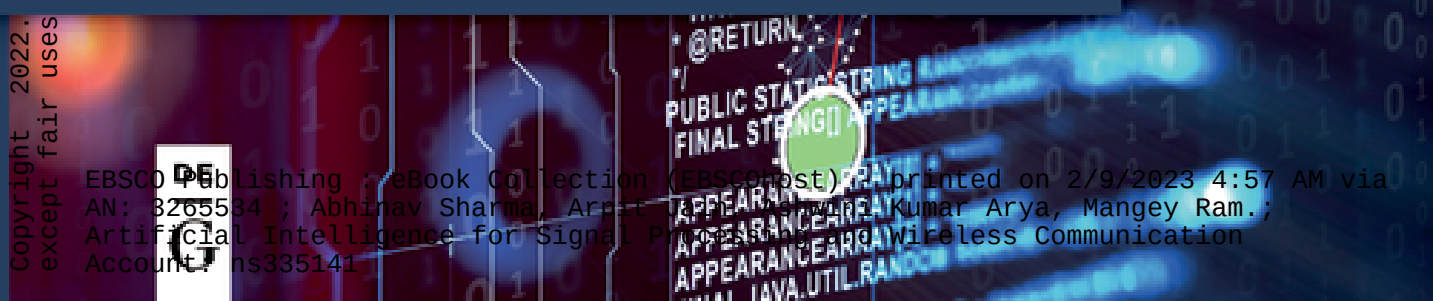
DE GRUYTER

ARTIFICIAL INTELLIGENCE FOR SIGNAL PROCESSING AND WIRELESS COMMUNICATION

Edited by Abhinav Sharma, Arpit Jain, Ashwini Kumar Arya et al.

DE GRUYTER SERIES ON THE APPLICATIONS OF MATHEMATICS IN ENGINEERING AND INFORMATION SCIENCES



Abhinav Sharma, Arpit Jain, Ashwini Kumar Arya, Mangey Ram (Eds.)
Artificial Intelligence for Signal Processing and Wireless Communication

De Gruyter Series on the Applications of Mathematics in Engineering and Information Sciences

Edited by
Mangey Ram

Volume 11

Artificial Intelligence for Signal Processing and Wireless Communication

Edited by
Abhinav Sharma, Arpit Jain, Ashwini Kumar Arya
and Mangey Ram

DE GRUYTER

Editors
Dr. Abhinav Sharma
Department of Electrical and Electronics Engineering
University of Petroleum and Energy Studies
Upes, Bidholi, Prem Nagar
Energy Acres
Dehradun 248007
Uttarakhand
India
abhinavgbpuat@gmail.com
abhinav.sharma@ddn.upes.ac.in

Dr. Arpit Jain
Senior Curriculum Leader
Byjus - White Hat Jr. Education Technologies Pvt. Ltd.
arpit.eic@gmail.com |
arpit.jain@whitehatjr.com

Dr. Ashwini Kumar Arya
Department of Electronics Engineering
Kyung Hee University
Gyeonggi-do
59 Dosu-ri
Toechon-myeon
Republic of Korea
akarya@khu.ac.kr

Prof. Dr. Mangey Ram
Department of Mathematics
Computer Sciences and Engineering
Graphic Era University
566/6 Bell Road
Clement Town, Dehradun 248002
Uttarakhand
India
drmrswami@yahoo.com

ISBN 978-3-11-073882-7
e-ISBN (PDF) 978-3-11-073465-2
e-ISBN (EPUB) 978-3-11-073472-0
ISSN 2626-5427

Library of Congress Control Number: 2021951627

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available on the Internet at http://dnb.dnb.de.

Cover image: MF3d/E+/Getty Images
Typesetting: Integra Software Services Pvt. Ltd.
Printing and binding: CPI books GmbH, Leck

www.degruyter.com

Acknowledgments

We acknowledge Walter de Gruyter for this opportunity and professional support. Our special thanks to Karin Sora, Daniel Tiemann, and Leonardo Milla (Walter de Gruyter) for the excellent support provided us to complete this book.

Thanks to the chapter authors and reviewers for their availability for this work.

Abhinav Sharma, University of Petroleum and Energy Studies, India
Arpit Jain, WhiteHat Education Technology Pvt. Ltd., India
Ashwini Kumar Arya, Kyung Hee University, South Korea
Mangey Ram, Graphic Era (Deemed to be University), India

https://doi.org/10.1515/9783110734652-202

Preface

Artificial intelligence (AI) is the science and engineering of making intelligent machines, especially intelligent computer programs. Machine learning (ML) and deep learning (DL) are the subfields of AI that gives computer the ability to learn without being explicitly programmed. Over the last few years, AI has discovered unprecedented rise in practical applications such as Google predictive search engine, Google assistant applications, and self-driving cars to name a few. In the current pandemic-driven global crisis, AI is proving to cater every frontier, may it be drug discovery, tracing, and diagnosing of COVID-19 patients. There are diverse applications of this technology in today's world such as in the field of agriculture, automation industry, health sector, engineering, education, and finance. This book focuses on the applications of AI in the field of digital signal processing and wireless communication. The organization of the book is as follows.

Chapter 1 presents the application of recurrent neural networks in sentiment analysis. The chapter provides details for implementing long short-term memory (LSTM) neural networks. The steps have been systematically divided into loading dataset, creating LSTM model, and training and testing the model performance. The author has assessed the performance of the network with various layers using loss and accuracy scores. Interested readers will also be able to understand the hyperparameters tuning for LSTM network and the significance of hyperparameters in improving the model performance.

Chapter 2 presents the application of DL and wireless sensor networks (WSN) in the field of precision agriculture. The chapter focuses on image processing-based disease identification using sensor-based data obtained from the agriculture field. An overview of various diseases and DL-based techniques used in industry to identify the diseases was provided to readers. Authors have provided a systematic review of applications from the literature. The chapter also integrates biosensors and Internet-of-things-based soil health monitoring with disease identification for improving the crop yield.

Chapter 3 focuses on review of computer vision and image processing techniques with applications pertaining to agriculture industry. The chapter provides a detailed review of various ML- and artificial neural network (ANN)-based techniques which can be deployed for image processing. The authors have provided a systematic methodology that can be utilized as a guide for implementing various ML techniques for image recognition. The chapter also explains various evaluation strategies available to assess the model performance. This chapter will act as a handy guide for readers who are interested in deploying ML-based applications in image processing and crop disease identification.

Chapter 4 presents the application of ANNs in the domain of wireless communication. With the modern communications advancing to 5G wireless communication standards, the service providers are rapidly switching over to 5G networks. Authors

https://doi.org/10.1515/9783110734652-203

have utilized ANNs and particle swarm optimization (PSO)-based methods to optimize a vertical stacked antenna for multiband and broadband features. The authors have provided detailed insights of the design process, and results for developed antenna have been analyzed for various frequencies.

Chapter 5 focuses on estimating the direction of signals in low signal-to-noise ratio environment using conventional and soft computing approaches. Direction of arrival (DOA) estimation is an important area of research in 4G/5G communication. Authors have developed Lévy flight mechanism-based moth flame optimization algorithm (LVMFO) to optimize deterministic maximum likelihood function for estimating the direction of narrow spaced signals.

Chapter 6 provides the techniques available for natural language processing (NLP). The NLP finds various applications in data science, sentiment analysis, and syntactic analysis to name a few. The authors have provided systematic review on data preprocessing and algorithm development. Various ML techniques and tools for NLP have been summarized along with their applications.

Chapter 7 presents the application of logistic regression for prediction of coronary artery disease. The dataset utilized by the authors is available as an open source in UCI ML repository. The performance of developed model has been evaluated on the basis of training and testing accuracy scores.

Chapter 8 presents a hybrid approach using bioinspired AI techniques for the design of fractal antenna. Author has utilized the capabilities of ANN and PSO for the efficient design of Sierpinski's gasket monopole antenna. The performance of the proposed methodology is validated by comparing the performance of simulated and fabricated results.

Chapter 9 presents an energy-efficient hybrid approach-based WSN deployment for railway track health monitoring. The chapter presents a systematic study of the possible ways of energy-efficient railway track condition monitoring systems. The analysis of various MAC protocols has been carried out with a conclusion that energy-efficient hybrid MAC protocols are more suitable for high data traffic applications. The contention-based MAC protocols and modified IEEE 802.15.4 standard MAC protocol are more suitable for moderate data traffic applications. The analysis concludes that the efficient spatiotemporal aggregation scheme can reduce the energy consumption to up to 50%.

Chapter 10 presents the application of ANN for acoustic signal classification for various milling operations. The model performance is evaluated for various iterations, and the performance has been compared based on R^2 values.

This book presents the implementation of ML and DL algorithms in audio, image, and video processing. Adaptive signal processing and biomedical signal processing are also explored through AI algorithms. Disease identification in plants and humans using feature extraction and classification is one of the important contents of this book. Localization and fault detection problems in WSNs are also identified using AI algorithms. Mathematical problems in signal processing and communications

such as DOA estimation that is too complex to be solved by conventional methods are explored using hybrid metaheuristic algorithms. Therefore, this book will be beneficial for undergraduate or postgraduate students, researchers from different fields, domain experts, industrialists, and academicians who are working on the applications of AI algorithms in the field of signal processing and wireless communication.

Editors
Abhinav Sharma
Arpit Jain
Ashwini Kumar Arya
Mangey Ram

Contents

Editors' biographies

Dr. Abhinav Sharma is working as an assistant professor (senior scale) in the Department of Electrical and Electronics Engineering in the University of Petroleum and Energy Studies, Dehradun, India. He received his B.Tech. from H. N. B. Garhwal University, Srinagar, India, in 2009, and M.Tech. and Ph.D. from Govind Ballabh Pant University of Agriculture and Technology, Pantnagar, India, in 2011 and 2016. He has taught subjects such as data communication networks, microprocessor and embedded system, artificial intelligence and machine learning, and communication systems. He has publications in IEEE, Taylor & Francis, Elsevier, Springer, Emerald, and many other national and international journals and conferences. His field of interest includes adaptive array signal processing, artificial intelligence and machine learning, and metaheuristic algorithms and smart antennas.

Dr. Arpit Jain is currently working as senior curriculum leader in WhiteHat Education Technology Pvt. Ltd., India. A seasoned academician having 11+ years of experience in university-level teaching in the field of electronics, control engineering, and machine learning, worked as assistant professor for 10 years at UPES India, a part of Global University System (GUS), the Netherlands. He has rich experience in curriculum design and has designed the curriculum for data analytics, and machine learning specializations. The areas of his research interests include real-time control system, fuzzy logic, machine learning, and neural networks. He has published research articles in SCI/Scopus indexed journals, edited books with IEEE, Emerald, RIVER, CRC, and many other reputed publishing houses. He received his B.Eng. from SVITS, Indore, in 2007, M.Eng. from Thapar University, Patiala, in 2009, and Ph.D. from UPES, India, in 2018.

Dr. Ashwini Kumar Arya is currently working as a research professor at the Institute for Wearable Convergence Electronics (IWCE), Department of Electronic Engineering in Kyung Hee University, South Korea. He received his B.E. from GBPEC Pauri Garhwal, India, in 2005 and M.Tech. from GBPUAT Pantnagar, India, in 2007, and Ph.D. from IIT Roorkee, India, in 2013, all with the major of electronics and communication engineering. He has rich experience in teaching and research, and served in various universities in India and abroad. He has published various research articles in SCI/Scopus indexed journals and conferences. His research focuses on the applications of RF engineering and EM theory in wireless communication and antenna designing technologies for various applications.

Prof. Dr. Mangey Ram received his Ph.D. major in mathematics and minor in computer science from G. B. Pant University of Agriculture and Technology, Pantnagar, India. He has been a faculty member for around 12 years and has taught several core courses in pure and applied mathematics at undergraduate, postgraduate, and doctorate levels. He is currently a research professor at Graphic Era (Deemed to be University), Dehradun, India. Before joining the Graphic Era, he was a deputy manager (probationary officer) with Syndicate Bank for a short period. He is editor in chief of *International Journal of Mathematical, Engineering and Management Sciences* and *Journal of Reliability and Statistical Studies*; editor in chief of six book series with *Elsevier*, CRC Press – A

https://doi.org/10.1515/9783110734652-205

Taylor and Francis Group, Walter De Gruyter (Germany), and River Publisher; and the guest editor and member of the editorial board of various journals. He has published 225+ research publications (journal articles/books/book chapters/conference articles) in IEEE, Taylor & Francis, Springer, Elsevier, Emerald, World Scientific, and many other national and international journals and conferences. Also, he has published more than 50 books (authored/edited) with international publishers like Elsevier, Springer Nature, CRC Press – A Taylor and Francis Group, Walter De Gruyter (Germany), and River Publisher. His fields of research are reliability theory and applied mathematics. He is a senior member of the IEEE, senior life member of Operational Research Society of India, Society for Reliability Engineering, Quality and Operations Management in India, and Indian Society of Industrial and Applied Mathematics. He has been a member of the organizing committee of a number of international and national conferences, seminars, and workshops. He has been conferred with Young Scientist Award by the Uttarakhand State Council for Science and Technology, Dehradun, in 2009. He has been awarded the Best Faculty Award in 2011, Research Excellence Award in 2015, and recently Outstanding Researcher Award in 2018 for his significant contribution in academics and research at Graphic Era (Deemed to be University), Dehradun, India.

Bharatendra Rai

Long short-term memory (LSTM) deep neural networks for sentiment classification

Abstract: Recurrent neural networks (RNNs) are useful for text data classification problems. However, when a sequence of words in text data has long-term dependencies, RNNs suffer from "vanishing gradient problem" that makes network training difficult for long sequence of words or integers. Long short-term memory (LSTM) neural networks are a special type of RNNs that help overcome this problem and make them possible to capture long-term dependencies between keywords or integers in a sequence that are separated by a large distance. This chapter provides an application example and illustrates steps for using LSTM deep neural network for movie review sentiment classification. The steps include text data preparation, creating LSTM model, training the model, and then assessing the model performance.

Keywords: RNN, LSTM, deep neural network, sentiment classification

1 Introduction

Deep learning models using a wide variety of neural networks have been used in different types of applications involving classification and prediction [1, 2]. Long short-term memory (LSTM) deep learning neural networks are a special type of recurrent neural networks (RNNs) that are useful with data involving sequences and provide certain advantages. Text data is one of the examples of data involving sequences. One of the key advantages of using LSTM networks lies in the fact that it addresses "vanishing gradient problem" that makes network training difficult for long sequence of words or integers. Gradients are used for updating RNN parameters and for long sequence of words or integers, and these gradients become smaller and smaller to the extent that effectively no network training can take place.

LSTM networks have been an active field of research. Yang et al. [3] used an LSTM network for video captioning application by taking sentences and video features as input for the model. Kong et al. [4] proposed an LSTM RNN-based framework for forecasting an electric load of a single energy user which is a challenging task due to the high volatility and uncertainty involved. Li et al. [5] proposed an

Acknowledgments: This work was carried out using Dell's Mobile Data Science Workstation Precision 7750 with NVIDIA Quadro RTX 5000. The author would like to thank Dell and Nvidia for loaning his powerful laptop to support his deep learning and artificial intelligence–related research.

Bharatendra Rai, University of Massachusetts, Dartmouth, USA, e-mail: brai@umassd.edu

https://doi.org/10.1515/9783110734652-001

evolutionary attention-based LSTM training with competitive random search for multivariate time series prediction. Sundermeyer et al. [6] compared count models to feedforward, recurrent, and LSTM neural network variants on two large vocabulary speech recognition tasks.

In this chapter, to illustrate steps involved in developing an LSTM model, we will make use of data on movie reviews from the Internet Movie Database (IMDb). This dataset is available within Keras library and consists of 25,000 movie reviews including train data that can be used for developing and assessing the model. Each of the 25,000 movie reviews has 12,500 positive and 12,500 negative movie reviews. We will make use of R language for this illustration. In the remaining part of the chapter, we discuss data preparation, creation of LSTM model, fitting an LSTM model using the movie review data, assessing the performance of the model, and finally we end this chapter with summary and conclusions in the last section.

2 Data preparation

We start by activating "keras" library and then obtain imdb movie review data by specifying a value for "num_words." Note that "num_words" is the number of most frequent words in the movie review text data that we want to include in our analysis. Although more words used may result in better outcome in terms of obtaining high sentiment classification accuracy, it can also lead to a significant increase in the amount of time needed for fitting the model. In the following R code, we have used 10000; however, one can explore other values too. Subsequently, we extract "train" data and store movie reviews in "train_x" and corresponding labels (positive or negative) in "train_y." Similarly, we extract "test" data and store movie reviews in "test_x" and corresponding labels (positive or negative) in "test_y":

```
library(keras)
imdb <- dataset_imdb(num_words = 10000)
c(c(train_x, train_y), c(test_x, test_y)) %<-% imdb
```

Each word in the movie review text is already preprocessed and each word is encoded as an integer value. In other words, sequence of words is available as a sequence of integers. These sequences of integers vary in length as some movie reviews are short and others are longer. In the "train" data minimum and maximum values for number of integers for each review are 11 and 2,494, respectively, and the median value is a 178. In the "test" data, minimum and maximum values for the number of integers for each movie review are 7 and 2,315, respectively, and the median value is 174. Thus, we observe that the length of movie reviews has high variability. Figure 1 shows the distribution of number of integers for each movie review in the "train" data.

The histogram shown in Figure 1 shows a right skewed pattern, indicating most of the movie reviews to have less than 500 integers and a few movie reviews to be lengthy. For developing an LSTM model we need to make the number of integers for each movie exactly the same. We can achieve this using "pad_sequences" function. In the following code provided, we make each movie review to contain exactly 500 integers:

```
train_x <- pad_sequences(train_x, maxlen = 500)
test_x <- pad_sequences(test_x, maxlen = 500)
```

To make each movie review to have exactly 500 integers, padding or truncation operation is performed on each movie review data. If a movie review has less than 500 integers, extra zeroes are added to the sequence of integers to artificially increase the number of integers to 500. This operation is called as padding. Similarly, if a movie review has more than 500 integers, the extra integers are deleted. This operation is called truncation.

This completes the data preparation process and in the next section we develop the LSTM model architecture.

