (2017) LSTM network: a deep learning approach for short-term traffic forcast. IET Intell Transp Syst 11(2):68–75
6. Hossain MA, Karim R, Thulasiram R, Bruce ND, Wang Y (2019) Hybrid deep learning model for stock price prediction. In: 2018 IEEE symposium series on computational intelligence (SSCI). https://doi.org/10.1109/SSCI.2018.8628641
7. Kamalov FK (2020) Forecasting significant stock price changes using neural networks. Neural Comput Appl 17655–17667. https://doi.org/10.1007/s00521-020-04942-3
8. Chen X, Wei L, Xu J (2017) House price prediction using LSTM. In: Cornell computer science, machine learning. https://arxiv.org/abs/1709.08432v1
9. Talathi SS (2015) Schematic diagram of a simple RNN network. https://www.researchgate.net/publication/283761596_Improving_performance_of_recurrent_neural_network_with_relu_nonlinearity/figures?lo=1
10. Talathi SS, Vartak A (2016) Improving performance of recurrent neural network with relu nonlinearity. In: Cornell computer science, neural and evolutionary computing. https://arxiv.org/abs/1511.03771