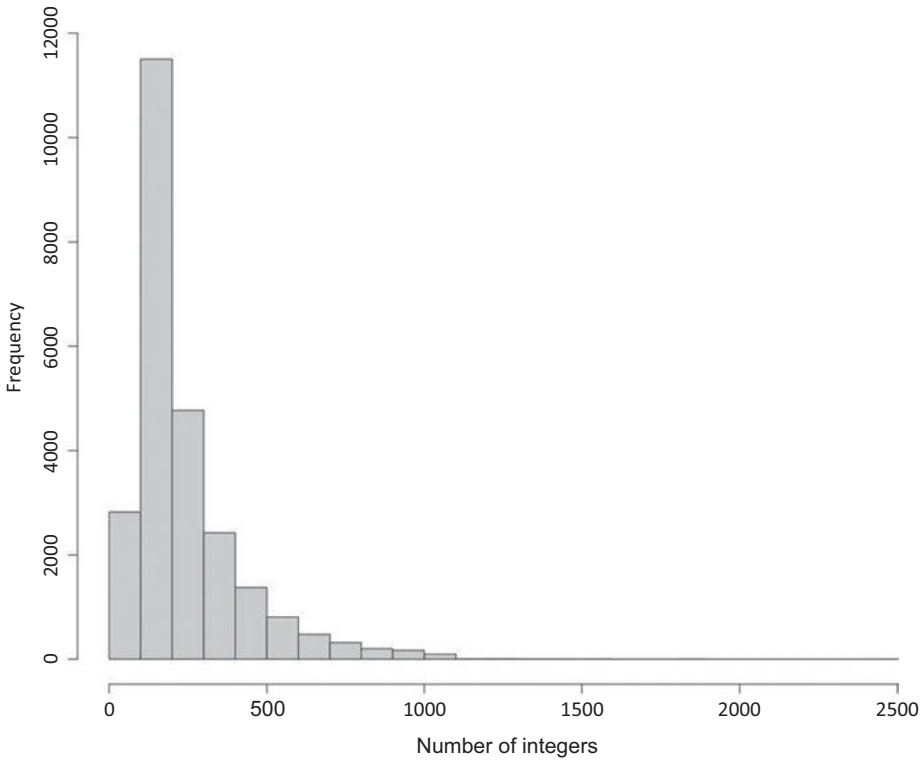


Figure 1: Histogram for number of integers per movie review in the "train" data.

3 LSTM model architecture

Keras library makes it easy to specify an architecture for the LSTM network. We use the following code to develop the architecture:

```
model <- keras_model_sequential()
model %>%
    layer_embedding(input_dim = 10000, output_dim = 32) %>%
    layer_lstm(units = 128, return_sequences = T) %>%
    layer_lstm(units = 16, return_sequences = T) %>%
    layer_lstm(units = 8) %>%
    layer_dense(units = 1, activation = "sigmoid")
```

We specify a sequential model for the LSTM network. In the embedding layer, we have specified "input_dim" to be 10000. Note that this should be the same number as "num_words" that we used in the previous section. If a different "num_words" is experimented with, it should be reflected in the embedding layer too. The network has three LSTM layers with the number of units as 128, 16, and 8. We may not know in advance what values to use for the number of units. In such situations, hyper-parameter tuning can be carried out by experimenting with different values and then choosing the one that gives better results. The last layer is a dense layer, and we use "sigmoid" activation function here. Note that in the three LSTM layers, the default activation function "tanh" is used as indicated in Figure 2.

The layers in the LSTM network and number of parameters are summarized in Table 1. As shown in the table, this LSTM network has 412,521 parameters.

Table 1: Summary of the LSTM network architecture.

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, None, 32)	320000
lstm_2 (LSTM)	(None, None, 128)	82432
lstm_1 (LSTM)	(None, None, 16)	9280
lstm (LSTM)	(None, 8)	800
dense (Dense)	(None, 1)	9
Total params: 412,521 Trainable params: 412,521 Non-trainable params: 0		

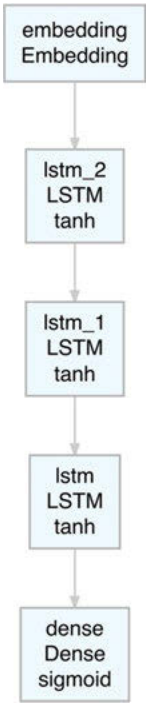


Figure 2: Layers in the LSTM network architecture.

After creating the network architecture, we can compile the model using the following code:

```
model %>% compile(optimizer = "rmsprop",
                  loss = "binary_crossentropy",
                  metrics = c("acc"))
```

When compiling the model, we specify "rmsprop" as optimizer which is useful when dealing with categorical response. We also specify the loss function as a "binary_-cross-entropy" since movie review sentiment is labeled as either positive or negative. We also specify "accuracy" as metric for assessing model performance during the network training. After compiling the model, we are ready to fit the LSTM model using "train" data.

4 Fitting the LSTM model

For fitting the LSTM model for movie review sentiment classification, we use the following code:

```
model_one <- model %>% fit(train_x, train_y,
                           epochs = 10,
                           batch_size = 128,
                           validation_split = 0.2)
```

We run the model with 10 epochs and a batch size of 128. We have also used 20% of the training data as validation data for assessing the model performance after each epoch. In order to obtain high accuracy values for the "test" data, we carried out hyperparameter tuning by changing number of units in the three LSTM layers, and also experimented with different values for the number of most frequent words and for "maxlen" when carrying out padding and truncation operation. Table 2 summarizes factors and levels used during hyperparameter tuning of the LSTM network to arrive at high accuracy in classifying movie review sentiment.

Table 2: Factors and levels used in hyperparameter tuning.

Factors	Level 1	Level 2	Level 3
Units in LSTM layer 1	32	64	128
Units in LSTM layer 2	16	32	64
Units in LSTM layer 3	8	16	32
num_words	5,000	10,000	–
Maxlen	200	500	–

As shown in Table 2, hyperparameter tuning involved a total of 108 ($3 \times 3 \times 3 \times 2 \times 2 = 108$) trials. To save time while running these experiments, they were performed on Dell's Mobile Data Science Workstation Precision 7750 with NVIDIA Quadro RTX 5000. Using this graphics processing unit (gpu) computing, we were able to complete all the experimental runs in about 21 h.

5 Results and model assessment

The best accuracy value of 88.22% was achieved based on 20% validation data used during fitting of the LSTM model. The best levels of factors explored during hyperparameter tuning were:

Units in LSTM layer-1: 128
Units in LSTM layer-1: 16
Units in LSTM layer-1: 8
num_words: 10000
maxlen: 500

Figure 3 shows accuracy and loss values for the best combination using training and validation data during 10 epochs.

From Figure 3, the following observations can be made:

- Loss values for training data continue to decrease from epoch 1 to epoch 10; however, the rate of decrease slows down toward the end.
- Loss values based on validation data decrease significantly from epoch 1 to epoch 2. Epochs 8–10 indicate that the loss values become approximately constant, and we may not be able to achieve any major improvements by having more than 10 epochs.
- Accuracy values for training data increase from epoch 1 (approximately 55%) to epoch 2 (approximately 80%). Subsequently, an increase in accuracy values slows down.
- Accuracy values based on validation data too increase significantly from epoch 1 (approximately 74%) to epoch 2 (approximately 85%). However, accuracy values for the last three epochs are approximately constant, indicating 10 epochs to be sufficient for fitting the LSTM model.
- At epoch 10, difference between loss and accuracy values based on training and validation data seems reasonable and does not show any cause for concern regarding overfitting.

The results obtained from LSTM network with hyperparameter tuning were compared with RNN model and LSTM model without hyperparameter tuning. The results are summarized in Table 3.

From Table 3 we can observe that the LSTM network performs better than the RNN model when classifying the movie review sentiment (over 12% improvement based on the “test” data). Use of GPU computing for carrying out hyperparameter tuning helps us to improve “test” data movie review sentiment classification accuracy by an additional 2%. Chollet and Allaire [7] carried out IMDB movie review sentiment classification using LSTM network and reported a validation accuracy of 88% using IMDB movie review data. However, test data accuracy is not reported in the study.

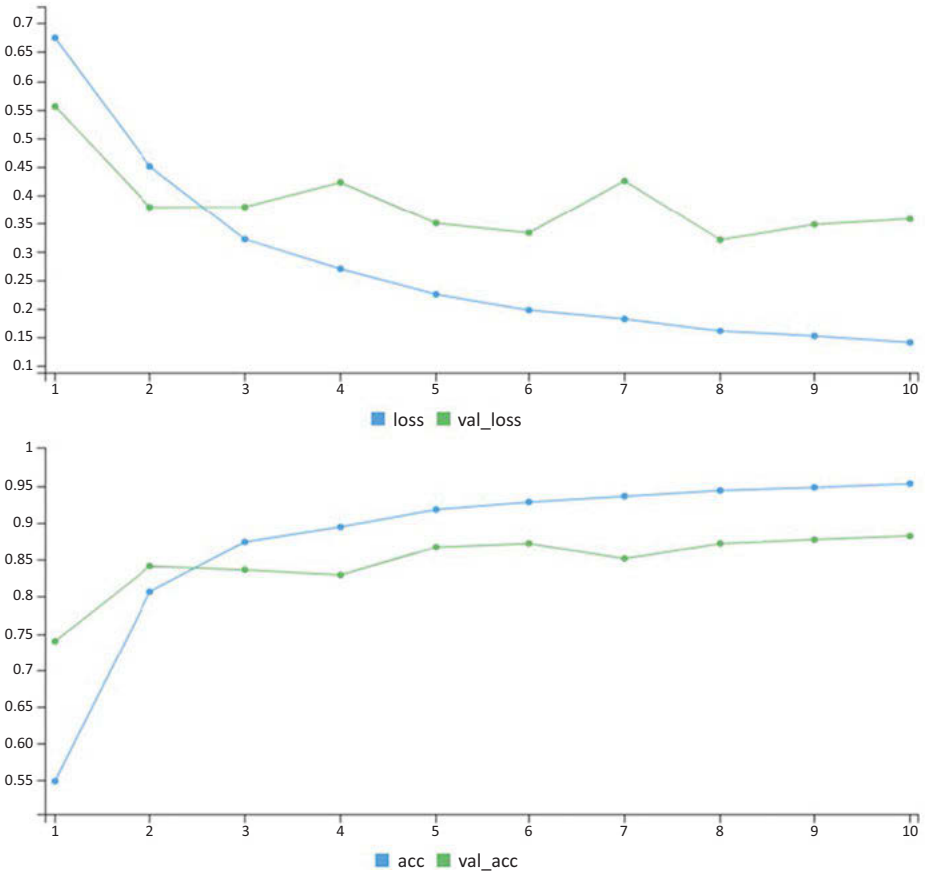


Figure 3: Loss and accuracy for training and validation data during model fitting.

Table 3: Summary of accuracy based on validation and test data.

Type of model	Accuracy based on validation data	Accuracy based on "test" data
RNN	71.18%	71.44%
LSTM	83.02%	84.08%
LSTM with hyperparameter tuning	88.22%	86.22%

6 Conclusion

In this chapter, we illustrated the use of LSTM networks for developing a movie review sentiment classification model. LSTM networks are a type of RNN with certain advantage. One of the problems faced by RNNs involves difficulty in capturing long-term dependency that may exist between two words or integers in a sequence of words/integers. We noted in this chapter that the length of movie reviews varies a lot. LSTM networks are designed to artificially retain long-term memories that are important when dealing with long sentences or long sequence of integers.

For arriving at high movie review sentiment classification accuracy, we utilized GPU computing for carrying out hyperparameter tuning experiments. It helped us to improve test data accuracy in correctly classifying movie review sentiment as positive or negative from about 71% for RNN to about 86% for LSTM with hyperparameter tuning.

References

[1] B. K. Rai Advanced Deep Learning with R: Become an expert at designing, building, and improving advanced neural network models using R, Packt Publishing, December 2019.

[2] B. K. Rai and A. Meshram, Application of neural network to detect freezing of gait in patients with Parkinson's disease, In M. Ram and S. B. Singh, ed., *Soft Computing*, De Gruyter, Published in 2020, 2020.

[3] Y. Yang, et al., Video captioning by adversarial LSTM, *IEEE Transactions on Image Processing*, 27(11), 5600–5611, 2018-11.

[4] W. Kong, et al., Short-term residential load forecasting based on LSTM recurrent neural network, *IEEE Transactions on Smart Grid*, 10(1), 841–851, 2019-01.

[5] Y. Li, et al., EA-LSTM: Evolutionary attention-based LSTM for time series prediction, *Knowledge-based Systems*, 181, 104785, 2019-10-01.

[6] M. Sundermeyer, et al., From feedforward to recurrent LSTM neural networks for language modeling, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3), 517–529, 2015-03-01.

[7] F. Chollet and J. Allaire, *Deep Learning with R*, Manning Publications.

Sathiya S., Cecil Antony, Praveen Kumar Ghodke

Plant disease identification using IoT and deep learning algorithms

Abstract: Precision agriculture implies the technological fusion in agricultural management, and it ensures the highest crop yield at low cost by precise monitoring and control of parameters related to plant health status. The measurand from different sensors connected with wireless networks plays a vital role in efficient crop management with less manual intervention. The sensor fusion approach detects and understands parameters such as soil moisture, temperatures, salinity, climate conditions, and plant growth. The measured data is used for decision-making in the utilization of water, pesticides, and fertilizers. In rapidly growing precision farming, the early identification of disease aids to effectively prevent the plants from diseases and its negative impacts on crop yields. The conventional methods of disease identification are greatly dependent on the expert's experience, also expensive and time-consuming. Hence, fast and precise methods are needed for the identification of plant diseases for efficient agricultural management. The extensive range of data from various sensors with the Internet of things provides a primary data source for acquiring disease information. To accurately identify the disease type, deep learning can be applied. Like humans, the deep learning approach can respond by training themselves from a vast amount of sensor data and associated image processing techniques. With the use of trained experience, it decides on its own for the accurate identification of plant diseases. This chapter explains the advancement in precision farming and the identification of plant diseases with a deep learning algorithm which further improves the food safety and efficiency of crop management.

Keywords: deep learning, disease detection, Internet of things (IoT), precision agriculture, sensor fusion, smart farming

Sathiya S., Department of Instrumentation and Control Engineering, Dr. B. R. Ambedkar National Institute of Technology Jalandhar, Jalandhar 144011, Punjab, India, e-mail: sathiya@nitj.ac.in
Cecil Antony, School of Biotechnology, National Institute of Technology Calicut, Kozhikode 673601, Kerala, India, e-mail: cecil@nitc.ac.in
Praveen Kumar Ghodke, Department of Chemical Engineering, National Institute of Technology Calicut, Kozhikode 673601, Kerala, India, e-mail: praveenkg@nitc.ac.in

https://doi.org/10.1515/9783110734652-002

1 Introduction to food quality and safety

As the global population increases annually by about 1.6%, the demand for plant-based food of every kind is expected to create a drastic hike and several challenges in agriculture. Improving the quantity and quality of food product utilization rich in nutrients is a great challenge to global food management. The less quality food may lead to several deficiency diseases, nevertheless, in the population of developed countries. To prevent the world population from nutritional deficiencies, the production and monitoring of high food quality yields are necessary. Increasing the quality of food production not only meets the food demands but also helps the world population to prevent themselves from nutritional diseases. Currently, around 800 million people including children are suffering from hunger and malnourishment which creates a major threat in their quality lives for adults and the growth factor of children. These malnourished populations belong to different regions of the world, especially poor economic countries, where foods are highly contaminated and of less nutritional quality. Food-related diseases are a major problem worldwide, which consequently affects the human well-being and the economy. They are least bothered and less recorded in most of the countries, but the estimation says that diseases such as diarrhea and malnutrition deficiency are due to biological, chemical contamination, and less quality of plant-based foods. Though these diseases are not fatal, they greatly affect the growth of children and various deficiencies in adults which will lead to poor mental and physical health [1].

As per the survey, no harm for humans is recorded by the internationally recommended food additives, fertilizers, and pesticides. But the improper usage of those chemical additives and pesticides causes the risk to create health issues in the consumers. Metal contaminants and adulteration with plant toxins in food products are being a major threat to food-borne diseases. In underdeveloped and developing countries, the economic consequences also directly involve unsafe food products which lead to devastating health issues, especially for wage earners. Some other studies say that food-borne diseases are also caused by infections in cattle, poultry, and disease-creating specific plant pathogens. Controlling the environmental contaminants also helps in ensuring food safety and maintaining or improving food productivity [2]. Inspecting and ensuring the quality of food production play a vital role in the worldwide food market, specifically in agricultural-based food products. In recent days, acquiring quality assurance has been mandated from the raw material to the final consumer product. It defines the policies, methods, procedures, and standards to be followed in the production of agricultural-based food products which ensures their quality and standards delivered to the customers' marketplace. Quality assurance also helps improve the standards and methods followed in the food industry for enhancement of food quality and safety, right from farming to the final food product. The quality management associated with food production plays a vital role in developing countries which provide best practices

in agricultural food production [3]. Farming and its quality management are necessary to be developed since the world population will be about 9.1 billion, and the demand for food in 2050 is estimated to be elevated by 70%. Particularly in developing countries, the food demand will be doubled, and there is a need for producing a large quantity of quality and nutritious food. Agriculture possesses a major role in most countries for massive food production, especially in developing countries like China, India, and Brazil. Thus, the farming process should be continuously maintained, managed, and protected by the farmers from sowing seed to the final yield. Due to this requirement, there is a need for smart farming by integrating the agricultural process with the latest technologies, called "precision agriculture" [4].

2 Precision agriculture

Precision agriculture is an integration of the latest technologies deployed to monitoring and management of wide-scale commercialized agriculture. The importance of the combination of technologies is to obtain a high yield of plants or crops with low investment. Automatically monitoring the various parameters such as weather conditions, optimal plant growth, and soil conditions by using sensors and communication technologies results in improved crop management, reduced labor cost, and less wastage as illustrated in Figure 1. The increasing popularity of wireless sensor networking is used for real-time monitoring of environmental, climatic, and soil conditions for large-scale agriculture [5].

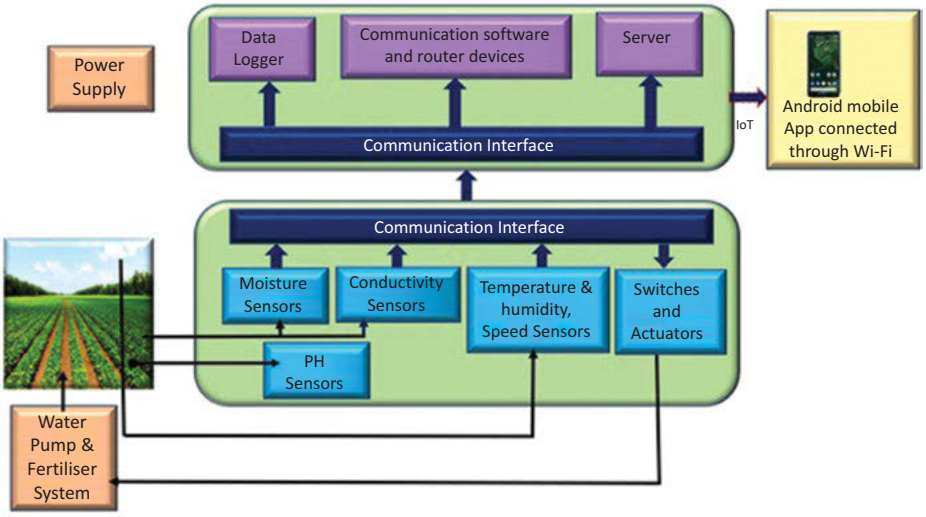


Figure 1: Sample functional block diagram of precision farming.

2.1 Sensing technologies used in precision agriculture

Various sensors are used to collect data of different physical and chemical parameters, and the actuators are used to react accordingly for tuning the parameters at desirable values with the help of a feedback control system. These sensors also accumulate information about plant growth, weeds, and pest, and pathogen-based plant diseases, and estimate fertilizers feed, plant growth, watering, and labor involvement. Based on the data collected from sensors, the data is fed to an intelligent system for decision-making which helps to estimate the following essential parameters [6]:

(i) Estimation of crop needs
(ii) Additional requirements for crops at different weather and environmental conditions
(iii) Fertilizer requirements
(iv) Identification of temperature and humidity
(v) Plant growth and disease identification
(vi) Water requirements

The widest sensors for monitoring the different agriculture parameters are as follows:

2.1.1 Sensor for measuring the water content in the soil

The water content of the soil is the water content in the sample soil to the total amount of the sample soil, and the capacitance-based sensor is used for measurement. The basic principle is the dielectric constant of the soil which changes the dielectric constant between the two plates of the capacitive sensor. The dielectric constant of the water is high (=80) compared to the soil constituent (=4) and air (=1). Thus, the change of water content in the soil substantially changes the capacitance value of the sensor, which can be further converted as the frequency change by employing the capacitive sensor into the oscillator circuit [7].

2.1.2 Sensor for measuring the soil moisture

The tension of the soil water is measured, which is the indication of the plant root system's quantum effort to extract water from the soil. The dry soil requires more effort, and wet soil requires less effort for the extraction of water from the soil. The resistance between the electrodes mounted into the gypsum block is measured. The sensor is placed into the soil, where the moisture is to be measured. When the soil is wet, the water content in the gypsum is more; thus, the resistance between the electrodes is less and vice versa [8].

2.1.3 Sensor for measuring soil electrical conductivity

The different soil variables are mapped by measuring the electrical conductivity of the soil. The salinity of soil is determined with the use of electrical conductivity. The working principle of the electrical conductivity sensor of soil is based on Faraday's law, which states that the change in the linkage of electromagnetic flux is directly proportional to the electromagnetic flux induced in a coil and it is measured in terms of milli-Siemens/meter. The measure of electrical conductivity is related to other soil parameters such as the porosity of the soil, water content, salinity, temperature, amount of organic matter, and clay materials present in the soil [9].

2.1.4 Soil pH sensor

The pH measurement range is from 0 to 14, where the solutions having pH values less than 7 are acidic, and values more than 7 are basic solutions. The soil which lacks essential nutrients is beyond the range of pH between 5.5 and 6.5, and is not optimum for agriculture. The fertilizers are applied to regulate the soil pH value within the desirable range, which improves crop production. The efficient way of fertilizer application is to study the pH of soil depending on the spatial variation, which needs a pH sensor to measure. The pH sensor has three important components such as a measuring electrode, a reference electrode, and a power source. The positive terminal of the battery power source is connected to the measuring electrode and the negative terminal is connected to the reference electrode. The measuring electrode is sensitive to the hydrogen ion concentration in the measurement sample which develops the potential. The reference electrode is insensitive to hydrogen ion concentration, which provides a constant potential. The pH meter is immersed into the sample, which gives the difference in potential between the measuring and reference electrode directly proportional to the pH value. At a neutral pH value of 7, the voltage difference is 0; if the sample is acidic, then the potential difference is measured, which is having opposite polarity for the potential difference produced by the basic sample [10].

2.1.5 Sensor for weed seek

This is used to identify the weeds and to spray the herbicide only onto the desired location precisely. It reduces the usage of chemicals and reduces the cost of applying the herbicide. It uses an active light source that emits the focused light beam onto the ground, and the associated circuit detects the weeds so that the valve sprays the chemical only onto the weeds precisely [11].

2.1.6 Sensor for measuring temperature

In agriculture, it is important to measure the soil temperature which helps to decide the type of crops suitable for a particular agricultural field. It can also produce an alert if the temperature crosses beyond the desired limit. The widest sensor is based on a P–N junction diode fabricated using complementary metal oxide semiconductor (CMOS) technology along with ion-sensitive field-effect transistor (ISFET) [12].

2.1.7 Sensor for measuring wind speed

The sensor is used to estimate the surface wind speed in terms of a two-dimensional vector. There are two types of wind speed such as instantaneous and average. The wind speed is changed dynamically, which is measured by a cup anemometer. It comprises three polypropylene cups and a stainless steel shaft guided by oil bearings. The average speed is determined by averaging the three instantaneous wind speeds for 10 min duration. To calculate the wind speed, a hall effect sensor is used, which produces electronic pulses based on the wind velocity [13].

2.2 Communication system involved in precision agriculture

In recent years, the introduction of advanced sensors has increased their usage in every aspect of life. The conversion of any physical or bioparameters into electrical quantity is used by communication technology to transfer the data from one location to other called “wireless sensor networks (WSN).” The WSN contains several nodes, which are devices that collect the data requirements as shown in Figure 2.

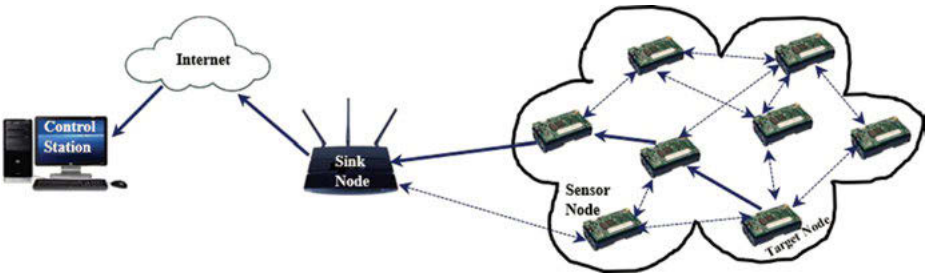


Figure 2: Operational diagram of wireless sensor network (WSN).

The WSN performs three functions such as sensing, communication, and computation with the help of various hardware and software protocols. The node that collects data from different sensors is called the source node. The node that collects data

from different source nodes is called a sink node or gateway node, which has high computing power. The addition of actuators in WSN is called as wireless sensor and actuator network used for both monitoring and controlling the parameters [6].

2.2.1 Communication technologies

The different communication technologies used for WSN such as Bluetooth, ZigBee, Wibree, and Wi-Fi have unique properties and capabilities. The ZigBee wireless technology (IEEE 802.15.4) is preferred over other technologies as it is cheap and consumes less power comparatively. The technology is introduced in 2003, and it is applied to various fields such as medical, industrial, and scientific research [6].

2.2.2 Wireless sensor node architecture

It is a basic element for WSN that has four modules which are sensor/actuator, communication, processing, and power modules as illustrated in Figure 3. Additional external memory can be connected if data storage is required for decision-making locally. The module of sensor/actuator is used to interface all sensors and actuators, which are considered based on the domain application, problem, and the way of distribution. The major parameters to be considered for the sensor node are the processor, memory size, frequency bandwidth, range of transmission, and compactness [14–16].

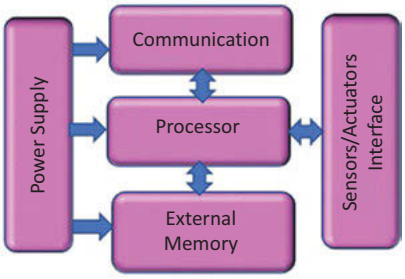


Figure 3: Architecture of wireless sensor node.

2.2.3 Major issues in wireless sensor networks

The WSN is applied in many exciting applications due to its flexible concepts. But the application of WSN in agriculture has many issues to be resolved before deployment of the technology. In large-scale farming, it is being done for several acres of land which may have spatial variations, resources, and microlevel changes in the climatic conditions. The design and deployment issues should also be considered as it

is mounted on the open environment where the parameter is uncontrolled. The major issues are discussed as follows [17–19].

2.2.3.1 Energy consumption

The event detection, data manipulation, and transmission of data tasks are assigned to each sensor node of WSN, whereas in the multihop network, the data routing is also assigned. All these tasks require energy for its operation which is usually provided by lithium or alkaline batteries. The lifetime of the sensor is dependent on the life of the battery. Thus, to enhance battery lifetime, the power management of various hardware and software is required. Though some of the renewable energy sources such as wind and solar can be implemented, the sensors installed in agricultural land require periodic recharging.

2.2.3.2 Acquisition, sampling, and transmission of data

As the energy is required for the operation of data collection, sampling, and transmission of data, by optimizing the data collection and sampling with proper programming, the energy will be efficiently used and saved. But in case of frequent and huge data collection, the energy source will be rapidly exhausted. The data collection and transmission in agriculture should be regulated to save energy based on the location and measured parameters. This can be done by storing the data locally and it can be transmitted as aggregated values of deviations intelligently. The sleep mode of transmitters can also be implemented, and it can be wakened up only when it is needed. The sensor nodes should be connected nearer to each other to ensure reliable multihop communication.

2.2.3.3 Fault tolerance

In agriculture, the sensors are installed in an open environment, which is prone to damages, interference in the transmission of signals, and blockages. The redundancy of sensor nodes, reorganization of a sensor network, and overlapping of sensor regions ensure the reliability of the WSN.

2.2.3.4 Size of sensor node and its housing

The sensor node should be compact and suitable for installation. It should also be shielded with housing so that it will be protected from physical damages due to environmental hazards such as rain, wind, animal, and mishandling by nonprofessionals.

2.2.3.5 Placement of sensor node

The sensor design is based on measuring parameters, algorithm, topology, and network, and also it should be carefully deployed to ensure the reliability of WSNs. As

the sensor should have maximum coverage for a parameter measurement, the position and altitudes should be carefully observed during installation. For example, the light sensors should be placed at a higher position to avoid the plant leaf hindrance, and the water and moisture should be placed at ground level.

2.3 Applications of wireless sensor networks in precision agriculture

The most common applications of WSNs in precision agriculture are smart irrigation, smart fertilization, smart pest control, and early disease detection [20, 21].

2.3.1 Smart irrigation systems

The smart irrigation system is an intelligent system that decides the requirement of water, based on the information collected from sensors. It plays a vital role to enhance plant health and productivity, and reduces cost. It is used to reduce water wastage by measuring the water content of the soil by using an electrical conductivity sensor, soil moisture sensor, and measuring the environmental conditions by using temperature and humidity sensors. The water requirement is decided by the intelligent processor and the estimated water level will be supplied by the actuators automatically. A cost-effective smart irrigation system can be done by Raspberry Pi, Arduino, electronic control, and valve. The ZigBee communication protocol can also be used.

2.3.2 Smart fertilization system

The fertilizer is a naturally or artificially acquired chemical substance utilized for enhancement of plant growth and yield, and also it is sprayed manually in the fields with the traditional method. The utilization of the fertilizer efficiently and precisely is done with a decision made by the intelligent system based on the data acquired from the different sensors that measure the fertilizer requirement at different locations. It utilizes the sensor called “pendulum sensor” mounted on the tractor, and measures the density of the crop. Based on the data collected from the sensor, the decision support module of the smart fertilization system sprays the fertilizers efficiently and precisely only at the required level. The IEEE 802.11 Wi-Fi communication system along with the global positioning system is used for a smart fertilization system.

2.3.3 Smart pest control and early disease detection systems

The pest attack results in disastrous diseases that affect the growth of the plant, thus reducing productivity. However, the early detection of diseases helps farmers to prevent the crop from damages by implementing proper control methods at the right time. The sensors such as RGB, fluorescence image, and spectral and thermal sensors are acquiring the data, which is further used by the intelligent system to take proper decisions on plant health status and to prevent plants from diseases. The temperature is related to the water level in the plants, which is measured by thermal sensors, and RGB sensors measure the biometric effect of the plant. Multi- and hyperspectral imaging sensors obtain the images of spatial information of objects. The photosynthesis process of the plant is monitored by fluorescence sensors and various imaging sensors, which are used to identify different types of diseases in the plants. The temperature, humidity, and wind speed sensors are used to monitor plant health and growth status based on the changes in weather and environmental parameters. The Internet of things (IoT) technology is used for pest control and disease detection, where sensors are deployed in the plants, and then the collected data is sent to the cloud. The farmer will receive information about the plant growth and health status and the decision to be taken further. The hyperspectral image sensors are used to analyze the health status and pest attack of plants by taking images from a different point of view with the help of mounted imaging cameras in the aerial manned or unmanned vehicles. These captured images are further analyzed deeply by machine learning (ML) algorithms for accurate identification of the disease. As deep learning algorithm neural networks could learn complex patterns and images, they are well utilized in the identification of different diseases by using the images captured by hyperspectral imaging camera sensors.

3 Plant diseases: a major threat to global food productivity and quality

Approximately 800 million people of the world population are suffering from food deficiency and 10% of food production is affected by plant diseases, and several researchers in plant pathology found that there is a close relationship between global food shortage and reduced food production due to plant pathogens [22, 23]. Few crops play the largest part of food consumption for the worldwide population, whereas few other crops are less intensive in their growth but provide essential nutrients in specific regions of the world. The productivity of these crops is damaged by the plant diseases during pre- and postharvesting periods caused by pathogens such as viruses, bacteria, fungus, nematodes, oomycetes, and other parasitic plants. In the 1840s, almost a million Irish people died due to starvation as their high dependency on potatoes,

which were severely damaged by a pathogen called *Phytophthora infestans* in Ireland [24]. Also in 1943, around 2 million people died due to Great Bengal famine where the rice crop was severely damaged by *Cochliobolus miyabeanus* [25], and during 1970–71, the corn crop was also affected by *Cochliobolus heterostrophus* in the USA, and their agricultural economy was greatly affected [26]. From the disasters, the high dependency of a large population on a single crop or very few crops is at risk when the crops are affected by the most destructive plant diseases. Particularly, the developing countries are at more risk as their population is growing at a faster rate, poorly resourced research and development, and poverty. As globalization promotes the native plants to be grown in new regions which are far from their origin, more likely to be affected by the new strains of pathogens. Due to poor governance and fewer resources in developing countries, the data for quantifying pathogenic diseases that reduce high yields and food products is quite difficult to acquire. Other issues are the accurate identification of various plant pathogens and the related characteristics such as their species, taxonomic grouping, strain, variant prokaryotic strain, and races, so that relevant control of the disease can be deployed.

3.1 Virus-based plant disease

Most of these virus-based plant diseases still cause a significant economic loss in a wide variety of crops. Around 700 familiar viruses create disastrous diseases in plants that have a wide range of hosts. Most of the viruses are single-stranded RNA viruses and other forms are double-stranded DNA viruses. For example, Barley yellow dwarf viruses spread worldwide which affected around 150 species of Gramineae or Poaceae including rice, maize, barley, and oats. But the most strongly appearing viruses in a wide variety of plants are tobacco mosaic virus, tomato spotted wilt virus, tomato yellow leaf curl virus, cucumber mosaic virus, potato virus Y, cauliflower mosaic virus, African cassava mosaic virus, plum pox virus, brome mosaic virus, and potato virus X [27].

3.2 Bacterial disease

Bacterial species belonging to different genera are the most damage-causing plant pathogens, almost 350 varieties of plant-based diseases are due to the species called *Xanthomonas* [28] *oryzae* PV. Oryzae is a type of bacterial disease that causes a major threat to the rice plants, particularly basmati rice in tropical Asian regions, and also the production of rice was severely affected in India by this bacterial disease [29]. Other bacterial diseases reported recently in Uganda affect the banana yields which produces symptoms in plants such as discoloration of vascular vessels, internal rotting in fruits, and rapid yellow. These symptoms are very much like the symptoms

caused by a pathogen called *Ralstonia solanacearum* [30]. Thus, the bacterial disease-causing pathogens must be accurately identified with their species, strain, taxonomic grouping, and races, so that the control of disease spreading can be properly deployed. For example, *Ralstonia solanacearum* exists with five races based on different host and geographical distributions. Currently, these pathogens are identified by the comparison of bacterium DNA fingerprints from rep-PCR (polymerase chain reaction) [31]. Some other commonly occurring bacterial diseases in the crops are black rot in brassicas; bacterial canker in tomato, capsicum, and chili; bacterial soft rot in lettuce, tomato, sweet potato, capsicum, potato, carrot, etc.; bacterial leaf spot in cucurbits; bacterial wilt in eggplant, tomato, and capsicum; bacterial blight in spring onions, coriander, and beetroot; and bacterial brown spot in beans and bacterial speck in tomato [32].

3.3 Fungal diseases

The fungal disease causes disastrous damages, as the spores of fungus easily infect other plants, with less infection time and high density spread due to rainwater or surface water, and wind for long distances. The phytotoxic compounds destroy the structure of plants and draw essential nutrients away from the plant. This pathogen infects foliar tissues of the plant which makes the complete loss of grain [33]. The fungus *Magnaporthe oryzae* creates the devastating effect, as it affects the most essential rice crop, which is a food source for one-half of the global population. The other severely damaging virus, commonly seen in crops, is *Botrytis cinerea*. This pathogen is also called gray mold, which infects around 200 plant varieties right from the seedling stage to the ripening of products such as cucurbits and strawberries [34]. The fungus *Puccinia graminis* f. sp. *tritici* creates stem rust, stripe rust, and leaf rust on wheat crops; among them, stem rust causes more damage to the crop [35]. The ascomycete *Fusarium graminearum* is a pathogen that creates more damage on cereal species and infects the floral tissues that result in economic losses. If the infected grain is further stored in a highly humid environment, the fungus growth will increase [36]. A soil-borne pathogen called *Fusarium oxysporum* Schlecht creates vascular browning, leaf epinasty, and wilt in crops such as melon, banana, cotton, and tomato [37]. *Blumeria graminis* creates powdery mildew in wheat and barley; *Mycosphaerella graminicola* creates a severe economic loss in wheat productivity; *Colletotrichum* causes preharvest spots and postharvest rots in several crops, especially banana; *Ustilago maydis* creates infections in corn yield; and *Melampsora Lini* affects flax and linseed production [38].

3.4 Oomycete diseases

Oomycete disease has almost similar characteristics as fungus diseases with some important differences as well. There is no chitin in their cell walls, very little sterol in their membranes, predominant diploid karyotype, and biflagellate zoospores. Some commonly occurring oomycetes in plants are *Phytophthora infestans* (creates blight in potatoes), *Hyaloperonospora arabidopsidis* (creates disastrous diseases in several essential crops such as grapes, maize, cucurbits, and lettuce), *Phytophthora ramorum* (most dangerous disease in oaks), *Phytophthora sojae* (causes root rot in soybean), *Phytophthora capsici* (in pepper, tomato, and cucurbits), *Plasmopara viticola* (in grape), *Phytophthora cinnamomic* (in tobacco), *Phytophthora parasitica* (in citrus), *Pythium ultimum* (in wheat, corn, and soybean), and *Albugo candida* (in Indian mustard) [39].

3.5 Nematode diseases

Nematode diseases which are of 17 orders, among them Tylenchida and Dorylaimida are plant parasites [40], which cause serious losses in crops. The latter one affects around 450 varieties of species. *Meloidogyne* affects many crops such as strawberry, potato, peanuts, onion, and carrot, [41, 42], and *Heterodera glycines* attacks soybean. Though it does not show direct symptoms, the production of crops is severely damaged.

3.6 Parasitic plant causing diseases

Striga and *Orobanche* are the most dangerous species among the 3,000 varieties of parasitic plants [43]. *Striga* species attacks cereals and legumes in the African continent which affects seed production resulting in huge loss in the cultivation of crops. Broomrapes are one of the parasitic weeds found in central Asia and the Mediterranean region that affects the roots of crops to acquire its required chlorophyll [44].

4 Existing technologies in plant disease detection

The identification of plant disease detection is categorized as direct and indirect methods. The former is used when there are many samples to be analyzed with molecular and serological methods to identify the various pathogens such as bacteria, fungus, and viruses. The latter one is used to identify the disease of the plant

through the change in the parameters such as morphology, temperature, transpiration rate, and volatile organic compounds (VOC).

4.1 Direct detection methods of plant diseases

The direct methods of plant disease detection are PCR, fluorescence in situ hybridization (FISH), enzyme-linked immunosorbent assay, immunofluorescence, and flow cytometry. Nowadays, PCR is widely used for plant pathogenic disease detection though it was initially used only for the identification of specific bacterial and viral diseases [45]. In addition, the advanced PCR technology called reverse-transcription PCR (RT-PCR) is also used for the identification of plant diseases due to its high sensitivity [46]. The multiplex PCR is also used for simultaneous identification of various RNA or DNA by operating in a single run, and the on-site and rapid detection of bacterial, virus, and fungal-based plant diseases are effectively detected by using real-time PCR. There are certain limitations in the PCR technology, as it majorly depends on DNA extraction efficacy and inhibitors, PCR buffer and it requires primer for initialization of DNA replication [47–49]. Another type of detection, called FISH, is used to detect bacterial diseases by targeting the gene from a sample with a combination of microscopy and DNA probe hybridization. It can also be used for the identification of other pathogenic diseases caused by a virus, fungus, and endosymbiotic bacteria. As the method is based on ribosomal RNA, it has high affinity, selectivity, and sensitivity. The limitation of this method is due to the autofluorescence materials that reduce its selectivity.

Immunofluorescence is an optical technique, which is used to identify the pathogenic disease in plant tissues. A fluorescent dye is conjugated to antibody into the plant tissue samples, and the distribution of the target molecule is visualized through microscopy. In combination with FISH, this method is used to identify specific pathogenic diseases. The sensitivity of the method is greatly affected by photobleaching which produces false-positive results [50, 51]. Flow cytometry is another optical technique based on a laser that is majorly used for cell counting and sorting, and detection of the biomarker for fast detection of cells. The cells are passed through an electronic apparatus through a liquid stream, which simultaneously measures various parameters. A laser beam is passed on the sample, which is further reflected by the sample, and its scattering effect or fluorescence is measured by the suitable detector resulting in efficient identification of plant diseases [52].

4.2 Indirect detection methods of plant diseases

The indirect detection of plant disease is based on stress and volatile profiling of plants. It is not only used for the detection of pathogenic diseases and biotic and abiotic stresses. One of the indirect methods of detection is thermography which is

based on the images of surface temperature differences of plant canopies and leaves. To capture the images, thermographic cameras are used, which captures the temperature difference of the plant leaves without external temperature interference. The method detects the diseases that create the loss of water in plants by measuring the temperature changes. As the method is extremely sensitive, the measurements are easily affected by environmental changes [53–56]. In fluorescence imaging, the incident light measures the plant leaf chlorophyll fluorescence, and any change in the fluorescence parameters are analyzed for the identification of plant pathogens. The leaf rust and powdery mildew infections are precisely identified by spatial and temporal variations of chlorophyll fluorescence [57].

Hyperspectral imaging is mostly applied for large-scale agriculture and plant phenotyping to determine the overall health status of plants. The techniques use a wide range of spectrum from 350 to 2,500 nm and are highly robust and rapid. The imaging camera collects the data in all X, Y, and Z axes, which are used to obtain a more accurate determination of plant health. It measures changes observed in the reflectance due to changes in the biophysical and biochemical characteristics in the plant caused by pathogenic diseases [58–60]. The plant disease detection is done by the volatile chemical signature profiling of the affected plants. The disease-causing pathogens damage the green leaves of the plants, and make them release a specific type of VOC during infection. The type of pathogenic diseases is accurately identified by profiling VOC. Gas chromatography (GC) is used for separation and analysis based on the retention time of various compounds in the stationary phase while carried by the mobile phase. The performance of GC is further enhanced by combining it with mass spectrometry [61].

4.3 Biosensor-based plant disease detection methods

The sample analytes detected using electrical, optical, magnetic, mechanical, and electrochemical are called biosensors. The detection is amplified by using the transducers made up of nanomaterials, and the selectivity is enhanced by using the bio-recognition elements (DNA, antibody, and enzymes). The nanoparticles such as metal, metal oxide nanoparticles, carbon nanotubes, and graphene are used for biosensor fabrication. The phenomenon such as encapsulation, adsorption, sophisticated combination, and covalent attachment is used for the immobilization of elements such as DNA, enzyme, and antibody. It provides a friendly platform for sensor design, high surface area, and high electronic conductivity [62]. The sensors detect the pathogens based on the reaction that takes place between the target analyte and the bio-recognition elements such as antibody and enzymes. Due to this affinity, the specificity of the sensor is greater compared to the biosensors based on nanoparticles [63].

Antibody-based sensors are extremely sensitive, rapidly detecting pathogens, and well suitable for real-time applications. It is applied for the detection of pathogens in water, seeds, and air in pre- and postharvesting of crops and fruits. The specific type of antibody is coupled with the transducer, where the specific antibody and the antigen binding are converted into a suitable signal for the measurement. The most widely used antibody-based biosensors are conductometric, amperometric, impedimetric, and potentiometric [64]. The surface plasma resonance, cantilever-based sensors, and quartz crystal microbalance are also the other types of affinity-based biosensing between antigen and antibody as illustrated in Figure 4.

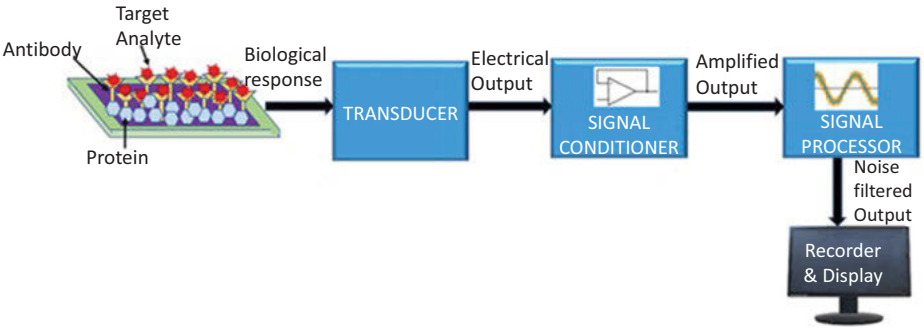


Figure 4: Schematic diagram of antibody-based biosensor.

The biosensor that detects pathogens with the use of affinity between the target analyte and the nucleic acid elements such as DNA or RNA is called a DNA/RNA-based affinity biosensor. It is most widely used for the detection of bacteria and fungi. The hybridization between the DNA probe and the DNA complementary analyte is measured by adopting the single-stranded DNA as a DNA probe on the electrodes as illustrated in Figure 5. The most widely used DNA-based biosensing methods are optical, strip type, electrochemical, and piezoelectric-based sensing [65, 66]. In enzymatic electrochemical biosensor-based disease, detection is done by using an enzyme as a biorecognition element. The specific enzymes for the target analyte are placed on the nanomaterial-based electrode. The bioelectrocatalytic reaction takes place between the target analyte and the electrode, which produces an electrical signal. The electrical signal is measured to obtain the quantitative information of the target analyte as illustrated in Figure 6.

The bacteriophage biosensors are based on the affinity between the bacteriophage and the target analyte. The bacteriophages are a virus, comprised of a DNA or RNA genome-encapsulated protein capsid, which infects the bacteria and multiplies within the bacteria and lyses them to propagate. The method offers high selectivity, sensitivity, and high thermostability, and is cheaper. The changes in the charge transfer reaction impedance at the interface are the measure of reaction

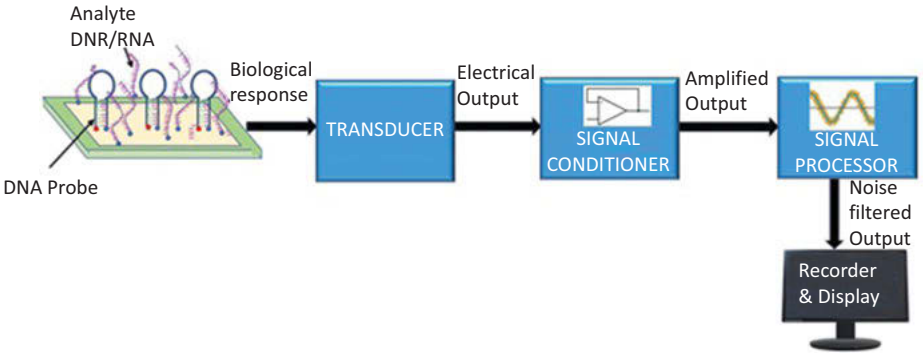


Figure 5: Schematic diagram of DNA/RNA-based biosensor.

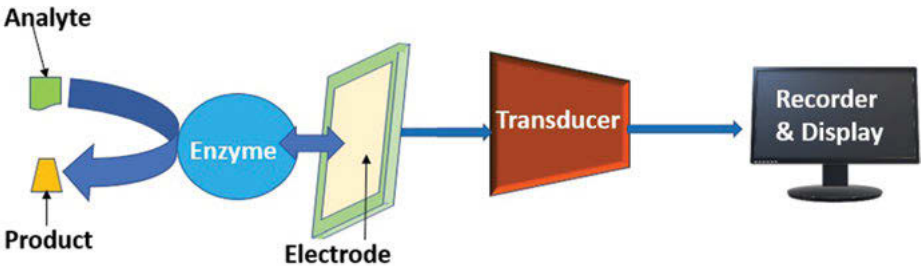


Figure 6: Schematic diagram of an enzymatic biosensor.

between the target analyte and bacteriophages [67]. Though the direct detection methods are well established, they are difficult in operation which need expertise, also time-consuming for analysis, and are not suitable for real-time detection. The indirect method of imaging techniques is suitable for on-site detection of large-scale agriculture, but they are very much prone to changes in environmental conditions. The biosensors provide extremely sensitive and selective detection, but still the parameters could be enhanced for early detection of diseases. Hence, the intelligent methods have to be implemented in combination with the existing biosensing technologies for the early detection of diseases with high sensitivity, selectivity, accuracy, reliability, and user friendly. Compared to other image recognition techniques, deep learning–based image recognition techniques are not required with specific feature extraction. It could find the proper features that will obtain robust and highly accurate contextual of the image features.

5 Overview of deep learning

Artificial neural networks (ANNs) are mathematical models, where the nodes are interconnected, as the human brain's neuron networks are connected. In the human brain, the neurons receive the signal transmitted from other neurons through synapse connection. The neuron is connected individually to the processing element, called the perceptron. In a network, the neuron accepts the input, processes it, and produces the output. The encoded electrical information is in the weights which are carried by the connection between the two neurons. The values of weight simulated by the electrical information help in learning, recognition, and prediction by creating the relationship between the networks. ANNs have been studied since the 1940s. By analyzing and summarizing the characteristics of neurons, McCulloch et al. proposed the McCulloch–Pitts (MP) model [68]. To understand the adaptation of cerebral neurons during the learning process, Hebb et al. suggested a cell assembly theory. The cell assembly theory influenced the creation of neural networks significantly [69]. The perception algorithm was then developed by Rosenblatt et al., in which a learning algorithm is a kind of binary classifier [70]. The adaptive linear element was proposed by Widrow and is a single-layer ANN based on the MP model. Unfortunately, Minsky and Papert pointed out the perception algorithm that has significant theoretical limitations and gave a pessimistic assessment of neural networks' prospects, causing the growth of neural networks to reach a nadir. In the early 1980s; however, Hopefield et al. suggested the *Hopfield network*. ANNs were resurrected as a result of *the Hopfield network* [71]. The Boltzmann computer was then proposed by Ackley et al. using the simulated annealing algorithm. Various shallow ML model approaches include support vector machine boosting, which was proposed one after another in the 1990s. ANNs have reached a new high due to the benefits of these approaches both in theory and in practice. ANNs reawakened interest in the scientific community after Ackley et al. proposed the idea of deep learning in the journal *Science* in 2006 [72].

Deep learning is a subset of ML, which is a branch of artificial intelligence (AI). In the last few decades, ML has been revolutionized incredibly in various fields, and the ANN is the subcategory of ML, which helps the evolution of deep learning. Due to the parallel graphics processing unit empowerment and largely available good quality of datasets contributed to the enhancement of deep learning applications, the deep learning models typically use hierarchical structures. Easy linear or nonlinear formulas may be used to convert the productivity of a lower layer to the involvement of a higher layer. These models can transform low-resolution data structure/features into high-quality image/abstract features. Deep models can be more powerful in feature representation than shallow ML models due to their unique characteristics. Traditional ML methods depend on the experience of the users, while deep learning methods rely on the data. As a result, we can conclude that deep learning methods have limited consumer demands. Computer efficiency

is steadily improving as computer technology advances. Meanwhile, knowledge is abounding on the Internet. These factors are providing a strong impetus for deep learning to evolve and become the dominant ML form [68, 70, 73].

Convolutional neural networks (CNNs) are a type of deep learning network, and are utilized for extracting the appropriate features automatically from the given images, which are further fed for classification done by the fully connected layers. The appropriately trained features play a vital role for classification as they mimic the vision of human experts. The main difference between conventional ML and deep learning is the specific feature extraction that should be done manually, and it depends on the user selection. In deep learning neural networks, particularly in a CNN, feature extraction is part of the learning process, which is automatically done as illustrated in Figure 7. If it is provided with sufficient training data, the deep learning neural system itself will identify the most suitable and useful features from the images, which could be extracted [74–76].

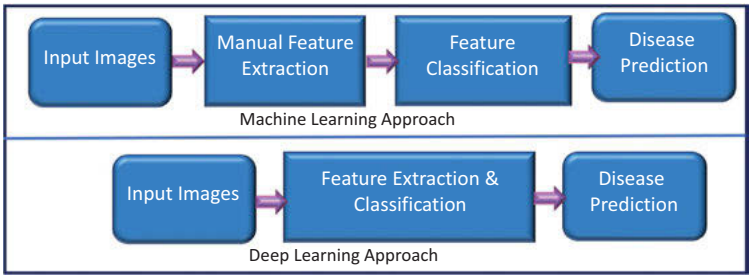


Figure 7: Difference between deep learning and machine learning operations.

CNN is a subcategory of ANN that takes the benefit of the spatial information of the given inputs. The network is first introduced by Fukushima in 1988 [77]. Due to computation hardware limitations for the training of datasets, it was not widely utilized. Later, the CNN was applied with a gradient-based learning technique, and the results were successful for the classification of handwritten digits. Recently, CNNs have become phenomenally successful in the applications of computer vision such as object identification, face recognition, and providing vision to robots and in self-driven cars. It has a standard architecture that consists of convolutional and pooling layers in the alternating sequence. The last layer is built with few numbers of fully connected layers, and a SoftMax classifier is present in the final layer as illustrated in Figure 8. The input volume is transformed into the output volume of neuron activation by the CNN toward the fully connected layers, which maps the given input data to a one-dimensional feature vector. Ultimately, CNN contains three basic layers such as (i) convolutional layers, (ii) pooling layers, and (iii) fully connected layers.

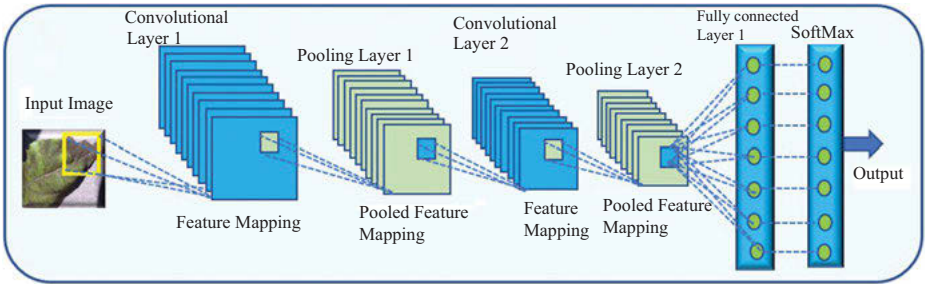


Figure 8: The basic structure of convolutional neural network (CNN).

5.1 Convolution layers

The maps of immediate features and various kernels along with the whole images are convolved by the convolution layer. Many works have published the convolutional operation as a substitute to the fully connected layers to reduce the learning time. The difference in the fully connected layer from the convolution layer is illustrated in Figure 9.

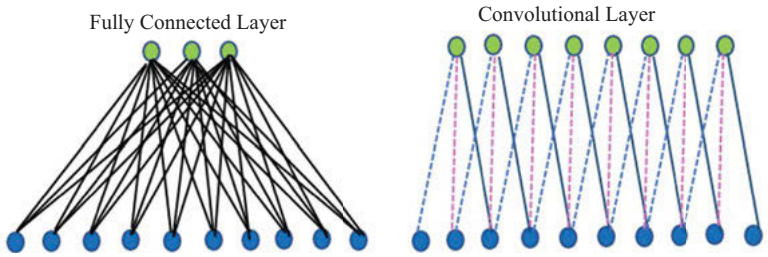


Figure 9: The structure of fully connected and convolutional layers of CNN.

5.2 Pooling layer

It deals with reducing the spatial dimensions of the convolution layer input volume which follows the pooling layer immediately and it does not affect the volume depth dimension. The process done by this layer is called downsampling, as there is a loss of information due to size, which leads to reduction. It is also beneficial to the network as it is forced to be trained only on meaningful features. It also reduces the computational overhead to the forthcoming layer of the CNN and works against overfitting. The most widely used strategies are average and max pooling.

5.3 Fully connected layers

After various convolution and pooling layers, the high-level reasoning is performed through fully connected layers. Neurons in this layer are completely connected to all activations of the previous layer. The activation is calculated by matrix multiplication with a bias offset. The two-dimensional feature maps are converted into a one-dimensional feature vector in the fully connected layer; further, it can be processed as a feature vector or fed for classification.

The most important task of deep learning in the agriculture sector in the early detection of plant infections. In India, most of the verifications are performed by hand, which makes it difficult to diagnose the disease and its kind. The difficulty in diagnosis has increased the significance of mechanized infection/disease recognition and prompted the creation of models/methods and/or systems that can more accurately diagnose the infection/disease. Most of the data are in the form of vulnerable to error images. Obtaining reliable information on disease signs is key to more accurately detect and diagnose diseases [78]. Identification and categorization of disease in the plants such as various spots/disfiguring of leaves can be considered as the primary source of disease knowledge using current computer vision technologies [79]. ML is a well-known deep learning method/technique for teaching computers to mimic human behavior in various fields of application. Machines respond by learning and using the same experience for the next to learn and apply using techniques. ML is a multidisciplinary field of study that has initiated a new research area, including agriculture science. The techniques can be used in a variety of fields of computation, allowing for the development of new algorithms. These algorithms are used to solve a variety of crop problems, such as early recognition of diseases and specific crop disease classification.

CNN is a deep learning technique, which has proven best in image/features recognition and reached great success in the agricultural field of science. Current methods can be best understood by monitoring, measuring, and analyzing vast agricultural data to resolve the issues. It is also essential to understand the technologies for short-term and long-term crop management to maintain sustainable ecosystems for large-scale crops. Big-data study is another significant analysis in deep learning model techniques. A deep learning model or method has three layers of information, and every layer contains neurons that connect to collected data structures, resulting in more intricate data knowledge and information. Deep learning models learn input features through a hierarchy of systematic neuron networks. Recent research has focused on evaluating deep learning models/methods for detecting plant infectious diseases using digital imagery, hyperspectral imaging, and data processing. Deep learning predicts that in the future, CNN will be the utmost widely recognized and convincing imaging model. As a result, hyperspectral data image analysis is a significant field of image processing study that holds a lot of promise. The importance of this chapter is to collect the published data to understand the importance of deep learning in the

agricultural science area. The data were collected and presented on how to manage productivity by monitoring plant health ahead of time in terms of climatic shifts, food protection, and sustainability during cultivation scenarios.

6 Significant applications of deep learning in disease detection

Candidate area collection, feature extraction, and classification are examples of conventional object detection methods. These manual feature extraction procedures are often expensive and time-consuming. Deep learning is capable of unsupervised feature learning. It can extract image features without the need for human interference, and also researchers are increasingly paying more attention to it. Deep learning is becoming more and more common in computer vision. Krizhevsky et al. achieved a breakthrough using CNN in ImageNet LSVRC 2012 and has made an excellent breakthrough in recent times [80]. Till date, the approach of combining three residual inception networks with one Inception-v4 has resulted in a top error of 3.08% for image recognition. Learned-Miller et al. proposed a deep learning approach that improved face recognition accuracy to about 87% [81]. At the moment, researchers at a Chinese university in Hong Kong have improved the accuracy of face recognition to over 99% [82].

Deep learning has noteworthy advancements in natural language processing (NLP), with numerous accomplishments in applications such as speech recognition, speech synthesis, and question-answering (QA). For a long time, conventional speech recognition systems relied heavily on Gaussian image models and hidden Markov models. However, these approaches are prone to disruptions from the outside world and are unable to cope with deep characteristics. The system's efficiency has improved significantly since it began using deep learning in speech recognition. In the Chinese speech test, Baidu's speech recognition device deep speech 2 has reduced the error rate to 3.7%. *Deepmind* released a new speech synthesis system called *WaveNet* at Google. *WaveNet* is a deep neural network capable of producing raw audio waveforms [83, 84]. *WaveNet* can produce more realistic sounds and music than other text-to-speech systems. *WaveNet* narrowed the difference between human and synthesized voices by over 50% in English and Chinese, according to *Deepmind*. QA is a common NLP research area that can provide a correct and concise response in natural language form for natural language problems. Watson's win on Jeopardy has shown that QA dependent on deep learning has its distinct advantage [85].

6.1 Deep learning applications in agriculture

The deep learning techniques are categorized as supervised, unsupervised, and partially supervised learning techniques. The supervised learning approach utilizes the labeled datasets, and the supervised deep learning technique includes input–output sets in a large volume. A cost function always evaluates the performance of the model with the help of an optimization algorithm. Partially supervised learning utilizes only partially labeled datasets, and the generative adversarial networks are one type of partially supervised deep learning network. Unsupervised learning technique operates data without the label. It learns the specific features to find the unknown relationship or the shape in the input data. Clustering and reducing dimensionality come under unsupervised learning techniques, which are depicted in Figure 10 [78, 85–88].

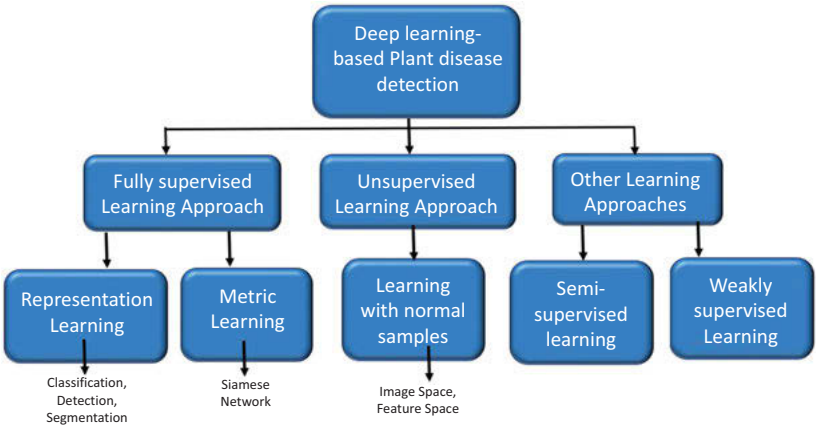


Figure 10: Different learning approaches for plant disease and pest detection.

Table 1 lists the eight appropriate works, representing the crop-associated research field, the deep learning models, and architectures implemented, and the overall efficiency was achieved based on the metrics employed. The data for training and testing was the same in all the cases mentioned in Table 1 [89]. In total, many areas have been identified in fruit counting, land cover classification, weed identification, crop-type classification, and plant recognition among the most common. Except for few authors, all of the literature was published in or after 2015, demonstrating how new and cutting-edge agriculture technology was updated [90, 91]. Most deep learning applications, such as obstacle detection and fruit counting, emphasize foreseeing future constraints, such as corn crop growth, and moisture content in the soil. Few works, on the other hand, address issues including weed detection, ground cover, soil analysis, livestock agriculture, obstacle detection, and weather prediction [92].

Table 1: Deep learning applications in agriculture field of research [89].

S no.	Task description	Dataset used	Deep learning model used	FW used	Preprocessing of data	Performance metric used	Value of metric used	Comparison with other technique
1	Filter the leaves of various plant species	The Flavia dataset contains 1,910 leaf images from 32 species, with at least 50 and up to 77 images per species	Author-defined CNN + RF classifier	Caffe	Feature extraction based on HoCS histograms, form and statistical attributes, normalized excessive green (NExG) vegetative index, white border doubling image size, segmentation	CA (classification accuracy)	97.13% ± 0.61%	RF classifier and feature extraction (shape and statistical features)
2	From healthy leaves, 13 different forms of plant disease identification	4,483 images in a database generated by the research group	CaffeNet CNN	Caffe	Cropping, square around the leaves to highlight the area of interest, resizing to 256 × 256 pixels, and image duplication elimination	CA	0.9632	Better results than SVM (no more details)
3	14 crop species and 26 diseases must be identified	A public dataset of 54,306 images of diseased and healthy plant leaves collected under controlled conditions is available from the PlantVillage database	AlexNet, GoogleNet CNNs	Caffe	Resized to 256 × 256 pixels, segmented, background information removed, color casts corrected	*F*1 score	0.9925	For methods that use hand-engineered features, there is a significant margin in standard benchmarks

4	In KSC, identify 13 different land-cover groups, while in Pavia, identify 9 different classes	Dataset 1 shows a mixed vegetation site over Kennedy Space Center (KSC) in Florida, and Dataset 2 shows an urban site over Pavia, Italy. Hyperspectral datasets are a form of hyperspectral data	PCA, autoencoder (AE), and logistic regression hybrid	The research group developed the dataset	Noise forced the removal of some bands	CA	0.9873	1.0% more precise than RBF-SVM
5	Classification of maize crops, wheat, soybeans, sugar beet, and sunflower	Landsat-8 and Sentinel-1A RS satellites captured 19 multitemporal scenes from a test site in Ukraine	Author-defined CNN	The research group developed the dataset	Calibration, multi-looking, speckle filtering (3 × 3 windows using the refined Lee algorithm), terrain correction, segmentation, and missing data restoration	CA	0.9462	Multilayer perceptron: 92.17%, RF: 88.1%
6	Recognize seven separate plant views: the whole plant, the branch, the flower, the fruit, the leaf, the stem, and the scans	There are 91,759 images in the LifeCLEF 2015 plant dataset, which is divided into 13,887 plant observations. Each observation depicts the plant from different perspectives: the entire plant, a leaf branch, a fruit, a stem scan, and a flower	AlexNet CNN	Caffe	Flowers and leaf scans have a higher level of accuracy than the other views	LC (LifeCLEF)	0.486	KNN, dense SIFT, and a Gaussian mixture model are all 20% worse than local descriptors for representing images

(continued)

Table 1 (continued)

S no.	Task description	Dataset used	Deep learning model used	FW used	Preprocessing of data	Performance metric used	Value of metric used	Comparison with other technique
7	Predict how many tomatoes are in the image	The research group created 24,000 synthetic photographs	Modified Inception-ResNet CNN	TensorFlow	A Gaussian filter has blurred synthetic images	RMSE (root mean square error), RFC (ratio of total fruits counted)	On real images 91.2% RFC and 1.17 of RMSE, and on synthetic images, 93.1% RFC, 2.53 of RMSE	ABT: 66.18% of RFC, and 13.52 of RMSE
8	Sort the 91 different types of weed seeds into categories	Dataset of 3,980 images containing 91 types of weed seeds	PCANet + LMC classifiers	Developed by the authors	Image filter extraction using a bank of PCA filters, binarization, and counting histograms	CA	0.9098	LMC classifiers + manual function extraction techniques: 64.82%

6.2 Classification of diseases with plant features

Owing to the importance of food safety and demand in production, the health status of the plants should be continuously monitored and prevented from pathogenic diseases. Hence, it is particularly important for the farmers to timely detect the diseases and recognizes the pest to prevent the plants from damages. The early detection of pathogenic diseases in rice and wheat crops plays a vital role to prevent the crops from diseases in regions like India, where they are considered as the main food source. The automatic detection of diseases in an early stage of disease done by deep learning neural networks is published by several researchers as depicted in Figure 11. In [93], a new identification method for rice disease based on deep learning CNN techniques is discussed. About 500 images of healthy and diseased leaves of rice and stems are captured to train CNN for the early identification of 10 diseases.

The identification of diseases using the CNN-based neuron model is successfully classified with an accuracy of 95.48%, which is higher than the traditional ML technique. Another work describes the automatic detection of wheat disease detection and diagnosis based on a weak supervised deep learning approach. Two types of architectures such as VGG-FCN (fully convolutional network)-VD16 and VGG-FCN-S are used, which identify the diseases with the accuracy of 97.95% and 95.12%. It can also be done in a real-time mobile app to make the disease identification farmer-friendly [94]. For the identification of diseases in the tomato, three main types of neural networks are considered such as faster region-based fully CNN, region-based FCN, and single-shot multibox detector. The proposed method was implemented in identification to increase the accuracy and decrease the false-positive results. Totally 5,000 images are used to identify 5 types of diseases in tomato plants and 86% accuracy is achieved [95]. Eight kinds of diseases in maize are identified and classified by GoogLeNet with 98.9% and cifar10 with 98.8%. Two improved methods are utilized for training and testing of different kinds of images of maize leaves which ultimately enhances the accuracy of model training and classification [96].

6.3 Deep learning methodology

Deep learning can be developed with a variety of algorithms for disease identification, but the basic step is to choose the data related to the disease identification and to select the appropriate features to examine. As illustrated in Figure 11(a), the dataset is prepared with various images of diseased and healthy plants at different parts of the specimen. The captured images are further processed with appropriate image processing techniques, normalized, and standardized. The disease dataset is used to train the deep learning model as explained in Section 5, the so-called training data, to train the model with previous conditions captured from the plant and the agricultural environment related to diseases. Following the training, the testing

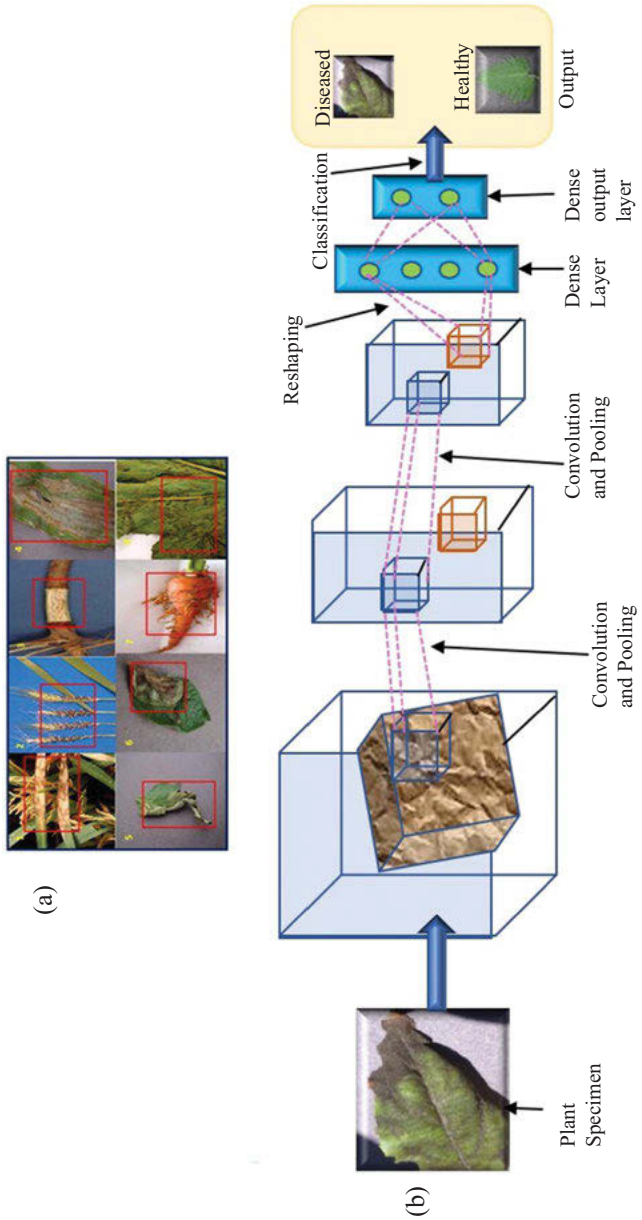


Figure 11: (a) Diseased plant images of (1) rice, (2) wheat, (3) cotton, (4) maize corn, (5) tomato, (6) potato, (7) carrot, and (8) pumpkin [97]. (b) Three-dimensional convolutional neural network for plant disease detection.

is done with the same model with another dataset called test data. Among the collected sample data from plant parameters, the major part is separated for training data and the remaining part is kept for testing purposes. The algorithms for learning can be either supervised or unsupervised learning. In a supervised learning algorithm, the training dataset trains the model and maps it into the known results. In unsupervised learning, the training dataset trains the model, and further, the model is validated with unknown patterns of data. To predict and classify the results based on the image patterns and for the decision made by deep learning models, data visualization tools are used [98–100].

7 Challenges and future directions in the plant disease detection

The CNN performance has been working tremendously for the past few years in feature extraction and image classification to identify the plant disease. But there are also certain challenges involved in implementing the CNN in plant disease identification. The accuracy of the model is reduced by around 30% when the images used in testing are a bit different from the trained images; thus, a most diverse set of data should be used for training to improve the accuracy. Most of the current identification methods train the model with the images of plant leaves with a certain condition, fixed background, and at the top view of the leaves. It is not necessary that the sensor will also capture the images in the same condition and background while applying them for real-time identification in the field. Moreover, the disease might not necessarily be in the leaves; it can affect any part of the plant such as stem and root, and at any side of the plant parts. Thus, the images used for training should include all parts of the diseased plant from a different perspective.

The deep learning methods are mostly used for various computation tasks, and plant disease and pest identification are a specific application of deep learning networks and there are only a few plant diseases, and pest samples are available in the database. These self-collected data are exceedingly small compared to the open standard libraries, and sometimes the diseases have a low incidence, and the cost of image acquisition is extremely high. Data amplification in the training of deep learning networks is significant in plant disease and pest detection. To acquire more samples of plant diseases and pests, various image processing operations such as mirroring, rotation, wrapping, and filtering are used. Sometimes, the lesions are small in size, which is ignored by the multiple downsampling process in the deep learning feature extraction network. The method occasionally produces false-positive results due to the noise and complexity of the background, particularly in the low-resolution images. Most of the images of plant diseases and pests are captured in indoor light which is quite different from the real natural light. The

dynamic changes based on the surrounding lights of the camera are also limited, which may cause discoloration of images.

Another difficulty in the visual identification is due to the difference in the focusing angle and the distance kept during the collection of images. Comparing with the conventional methods, deep learning neural networks show better performance, but they are extremely complicated in computation. To find the plant diseases more accurately, the network must be trained with a large sample of data, which will increase the complexity and time of computation; thus, it cannot meet the real-time needs. But to increase the speed, the complexity of the computation must be reduced, which will create a negative impact on the accuracy. Thus, the research should be focused on developing the algorithm with less computational complexity without compromising the accuracy. The detection with deep learning is based on three main tasks in precision agriculture, such as data labeling, training the neuron model, and inference of the model. Thus, more attention must be paid to develop an effective algorithm to identify the plant disease and pest with high speed and greater accuracy.

8 Conclusions

As agriculture is the backbone of many countries, especially India, damages in crops and other plants due to pathogenic diseases greatly affect the crop yield which results in reduction of food production quality and quantity, which will not meet the demand. In the early detection of plant diseases and to prevent it from damage, the image identification tools and neural network-based classification algorithms play a vital role with better accuracy, compared to the other conventional identification methods such as direct detection, indirect detection, and biosensing-based identification. The deep learning CNNs had broadly established prospects and potential in the feature extraction of images and classification of diseases. But still, there is certain limitation involved in the real-time implementation in the crop field of large-scale agriculture. Many of the published researches are limited to the laboratory which can only identify the diseases and pests only if it is tested in the laboratory environment. The images of plant diseases during training can vary from region to region, which is not universal.

Most of the images are taken in visible light, but in the field, other regions of an electromagnetic wave may also have certain information. Future research includes the multispectral fusion of information to identify plant diseases and pests, and it needs to cover the wide range of identification of agricultural fields. In some cases, the earlier stages of diseases do not show any obvious changes or symptoms, so the early identification by visual observation becomes difficult and significant to prevent the plant from damages. In the future, it is important to combine the meteorological data and plant-

related data such as the surrounding temperature and the environment humidity for the detection. The images of healthy and diseased plant data are visually observed and collected manually. As the accuracy of a CNN with supervised learning is based on the number of data samples fed to the network during training, it needs a lot of manpower and time; thus, unsupervised learning of CNN is preferred. At the same time, it needs a lot of memory, and the training is time-consuming, which limits this technique to be deployed on mobile platforms. Future studies should focus on reducing the complexity involved in the implementation of these techniques with greater accuracy.

The establishment of field detection and diagnosis of plant disease will help improve the accuracy and effectiveness in the identification of plant and pest detection. Thus, in the future, it is important to move from image extraction at surface level to the occurrence mechanism identification of plant disease and pest, right from the simple laboratory environment to the real-time application. In summary, with the evolution of different tools and techniques in AI, much research is being raised for the identification of plant diseases and pests in farms and moved from simple image processing and ML methods to deep learning neural networks. Though it has a long way to cover from laboratory research to real-time application, this technology has great potential and application value. The implementation of this technology effectively in the early detection of plant disease can be accomplished by the joint effort taken by the expertise of various relevant fields and the integration of their experience and knowledge to bring the plant disease and pest detection based on deep learning to the next level.

References

[1] Food and Agriculture Organization of the United Nations, Plant Health and Food Security, 2017.

[2] H. E. Pattee, *Evaluation of Quality of Fruits and Vegetables*, Van Nostrand Reinhold Co, New York, 1985.

[3] D. Bigioi and I. Dobre, The importance of quality management for the agri-food products, *Journal of Environmental Protection and Ecology*, 8(3), 688–700, 2007.

[4] S. Navulur and M. N. Giriprasad, Agricultural management through wireless sensors and Internet of things, *International Journal of Electrical and Computer Engineering*, 7(6), 3492–3499, 2017.

[5] S. Roy and S. Bandyopadhyay. A test-bed on real-time monitoring of agricultural parameters using wireless sensor networks for precision agriculture, in *First international conference on intelligent infrastructure the 47th annual national convention at computer society of India CSI*, 1–7, 2013.

[6] Z. A. S. Aqeel-ur-Rehman, A. Z. Abbasi, and N. Islam, A review of wireless sensors and networks applications in agriculture, *Computer Standards & Interfaces*, 36, 263–270, 2014.

[7] R. Nisheeth, B. Lakshmi, and R. K. Kodali, WSN sensors for precision agriculture, in *IEEE Region 10 Symposium*, 651–656, 2014.

[8] G. Pajares, Advances in sensors applied to agriculture and forestry, *Sensors*, 11(9), 8930–8932, 2011.
[9] P. M. R. Dampney, J. A. King, R. M. Lark, H. C. Wheeler, R. I. Bradley, and T. Mayr, Non-intrusive sensors for measuring soil physical properties, Managing soil and roots for profitable production, in *HGCA conference*, 61–67, 2004.
[10] M. Schirrmann, R. Gebbers, and E. Kramer, Soil pH mapping with an on-the-go sensor, *Sensors*, 11(1), 573–598, 2011.
[11] T. N. Limited, Installation and Operation Guide for WeedSeeker Automatic Spot Spray System.
[12] R. K. Kodali and N. N. Sarma, Experimental WSN setup using XMesh networking protocol, in *Advanced Electronic Systems (ICAES)*, *2013 International Conference*, 267–271, 2013.
[13] M. C. W. Sensor, Meteocontrol energy and weather services. www.meteocontrol.com.
[14] A. Baggio, Wireless sensor networks in precision agriculture, 2005.
[15] C. C. R. Morais, B. Cunha, M. Cordeiro, C. Serodio, and P. Salgado, Solar data acquisition wireless network for agricultural applications, *The 19th IEEE Convention of Electrical and Electronics Engineers in Israel*, 527–530, 1996.
[16] M. R. R. Morais, M. A. Fernandes, S. G. Matos, C. Ser Dio, and P. Ferreira, A ZigBee multipowered wireless acquisition device for remote sensing applications in precision viticulture, *Computers and Electronics in Agriculture*, 62(2), 94–106, 2008.
[17] A. Willig, Wireless sensor networks: Concept, challenges and approaches, *E I Elektrotechnik Und Informationstechnik*, 123(6), 224–231, 2006.
[18] A. G. Anastasi, M. Conti, and M. Di Francesco, Energy conservation in wireless sensor networks: A survey, *Ad Hoc Networks*, 7(3), 537–568, 2009.
[19] Y. X. S. Ozdemir, Secure data aggregation in wireless sensor networks: A comprehensive overview, *Computer Network*, 53(12), 2022–2037, 2009.
[20] N. S. Z. A. Aqeel-ur-rehman and N. A. Shaikh, An integrated framework to develop context-aware sensor grid for agriculture, *Australian Journal of Basic and Applied Sciences*, 4(5), 922–931, 2010.
[21] A. C. C. Goumopoulos and A. D. Kameas, An ontology-driven system architecture for precision agriculture applications, *International Journal of Metadata, Semantics and Ontologies*, 4(1), 72–84, 2009.
[22] P. Christou, The potential of genetically enhanced plants to address food insecurity, *Nutrition Research Reviews*, 17, 23–42, 2004.
[23] FAO, The state of food insecurity in the world (SOFI), Rome, Italy: FAO, UN, 2000. [Online]. Available: www.fao.org/FOCUS/E/SOFI00/sofi001-.
[24] C. T. Rogerson and E. C. Large, *The Advance of the Fungi*, 57(6). London, UK: Jonathan Cape, 1965.
[25] S. Y. Padmanabhan, The great Bengal famine, *Annual Review of Phytopathology*, 11, 11–26, 1973.
[26] A. J. Ullustrup, The impact of the southern corn leaf blight epidemics of 1970–71, *Annual Review of Phytopathology*, 10, 37–50, 1972.
[27] G. D. F. Karen-Beth, G. Scholthof, S. Adkins, H. K. Czosnek, P. Palukaitis, E. Jacquot, T. Hohn, B. Hohn, K. Saunders, T. Candresse, P. Ahlquist, and C. Hemenway, Top 10 plant viruses in molecular plant pathology, *Molecular Plant Pathology*, 12(9), 938–954, 2011.
[28] J. De Ley, F. Leyns, M. Decleene, and J. G. Swings, The host range of the genus Xanthomonas, *The Botanical Review; Interpreting Botanical Progress*, 50, 308–356, 1984.
[29] M. Joseph, S. Gopalakrishnan, R. K. Sharma, V. P. Singh, and A. K. Singh, Combining bacterial blight resistance and Basmati quality characteristics by phenotypic and molecular

marker-assisted selection in rice, *Molecular Breeding: New Strategies in Plant Improvement*, 13, 377–387, 2004.
[30] G. Blomme, M. Dita, K. S. Jacobsen, L. P. Vicente, A. Molina, W. Ocimati, and S. Poussier, Bacterial diseases of bananas and enset: Current state of knowledge and integrated approaches toward sustainable management, *Frontiers of Plant Science*, 8(1290), 1–25, 2017.
[31] K. T. Yuliar and Y. A. Nion, Recent trends in control methods for bacterial wilt diseases caused by Ralstonia solanacearum, *Microbes and Environments/JSME*, 30(1), 1–11, 2015.
[32] Overview: Pests, diseases and disorders, *Biosecurity & Agrichemical*. https://ausveg.com.au/biosecurity-agrichemical/crop-protection/overview-pests-diseases-disorders/.
[33] C. Struck, Infection strategies of plant parasitic fungi, In: *The Epidemiology of Plant Diseases*, Springer, Dordrecht, 117–137, 2006.
[34] P. Van Baarlen, E. J. Woltering, M. Staats, and J. A. van Kan, Histochemical and genetic analysis of host and non-host interactions of Arabidopsis with three Botrytis species: An important role for cell death control, *Molecular Plant Pathology*, 8, 41–54, 2007.
[35] J. C. Zadoks, Cereal rusts, dogs and stars in antiquity, *Cereal Rusts Bull*, 13, 1–10, 1985.
[36] N. Magan, D. Aldred, K. Mylona, and R. J. Lambert, Limiting mycotoxins in stored wheat, *Food Additives and Contaminants*, 27, 644–650, 2010.
[37] C. B. Michielse and M. Rep, Pathogen profile update: Fusarium oxysporum, *Molecular Plant Pathology*, 10, 311–324, 2009.
[38] G. D. F. Ralph Dean, J. A. L. Van Kan, Z. A. Pretorius, K. E. Hammond-Kosack, A. D. Pietro, P. D. Spanu, J. J. Rudd, M. Dickman, R. Kahmann, and J. Ellis, The top 10 fungal pathogens in molecular plant pathology, *Molecular Plant Pathology*, 13(4), 414–430, 2012.
[39] S. Kamoun, O. Furzer, J. D. G. Jones, H. S. Judelson, G. S. Ali, R. J. Dalio, S. G. Roy, L. Schena, A. Zambounis, F. Panabières, D. Cahill, M. Ruocco, A. Figueiredo, X.-R. Chen, and J. Hul, The top 10 oomycete pathogens in molecular plant pathology, *Molecular Plant Pathology*, 16(4), 413–434, 2015.
[40] J. B. Goodey and M. T. Franklin, *The Nematode Parasites of Plants Catalogued under Their Hosts*, 3rd edn, Commonwealth Agricultural Bureaux, Farnham Royal, UK, 1965.
[41] J. M. Evans, K. Trudgill, and D. L. Webster, *Plant Parasitic Nematodes in Temperate Agriculture*, CAB International, 1993.
[42] R. E. Gaunt, The relationship between plant disease severity and yield, *Annual Review of Phytopathology*, 33, 119–144, 1995.
[43] D. M. Joel, The long-term approach to parasitic weeds control: Manipulation of specific developmental mechanisms of the parasite, *Crop Protection (Guildford, Surrey)*, 19, 753–758, 2000.
[44] S. Gr, The physiology and biochemistry of parasitic angiosperms, *Annual Review of Plant Physiology and Plant Molecular Biology*, 41, 127–151, 1990.
[45] J. Cai, H. Caswell, and J. Prescott, Nonculture molecular techniques for diagnosis of bacterial disease in animals a diagnostic laboratory perspective, *Veterinary Pathology*, 51, 341–350, 2014.
[46] M. López, M. M. Bertolini, E. Olmos, A. Caruso, P. Corris, M. T. Llop, P. Renyalver, and R. Cambra, Innovative tools for detection of plant pathogenic viruses and bacteria, *International Microbiology: The Official Journal of the Spanish Society for Microbiology*, 6, 233–243, 2003.
[47] B. P. H. J. Lievens, B. Brouwer, M. Vanachter, A. C. R. C. Cammue, and B. P. A. Thomma, Real-time PCR for detection and quantification of fungal and oomycete tomato pathogens in plant and soil samples, *Plant Science: An International Journal of Experimental Plant Biology*, 171, 155–165, 2006.

[48] R. D. Schaad and N. W. Frederick, Real-time PCR and its application for rapid plant disease diagnostics, *Canadian Journal of Plant Pathology*, 24, 250–258, 2002.
[49] J. M. Van der Wolf, J. R. C. M. van Bechhoven, P. J. M. Bonants, and C. D. Schoen, New technologies for sensitive and specific routine detection of plant pathogenic bacteria, In: *Plant Pathogenic Bacteria*, Springer, Berlin, Germany, 2001.
[50] E. Ward, S. J. Foster, B. A. Fraaije, and H. A. Mccartney, Plant pathogen diagnostics: Immunological and nucleic acid-based approaches, *The Annals of Applied Biology*, 145, 1–16, 2004.
[51] A. D. L. Wullings, B. A. VanBeuningen, A. R. Janse, and J. D. Akkermans, Detection of Ralstonia solanacearum, which causes brown rot of potato, by fluorescent in situ hybridization with 23S rRNA-targeted probes, *Applied and Environmental Microbiology*, 64, 4546–4554, 1998.
[52] R. W. Chitarra and L. G. van Den Bulk, The application of flow cytometry and fluorescent probe technology for detection and assessment of viability of plant pathogenic bacteria, *European Journal of Plant Pathology / European Foundation for Plant Pathology*, 109, 407–417, 2003.
[53] D. Chaerle, L. Leinonen, I. Jones, and H. G. Van Der Straeten, monitoring and screening plant populations with combined thermal and chlorophyll fluorescence imaging, *Journal of Experimental Botany*, 58, 773–784, 2007.
[54] E. C. Lindenthal, M. Steiner, U. Dehne, and H. W. Oerke, Effect of downy mildew development on transpiration of cucumber leaves visualized by digital infrared thermography, *Phytopathology*, 95, 233–240, 2005.
[55] M. Oerke, E. Steiner, U. Dehne, and H. W. Lindenthal, Thermal imaging of cucumber leaves affected by downy mildew and environmental conditions, *Journal of Experimental Botany*, 57 (9), 2121–2132, 2006.
[56] U. Oerke, E.-C. Fröhling, and P. Steiner, Thermographic assessment of scab disease on apple leaves, *Precision Agriculture*, 12, 699–715, 2011.
[57] B. Stoll, M. Schultz, H. R. Baecker, and G. Berkelmann-Loehnertz, Early pathogen detection under different water status and the assessment of spray application in vineyards through the use of thermal imagery, *Precision Agriculture*, 9, 407–417, 2008.
[58] G. Bürling, K. Hunsche, and M. Noga, Use of blue-green and chlorophyll fluorescence measurements for differentiation between nitrogen deficiency and pathogen infection in winter wheat, *Journal of Plant Physiology*, 168, 1641–1648, 2011.
[59] P. Delalieux, S. van Aardt, J. Keulemans, W. Schrevens, and E. Coppin, Detection of biotic stress (Venturia inaequalis) in apple trees using hyperspectral data: Non-parametric statistical approaches and physiological implications, *European Journal of Agronomy*, 27, 130–143, 2007.
[60] Y. Kobayashi, T. Kanda, E. Kitada, K. Ishiguro, and K. Torigoe, Detection of rice panicle blast with multispectral radiometer and the potential of using airborne multispectral scanners, *Phytopathology*, 91, 316–323, 2001.
[61] S. L. Zhang, M. Qin, Z. Liu, and X. Ustin, Detection of stress in tomatoes induced by late blight disease in California, USA, using hyperspectral remote sensing, *International Journal of Applied Earth Observation and Geoinformation*, 4, 295–310, 2003.
[62] R. P. Fang, Y. Umasankar, and Y. Ramasamy, Electrochemical detection of p-ethylguaiacol, a fungi infected fruit volatile using metal oxide nanoparticles, *Analyst*, 139, 3804–3810, 2014.
[63] I. Shipway, A. N. Katz, and E. Willner, Nanoparticle arrays on surfaces for electronic, optical, and sensor applications, *ChemPhysChem*, 1, 18–52, 2000.
[64] R. M. Sadanandom and A. Napier, Biosensors in plants, *Current Opinion in Plant Biology*, 13, 736–743, 2010.

[65] A. F. Skottrup, P. D. Nicolaisen, and M. Justesen, Towards on-site pathogen detection using antibody-based sensors, *Biosensors & Bioelectronics*, 24, 339–348, 2008.
[66] M. Nugaeva, N. Gfeller, K. Y. Backmann, N. Duggelin, M. Lang, H. P. Guntherodt, and H.-J. Hegner, An antibody-sensitized microfabricated cantilever for the growth detection of Aspergillus niger spores, *Microscopy and Microanalysis*, 13, 13–17, 2007.
[67] D. Mc Grath and S. van Sinderen, *Bacteriophage: Genetics and Molecular Biology*, Horizon Scientific Press, Norfolk, UK, 2007.
[68] W. S. McCulloch, A logical calculus of the ideas immanent in nervous activity, *The Bulletin of Mathematical Biophysics*, 5, 115–133, 1943.
[69] D. O. Hebb, The organization of behavior: A neuropsychological theory. New York: John Wiley and Sons, Inc., 1949. 335 p. $4.00, *Science Education*, 34(5), 336–337, Dec 1950, doi: 10.1002/sce.37303405110.
[70] F. F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain, *Psychological Review*, 6, 386–408, 65AD, 1958.
[71] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proceedings of the National Academy of Sciences*, 79(8), 2554–2558, Apr 1982, doi: 10.1073/pnas.79.8.2554.
[72] D. Ackley and G. Hinton, A learning algorithm for Boltzmann machines, *Cognitive Science*, 9, 147–169, 1985.
[73] H. Hornik, K. Stinchcombe, and M. White, Multilayer feedforward networks are universal approximators, *Neural Network*, 2(5), 359366, 1989.
[74] G. Lecun, Y. Bengio, and G. Hinton, Deep learning, *Nature*, 521(7553), 436–444, 2015.
[75] S. H. Lee, C. S. Chan, P. Wilkin, and P. Remagnino, Deep-plant: Plant identification with convolutional neural networks, in *Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP)*, 452–456, 2015.
[76] J. Schmidhuber, Deep learning in neural networks: An overview, *Neural Computing and Applications*, 61, 85–117, 2015.
[77] J. Gua, et al, Recent advances in convolutional neural networks, *Pattern Recognition*, 77, 354–377, 2018.
[78] M. Juncheng, L. Xinxing, W. Haojie, C. Yingyi, and F. Zetian, Monitoring video capture system for identification of greenhouse vegetable diseases, *Transactions of the Chinese Society of Agricultural*, 46, 282–287, 2015.
[79] J. G. A. Barbedo, A review on the main challenges in automatic plant disease identification based on visible range images, *Biosystems Engineering*, 144, 52–60, Apr 2016, doi: 10.1016/j.biosystemseng.2016.01.017.
[80] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet classification with deep convolutional neural networks, in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, 1097–1105, May 2012.
[81] G. B. Huang, H. Lee, and E. Learned-Miller, Learning Hierarchical Representations for Face Verification with Convolutional Deep Belief Networks, 2012, doi: 318_P3A-07.
[82] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2892–2900, 2015.
[83] O. A. van Den, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, and A. Graves, WaveNet: A generative model for raw audio, *Computer Vision and Pattern Recognition*, 2016.
[84] D. Ferrucci, A. Levas, S. Bagchi, and D. Gondek, Watson: Beyond Jeopardy!, *Artificial Intelligence*, 199(200), 93–105, 2013.
[85] K. Fukushima, Neocognitron: A hierarchical neural network capable of visual pattern recognition, *Neural Networks*, 1, 119–130, 1988.

[86] J. G. A. Barbedo, A review on the main challenges in automatic plant disease identification based on visible range images, *Biosystems Engineering*, 144, 52–60, 2016.
[87] Y. Sun, X. Wang, and X. Tang, Deeply learned face representations are sparse, selective, and robust, Dec. 2014.
[88] I. Goodfellow, J. Pouget-Abadie, M. Mirza, Y. B. David Warde-Farley, S. Ozair, and A. Courville, Generative adversarial networks, *Communications of the ACM*, 63, 139–144, 2020.
[89] A. Kamilaris, Deep learning in agriculture: A survey, *Computers and Electronics in Agriculture*, 147, 70–90, 2018.
[90] T. G. M. Demmers, Y. Cao, S. Gauss, J. C. Lowe, and D. J. Parsons, Neural predictive control of broiler chicken growth, *IFAC Proc*, 43, 311–316, 2010.
[91] Y. Chen, Z. Lin, X. Zhao, and G. Wang, Deep learning-based classification of hyperspectral data, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7, 2094–2107, 2014.
[92] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, and T. Perez, DeepFruits: A fruit detection system using deep neural networks, *Sensors*, 16, 1222, 2016.
[93] Y. Z. Y. Lu and Y. Shujuan, Identification of rice diseases using deep convolutional neural networks, *Neurocomputing*, 267, 378–384, 2017.
[94] J. Lu, J. Hu, G. Zhao, F. Mei, and C. Zhang, An in-field automatic wheat disease diagnosis system, *Computers and Electronics in Agriculture*, 142, 369–379, 2017.
[95] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition, *Sensors*, 17(9), 2022, 2017.
[96] G. Zhang, Foodborne pathogenic bacteria detection: An evaluation of current and developing methods, *Meducator*, 1, 15, 2013.
[97] Arkansas Plant Diseases Database. https://www.uaex.edu/yard-garden/resource-library/diseases/.
[98] M. A. Disha Garg, Deep learning and IoT for agricultural applications, In: *Internet of Things (IoT)*, Springer, Cham, 273–284, 2020.
[99] P. Sharma and Y. P. S. Berwal, Performance analysis of deep learning CNN models for disease detection in plants using image segmentation, *Information Processing in Agriculture*, 7(4), 566–574, 2020.
[100] J. Xiong, D. Yu, S. Liu, and S. X. W. Lei, A review of plant phenotypic image recognition technology based on deep learning, *Electronics*, 10(81), 1–19, 2021.

Chandrasinh Parmar, Nishith Kotak, Vishal Sorathiya,
Shobhit K. Patel

A comprehensive study of plant pest and disease detection using different computer vision techniques

Abstract: There are a vast variety of crops, land characteristics, fertilizers, and thereby different ranges and extents of the diseases, which need an ensemble method to detect and cure these crop diseases. The research fraternity is striving more to get a better solution for curing the crop disease which will thereby increase the yield of the crop production. With the blessings of machine vision and its supportive devices, a farmer in any region may get information in the early stage of the disease and can save his/her crop before it spreads further. For a huge farm, it is difficult for the farmers to analyze the crop at each and every place manually in time. Creating an ad hoc-type sensor-based network, which monitors the soil condition, atmospheric condition, and other features of the crop from different parts of the farm, will solve the uncertain troubles faced by the farmers. Using this type of cluster-head network, the computation can be done at the master-driven side using high-speed computation. An ensemble approach is required to be developed that results in cumulative decision to classify in between the healthy and infected crop. The identification will help farmers to detect disease from its symptoms to take preventive measures. This chapter provides details of various techniques for classification of diseases in plants and the fundamental theory of detecting pests and diseases through various machine learning algorithms, the current technology available in the market. The authors have majorly focused on studies of fundamentals, current trends, and future scopes of disease detection in crops using various machine vision techniques. The chapter also showcases the future scope of machine vision in the agriculture industry.

Keywords: machine vision, plant disease detection, agriculture, artificial intelligence, computer vision, image processing

Chandrasinh Parmar, Department of Information and Communication Technology, Marwadi University, Rajkot, Gujarat, e-mail: chandrasinh.parmar@marwadieducation.edu.in
Nishith Kotak, Department of Information and Communication Technology, Marwadi University, Rajkot, Gujarat, e-mail: nishith.kotak@marwadieducation.edu.in
Vishal Sorathiya, Department of Information and Communication Technology, Marwadi University, Rajkot, Gujarat, e-mail: vishal.sorathiya@marwadieducation.edu.in
Shobhit K. Patel, Department of Computer Engineering, Marwadi University, Rajkot, Gujarat, e-mail: shobhitkumar.patel@marwadieducation.edu.in

https://doi.org/10.1515/9783110734652-003

1 Introduction

Agriculture is among the most vital economic sectors of the world, making it an important source for economic growth. In India and many countries, the agriculture sector plays a major role in contributing toward the gross domestic product growth. Types of crops are being selected by the farmers based on various conditions like weather, economic value, and soil type. Generally, the different crops are greatly affected by various types of plant diseases. It is very essential to look for the new methods to fulfill the requirement of crop production that is generated because of large population growth, weather uncertainty, and political uncertainty. In agricultural sector, plant and crop diseases have a major effect on the crop production. Hence, it is very essential to keep the crop and plants in dirt-free and healthy condition. Crop growth monitoring is inevitable for revenue generation and thereby profit maximization. The different types of diseases in crop and plants require various types of solution to achieve high yield production. It is essential to look for new research domain that creates high productivity innovations which will be more effective and accurate. Nowadays, technological advancement makes possible to provide solution for agricultural growth. It is possible to provide the appropriate decision for high farming production by collecting information and data. Just as human beings suffering from deadly ailments, crops also suffer from diseases, which on detection in the early stage might prevent further damage to the crop. Majorly, the crop diseases are contagious, which spread through the contact of leaves, roots, or even with sharing a common soil over which they are planted on the farm. This might prove to be disastrous for a farmer if the infection gets spread throughout the farm. With the advancement of technology and the available resources, the detection of the early disease in crops, just by identifying its symptoms, will bring laurels to the agricultural industry. With the ever-rising demand for computer intelligence, it has now become possible to cure these crop-killing diseases. There is the availability of humongous data and images of crops but utilizing them for an analytical and problem-solving cause will really help the society. There are many different techniques through which the image of a crop is taken and thereby feed it to the trained system model for detection of crop health. Recent developments in artificial intelligence (AI) have highlighted and accelerated the use of different forms of AI technology in the agriculture sector. These technologies can be realized by handheld computers, autonomous agents (devices) running in unrestricted settings, autonomous and interactive situations, computer vision, sensing, and interaction with real-time environment. Integrating several collaborators and their disparate knowledge streams has resulted in the use of semantic technology [1, 2]. Interdisciplinary cooperation with specialists, such as the field of AI and agricultural science [1], is necessary to create reliable predictions for the planning and control of agricultural activities continuously. Many studies have examined by the computer algorithms for making the detailed classification of plant leaf diseases. These studies included various types of leaves, such as peach, orange, cherry,

cotton, and apple [3–5]. Sun et al. [3] proposed an architecture for leaf disease detection through the conversion of images from the red, green and blue (RGB) field to the hue saturation value (HSV) field for detecting leaf diseases. Poli et al. [6] presented the particle swarm optimization (PSO) algorithm to detect a cotton-diseased leaf. It achieved 95% test precision using feedforward neural network approach as presented. More than 4,483 images were used for a deep learning architectural system. These images are divided into 13 different grades. It achieved competitive accuracy of 91–93% for different fruit types like peach, apple, pear, grapevine leaves, and cherry [5].

Deep learning network architecture can be realized with many processing layers. It is also being utilized in areas like self-driving vehicles and in the study of vast numbers of dataset-based applications like image recognition and speech recognition. Traditional deep learning methods are different as compared to conventional machine learning methods, especially in terms of feature extraction. The conventional machine learning method uses the feature extraction method in manners of preceding stages of classifier. In deep learning method, the features are extracted and presented using multiple levels of hierarchy. So, the deep learning approach is better than the conventional machine learning approach in terms of feature extraction [25]. Many kinds of machine learning tools are in use, and the convolutional neural network (CNN) are most famous among all tools [26, 33]. Several researches were undertaken with respect to agriculture sector, in particular the area of plant identification and plant disease identification using the CNN tool. Another research [27] presented the identification of 26 plant diseases using open database of leaves with the images of different plants (total 14 plant image consider for calculation). The captured image and database ar e compared using two well-known architectures of GoogleNet [28, 29] and CNN AlexNet [27, 30–32]. For the maize, the result was generated up to 26.1%, 35% is observed for *Cercospora*, 73.9% for the common rust, and 100% for the Northern leaf blight. In a similar methodology, the detection of plant diseases by leaf image was developed, using a similar volume of Internet information, including 13 diseases and 5 plant precision of their models, which ranged from 91% to 98% based on test data. Recently, in some of the studies [34–36], performance of CNN-based pattern recognition methods for the identification of plants by using three different image databases of both fruit or plant leaves and whole plants was compared. Concluding results that CNNs outperform conventional methods were obtained. In [37], the investigator develops machine learning algorithm models to detect 9 different tomato diseases and pests (number of plants: 36,573). In this technique, very satisfactory efficiency achieved. In [27], the detection and diagnosis of plant diseases are finally carried out using deep learning. The models were trained 87,848 pictures of database. This database contains images of 25 different plants with 58 different sets of classes, using multiple model architectures. CNN offers a number of architectures for plant disease identification: AlexNetOWTBn [31, 38], Overfeat [39, 40], GoogLeNet [41], AlexNet [32, 42], Visual Geometry Group (VGG) [43–45], and CNN with tuning of various

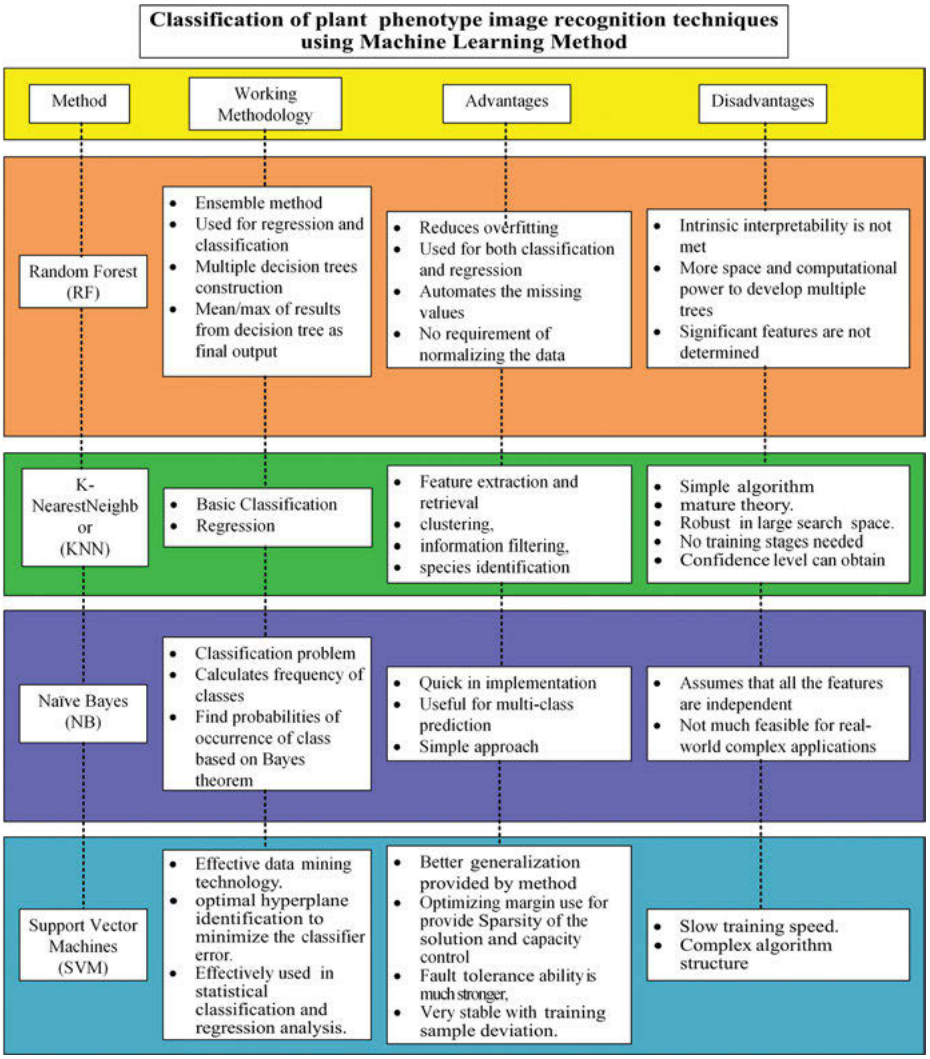


Figure 1: Classification of plant phenotype image recognition techniques. Comparative analysis of different methods of K-nearest neighbor (KNN) [7–9], support vector machines (SVM) [10–12], random forest (RF) [13, 14], and naïve Bayes (NB) [15, 16], in terms of methods, working methodology, advantages, and disadvantages.

hyper-parameters such as number of hidden layers, structure of network, and rate of learning. These parameters help design better model network training [46]. The detailed classification and the comparative analysis of different machine learning and deep learning methods are shown in Figures 1 and 2, respectively.

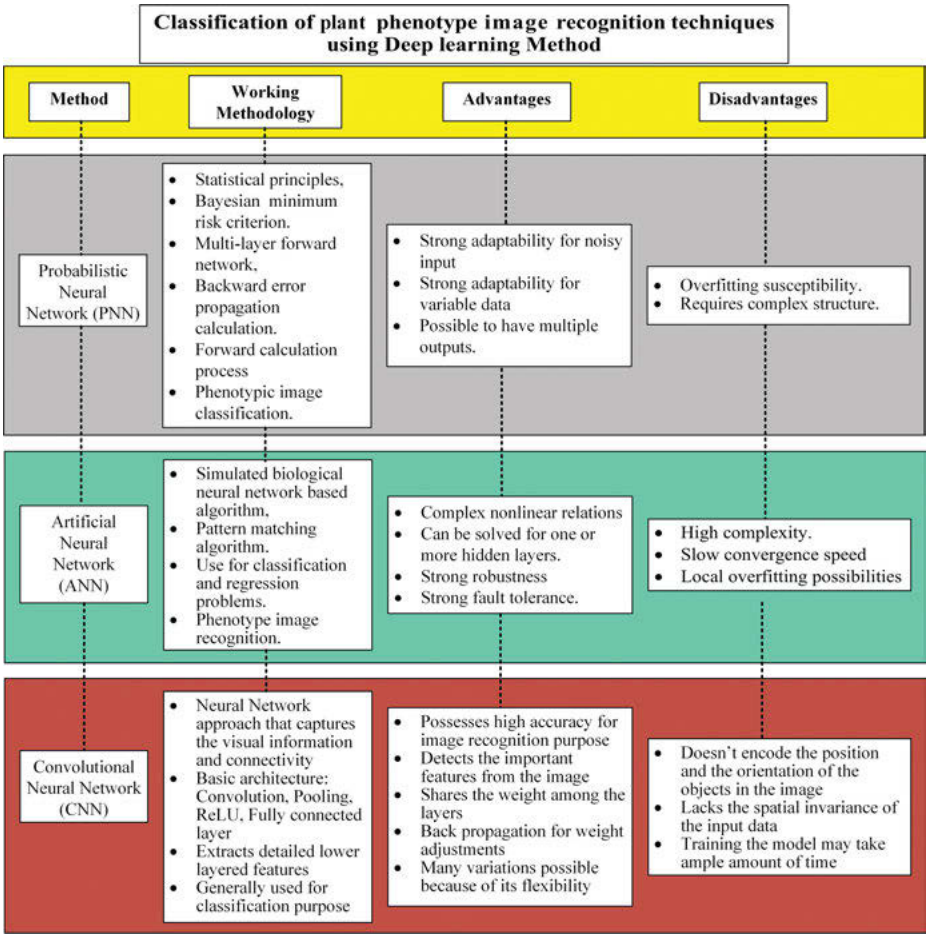


Figure 2: Classification of plant phenotype image recognition techniques using deep learning methods. Comparative analysis of different methods such as probabilistic neural network (PNN) [17–19], artificial neural network (ANN) [20, 21], and convolutional neural network [22–24], in terms of methods, working methodology, advantages, and disadvantages.

2 Generalized approach for crop disease detection

It is observed from different machine learning and deep learning methods that a large set of images are required to be given as input. The flowchart of the generalized approach for crop disease detection is given in Figure 3. These input image dimensions are required to be in uniform format depending on the classifier selection. The trained dataset comprises the set of images with the labels of the crop name and the healthy or disease name of the crop. The raw or scaled original image may not be a

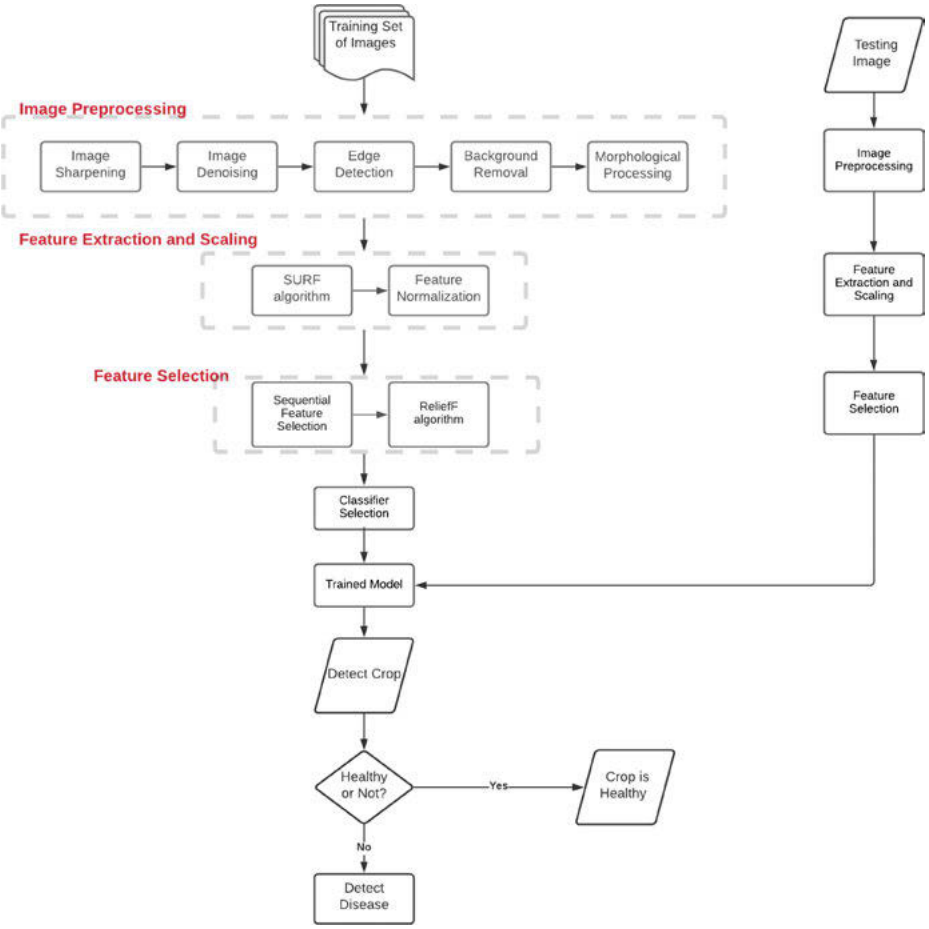


Figure 3: Generalized flowchart for crop disease detection.

good idea to be fed directly into the model. Hence, the image preprocessing step is required to be undertaken. The resultant processed image is fed to the feature extraction, selection, and feature selection module, which forms the set of features. The appropriate machine learning or deep learning–based classifier which helps in the training process of the model is used. Based on the trained model, the test image dataset is then preprocessed and the features are extracted and scaled, and eigenfeatures are selected. These, when served as an input to the trained model, determine the crop from texture and color features, and further classified into healthy and unhealthy classes. The crops that are classified as unhealthy are further processed for the disease classification. The description of each step is provided as follows:

2.1 Image preprocessing

Image preprocessing [47–49] and data cleaning are crucial steps in any machine learning–based applications. The raw data is far away scattered in conveying the critical information which is required to determine the features for any classification purpose. It helps to fetch the lowest level of abstraction. Preprocessing steps improve the image data that suppresses the undesired outliers or noises and enhances the image information for further analytical tasks. For crop disease detection, the image of the crop needs to be cleaned and filtered out to obtain the desired set of features. Following steps are required to be performed before heading for the feature selection:

2.1.1 Image sharpening

The algorithms, like human perceptions, are very sensitive to the fine details of the images, change in color, texture, and edges. Image sharpening is an image enhancement technique that creates the biasness between these fine details and another part of the image. It highlights the edges of the images by passing the original image through the high-pass filtering process. This act of image enhancement increases the contrast between the bright and dark regions in the image, which helps in bringing out features.

2.1.2 Image denoising

The images that have been captured might have degraded quality because of the noise, which leads to degradation of the feature selection procedure. So, an approach is required to denoise the image without losing the spatial information like corners, edges, textures, color, and other sharp structures.

2.1.3 Edge detection

The captured image might contain unnecessary sidewise information that does not contribute to the desired process. Hence, it is essential to remove that side information from the image. Edge detection is a mathematical operation that deals with determining a sharp change in the image brightness or if there is any discontinuity. This helps in extracting the core part of the information and provides the contour over the informatory image.

2.1.4 Background removal

Background removal is a crucial task, without which it may degrade the quality of feature extraction from leaf images. This step involves the removal of the background to avoid any potential bias in extracting features. This can be done by superimposing the generated contour from the edge detection step over the original image.

2.1.5 Morphological processing

Morphological operation is a nonlinear operation that refers to applying a structural element to an input image that processes the image based on the shapes. In the output image, the corresponding pixel and its neighbor pixel were compared to identify the values of each pixel.

2.2 Feature extraction and scaling

A major success of the classification system depends heavily on feature extraction and a selection step. It is a dimensionality reduction approach that deals with the selection of the important feature out of the given large number of features. Major features in the crop disease classification technique are color, shape, and textures.

The color feature of the image helps in determining the disease in the crop. The color of the diseased crop may tend to change and might have an intra-class color difference from that of the healthy crop. The color feature can be determined using a color histogram and color moments. The shape feature majorly contributes to determine the type of the crop since the shape may not help significantly in determining the disease in the crop except the diseases having symptoms of the cropped leaves. The texture of the leaf contributes heavily to the detection of the disease as the texture of the leaves changes from disease to disease of the crops. The texture-based feature can be considered as the major predictor element for detecting the disease.

2.2.1 Speed-up robust features (SURF)

Speed-up robust feature (SURF) algorithm is a robust approach that focuses on local, similarity invariant features and comparisons of the images. It is used in various practical applications like object recognition, image classification, and 3D image reconstruction. SURF starts by detecting the interesting point through blob detection. Blobs are the regions in the image which are similar to each other but beyond the region that differs vastly from the neighboring region in terms of color,

brightness, or contrast. These features are used to construct the bag of features for that class of the image.

2.2.2 Feature normalization

The extracted features are represented in the form of vectors. These features may have different units/ranges and might not be standardized within themselves. So it is essential to normalize the feature to compare them and utilize them in a proper selective manner. Feature scaling is usually applied to make the feature values constrained within a specific range of values. Max–min normalization is widely used in the feature normalization process. Standardization applies to scale the features with zero mean and unit variance.

2.3 Feature selection

Out of the features extracted, many features may not play a critical role in determining the classification purpose or they might be redundant in the presence of the other features. Feature selection [51] is required for the proper and quick model training, dimensionality reduction, simplifying the model, and reduce the chance of overfitting. Hence, selection of the proper useful feature is essential before training the model.

2.3.1 Sequential feature selection

As the name suggests, it can be performed in a definite sequence in two configurations, that is, one in a forward manner and the other in a reverse manner. The forward feature selection process selects one of the features and adds it successively in each iteration to form the set of features. While in a reverse manner, all features are selected and then removed iteratively which are redundant.

2.3.2 ReliefF algorithm

ReliefF algorithm is a feature ranking algorithm, which ranks the subsequent features and determines the significant features. The significant features will use to provide rank weights for individual features. The algorithm rewards the features that give change in the values to neighbors of different classes while it penalizes those features that give different values to the neighbors of the same classes.

2.4 Classifier selection

The preprocessed images from the features have been extracted from the bag of features, which have been fed as an input to the classifiers. These are classified into two categories: machine learning–based classifiers and deep learning–based classifiers.

2.4.1 Machine learning–based classifier techniques

2.4.1.1 K-nearest neighbor

K-nearest neighbor (KNN) [7] is a supervised machine learning technique that is used for regression and classification purposes. It plays an important role in determining the outliers. It works on the instinct that similar things appear to be nearby. KNN works on the principle of the idea of similarity represented as proximity, closeness, or distance. The classification is done based on the polling process. More the number of votes cast to a candidate class, the more the probability of the testing data to be falling in that casted class. It does not need to change its behavior on the addition or removal of any data observation. It is termed as a lazy learner algorithm that does not learn anything from the trained data but when the data is exposed to any testing sample, it performs its action. It is a nonparametric algorithm. The main considering parameter of this approach is the selection of the hyperparameter K. Smaller the value of K, more the possibility of overfitting will be, while higher the value of K, poor the classification accuracy will be due to the higher bias. Thus, the selection of K is very essential for KNN-based machine learning approach. Let (X_i, C_i), where i = 1, 2, . . ., n be the data points. X_i denotes feature values and C_i denotes the label class for X_i for each i. Assuming the number of classes as "c," $X_i \in 1, 2, 3, \ldots, c$ for all values of i. Let x be the point of query, whose neighbors we want to find based on the neighbor classes.

Pseudocode

- Calculate $d(x, x_i)$, $i = 1, 2, \ldots, n$, where d denotes the Euclidean distance between the points.
- Arrange the calculated n Euclidean distances in nondecreasing order.
- Consider the first k ($k > 0$) points from the sorted list of distances.
- Let k_i denote the number of points belonging to the ith class among k points. If $k_i > k_j \quad \forall i \neq j$, then put x in class i.

2.4.1.2 Support vector machine

Support vector machine (SVM) [10, 50, 52] determines a hyperplane of N-dimensions having N-features in the problem statement to classify the data points. SVM can be used for regression as well as classification-based approaches, but majorly it is used

for classification-based techniques. There are multiple decision boundaries or hyperplanes that can differentiate between the two classes, but the main intention behind the working of SVM is to divide the classes such that they have a maximum margin between the decision plane and the data points. Maximizing the marginal distance increases the confidence of the prediction of the class. It is also used in outlier detection. The selection of hyperplane is equal to the number of dimensions which depends on the number of features. In reality, the data may not be linearly separable. So, they are shifted to a higher dimensional space to separate the data to have a better decision boundary. This separation can be done using different kernel functions (Table 1) in SVM.

Pseudocode

Let X be the training data features and y be the class label of the data observation:

- Select some value of "C" (hyperparameter)
- repeat
- for all $\{X_i, y_i\}, \{X_j, y_j\}$ do
  optimize support vectors v_i and v_j
  end for
- until no change in sv

Table 1: Kernel functions in SVM.

S. no.	Kernel method	Kernel function
1	Linear	$K(X, Y) = \text{sum}(X^*Y)$
2	Polynomial	$K(X, Y) = 1 + \text{sum}(X^*Y)^d$
3	Gaussian	$K(X, Y) = \exp(-\|x-y\|^2/2\sigma^2)$

2.4.1.3 Random forest

Random forest is a supervised technique that uses collection of decision tree. A decision tree is a very primitive approach that overfits the generated model. Hence, random forest, an ensemble approach, is used as the bagging or boosting technique to overcome this issue of overfitting and is also used for the classification purpose. The problem of overfitting is dealt with by considering the number of decision trees considered as a subtree of the entire tree, on which the maximization of the resultant class is considered as a part of the decision class for the given observation. Moreover, this is a complex algorithm that results in a very slow prediction speed. This approach performs in the manner of the voting basis that each subtree outputs the resultant class, and the class having a majority is termed to be the class of that observation.

Pseudocode

- Randomly select "k" features from total "m" features, where $k << m$.
- Among the "k" features, calculate the node "d" using the best split point.
- Split the node into child nodes using the best split.
- Repeat steps 1–3 until "l" number of nodes has been reached.
- Build forest by repeating steps 1–4 for "n" number times to create "n" number of trees.

2.4.1.4 Naïve Bayes

Naïve Bayes [15] is a probabilistic classifier-based approach that works based on the assumption that all features are independent of each other. It uses the primary Bayes theorem for its classification purpose. The assumptions with which it works are wrong in the real sense but still the approach can be applied if all features are independent of each other. Since it directly works based on the Bayes theory, it is quite simple, easy to compute, and also the performance speed is quite high. It can also be used in multiple supervised learning approaches and in maximum likelihood approximation concepts.

Pseudocode

- Calculate the mean and standard deviation of each feature from each class.
- Repeat until the probability of all features is determined.
- Calculate the probability of ith feature using Gaussian density equation in each class.
- Calculate the likelihood of each class using conditional probability.
- Get the maximum likelihood.

2.4.2 Deep learning–based classifier techniques

2.4.2.1 Convolutional neural network

CNN is a very basic architecture that brings out the revolution in the field of AI over the image domain. CNN [22] model comprises convolutional filter or kernel with a max-pooling layer in each hidden layer, which also possesses the nonlinear activation function to bring out the nonlinearity in the hidden state. On the output side, the fully connected layer is used, which connects with the neurons of layers, followed by the Softmax function. The softmax function provides the probability of various classes and helps in determining the desired output class for the given input image data.

2.4.2.2 VGG16

VGG [43] developed a very fine deep CNN as a part of the model development for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) organized in 2014. This layer comprises the typical CNN-based model with a convolutional layer, padding, max-pooling layer, and a fully connected layer with the softmax as the output layer. VG-19 is a variant of VGG16 which has additional three convolutional layers that help in image identification.

2.4.2.3 AlexNet

AlexNet [53] is an eight-layered CNN architecture that possesses approximately 60 million+ parameters that were developed as a part of the competition of ILSVRC organized in 2012. The eight layers are divided in the form of five convolutional with max-pooling layer and the last three layers are the fully connected layer, which has leaky-ReLU activation function. The Softmax output layers help in the classification process of the classes of the image.

2.4.2.4 ResNet50

Deep neural networks are quite difficult to train. Hence, a residual-based framework [54, 55] is used for training the network that is subsequently deeper in the network. The architecture is capable of handling 150+ layers that helps in a large amount of training. Also, the issue of vanishing gradient dealt with nicely in the ResNet architecture. Similar to the typical CNN network, ResNet also comprises convolution and pooling layers. The kernel filter used is of the size of 3×3, the same as that of VGG16, and the input size of the image is 224×224 pixels.

2.4.2.5 Inception V3

Inception V3 [56, 57] is the extended version of Inception V1 and V2. The model of V1 was extended with the batch normalization feature in the inception model V2, which was enhanced with the factorization idea in V3. Version V3 helps reduce the parameters and connections, thus increasing the computational speed without having any impact on the accuracy or efficiency. The model comprises various building block architectures like convolution which creates feature maps using the input filter-based kernels, average pooling layers that average the feature map, max pooling layer that computes the maximum pixel that helps reduce the size of the input feature at the cost of increasing the dense layers for features, concated layers that combine the inputs of the same size, dropout layers that help in providing regularization, and finally the fully connected layer that connects the neurons of each layer. In between the layers, activation functions provide the activation to multiple neurons as per the weightage importance of the linkage between the two layers. The activation norm is done to introduce the batch normalization, and finally, the softmax function is used

to determine the classification by computing the classification loss. The architecture consists of 42 layers, and an input layer accepts the image of 299 × 299 pixels.

The reference links for the detailed step-wise understanding of the mentioned deep learning model architecture are mentioned in Table 2.

Table 2: Deep learning models and their architecture reference sources.

Deep learning model	Reference links
Convolutional neural network	[58]
VGG-16	[59]
AlexNet	[60]
resNet50	[61]
Inception V3	[62]

3 Evaluation parameters

Once the model gets trained, it is trained for the determination of the performance of the trained model using the testing dataset of the images. The trained model is evaluated based on the following performance matrices:

3.1 Confusion matrix

A confusion matrix, also called an error matrix, is used to identify the performance of the testing dataset. It determines the performance of the classification model as suggested in Table 3. It is the most used way of representing the performance parameters for a classification problem statement. It can be represented as follows:

True positive (TP): The number of cases where the predicted and actual cases are positive, that is, the plants/crops have diseases.

True negative: The number of cases where the predicted and actual cases are negative, that is, the plants/crops do not have diseases.

False positive: The predicted class was that the plant has the disease but they do not have the disease. They show the type I error.

False negative (FN): The predicted class was that the plant does not have the disease but they possess the disease. It is also shown as a type II error.

Type II errors are more severe in the case as they are more harmful for the field/crop's further production, as they tend to damage the entire field of crops and it needs to be taken care that the cases about FN are lowered.

Table 3: Classification problem statement.

Actual/predicted	Predicted: YES	Predicted: NO
Actual: YES	True positive (TP)	False negative (FN)
Actual: NO	False positive (FP)	True negative (TN)

3.2 Precision and recall

Precision represents the correctness of the diseases retrieved out of all the retrieved examples. The recall represents the proportion of the relevance of diseases out of all retrieved examples. Both of these parameters are required to determine the performance of the classifier:

$$\begin{aligned} \text{Precision} &= \text{True positive (TP)}/(\text{true positive (TP)} + \text{false positive (FP)}) \\ \text{Recall} &= \text{True positive (TP)}/(\text{true positive (TP)} + \text{false negative (FN)}) \end{aligned} \tag{1}$$

3.3 Accuracy

Accuracy is the major parameter to determine the classifier's performance. It determines the correct classification out of the total population:

$$\text{Accuracy} = (\text{True positive (TP)} + \text{true negative (TN)})/\text{total population} \tag{2}$$

3.4 *F*1 score

It is a harmonic mean between the precision and recall parameters for a classifier performance. It gives a better measure of performance for any classifier-based model. This metric of evaluation is more suitable for the uneven distribution of the classes compared to the accuracy:

$$F1 \text{ score} = (2 \times \text{precision} \times \text{recall})/(\text{precision} + \text{recall}) \tag{3}$$

3.5 Average precision

There can be multiple classes in the classification model. Hence, the precision needs to be calculated for all classes individually and then the average of all of these precisions is calculated:

$$\text{Average precision} = \left(\frac{1}{\text{No. of classes}}\right) \sum_{k=1}^{\text{No. of classes}} \text{precision}(k) \tag{4}$$

3.6 Average recall

There can be multiple classes in the classification model. Hence, the recall needs to be calculated for all classes individually and then the average of all of these recalls is calculated:

$$\text{Average recall} = \left(\frac{1}{\text{No. of classes}}\right) \sum_{k=1}^{\text{No. of classes}} \text{recall}(k) \tag{5}$$

3.7 Average *F*1 score

There can be multiple classes in the classification model. Hence, the *F*1 score needs to be calculated for all classes individually, and then the average of all of these *F*1 scores is calculated:

$$\text{Average } F1 \text{ score} = \left(\frac{1}{\text{No. of classes}}\right) \sum_{k=1}^{\text{No. of classes}} F1 \text{ score}(k) \tag{6}$$

3.8 Average accuracy

There can be multiple classes in the classification model. Hence, the accuracy needs to be calculated for all classes individually and then the average of all of these accuracies is calculated:

$$\text{Average accuracy} = \left(\frac{1}{\text{No. of classes}}\right) \sum_{k=1}^{\text{No. of classes}} \text{accuracy}(k) \tag{7}$$

3.9 Null error rate

There is certain classification training that may get biased toward the majority class. This is the term that defines how often the prediction can get wrong if the majority class is always predicted. It is generally used as a baseline metric of evaluation, where the probability of the trained class model is compared with. Higher the error rate than the null error rate, the better is the trained classifier model.

3.10 Cohen's kappa

It is the measure of determining the performance of the model compared with how well it could have performed, by chance. The higher the kappa score, the better is the trained model. The kappa score gets higher by the difference between the accuracy and null error rate.

3.11 ROC curve

It is a graph-based performance evaluation metric, which summarizes the performance of the trained model over different thresholds. It is the graph between true positive and false positive, and the threshold varied to observe the performance of a given class.

4 Future scope

The detection of crop disease is quite a complex, challenging, and nontrivial task. All the experiments performed in this field are generally under constrained conditions. The disease in the plants spreads gradually, without giving the farmers a hint of its existence. The previous approaches involved the usage of the sensory data collected from the different sensors like humidity, moisture, pH, and soil nutrient levels that monitor the nature of the soil constituents. But the disease can also be spread through pests that develop and grow over the leaves, which can be identified from the modern technological advancements in the field of computer vision and machine vision. Rather than capturing the images on the field, it is always a good choice to use the artificial humanoid or drone that flies over the crop field and captures the images of the crops that can process within itself and transfer the information to the driving center either directly or through ad hoc transmission mode depending on the field size. The image captured in the environmental conditions often plays an important role in determining the disease. The effect of intensity,

brightness, occultation, exposure, overlapped leaves, and others may also result in the false detection of crop health. This solution may also not always be a reliable solution that guarantees the stoppage of pest spread in the field in the initial phase. The nature of pest growth plays an important role that may grow in the bottom part of the leaves or any surface over the overlapped leaf that may not be visible through the image. The research fraternity is still striving for the outcome of the solution that can be reliable and beneficial in stopping the harmful impacts and effects of the pest spreading in the field. The possible remedies for the particular diseases can be stored in the machine with the help of agriculture experts. It will further save the time for the human-driven process to stop spreading up diseases. The advancement in the technological aspects will bring laurels in the field of agriculture and lead to the growth of the agriculture industry.

References

[1] A. Dengel, Special issue on artificial intelligence in agriculture, *KI – Künstliche Intelligenz*, 27 (4), 309–311, September 2013.
[2] D. Bonino and G. Procaccianti, Exploiting semantic technologies in smart environments and grids: Emerging roles and case studies, *Science of Computer Programming*, 95, 112–134, December 2014.
[3] G. Sun, X. Jia, and T. Geng, Plant diseases recognition based on image processing technology, *Journal of Electrical and Computer Engineering*, 1–7, 2018, 2018.
[4] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, Deep neural networks based recognition of plant diseases by leaf image classification, *Computational Intelligence and Neuroscience*, 1–11, 2016, 2016.
[5] P. Revathi and M. Hemalatha, Identification of cotton diseases based on cross information gain deep forward neural network classifier with PSO feature selection, *International Journal of Engineering and Technology*, 5(6), 4637–4642, 2014.
[6] R. Poli, J. Kennedy, and T. Blackwell, Particle swarm optimization, *Swarm Intelligence*, 1(1), 33–57, August 2007.
[7] Y. Sun, Y. Liu, G. Wang, and H. Zhang, Deep learning for plant identification in natural environment, *Computational Intelligence and Neuroscience*, 1–6, 2017, 2017.
[8] H. Yalcin. Analysis of agricultural features. In *2019 27th Signal Processing and Communications Applications Conference (SIU)*. IEEE, April 2019.
[9] Y. Kim, B.-T. Zhang, and Y. T. Kim, *Machine Translation*, 16(2), 89–108, 2001.
[10] J. Xiong, D. Yu, S. Liu, L. Shu, X. Wang, and Z. Liu, A review of plant phenotypic image recognition technology based on deep learning, *Electronics*, 10(1), 81, January 2021.
[11] T. Rumpf, A.-K. Mahlein, U. Steiner, E.-C. Oerke, H.-W. Dehne, and L. Plümer, Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance, *Computers and Electronics in Agriculture*, 74(1), 91–99, October 2010.
[12] J. Liu, S. Zhang, and S. Deng, A method of plant classification based on wavelet transforms and support vector machines, In: *Emerging Intelligent Computing Technology and Applications*, Springer, Berlin Heidelberg, 253–260, 2009.

[13] J. Ma, K. Du, F. Zheng, L. Zhang, Z. Gong, and Z. Sun, A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network, *Computers and Electronics in Agriculture*, 154, 18–24, November 2018.
[14] K. Pankaja and V. Suma, Plant leaf recognition and classification based on the whale optimization algorithm (WOA) and random forest (RF), *Journal of the Institution of Engineers (India): Series B*, 101(5), 597–607, July 2020.
[15] A. Akhtar, A. Khanum, S. A. Khan, and A. Shaukat. Automated plant disease analysis (APDA): Performance comparison of machine learning techniques. In *2013 11th International Conference on Frontiers of Information Technology*. IEEE, December 2013.
[16] F. R. F. Padao and E. A. Maravillas. Using naïve Bayesian method for plant leaf classification based on shape and texture features. In *2015 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*. IEEE, December 2015.
[17] C. Mallah, J. Cope, and J. Orwell, Plant leaf classification using probabilistic integration of shape, texture and margin features, In: *Computer Graphics and Imaging / 798: Signal Processing, Pattern Recognition and Applications*, ACTAPRESS, 2013.
[18] S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang. A leaf recognition algorithm for plant classification using probabilistic neural network. In *2007 IEEE International Symposium on Signal Processing and Information Technology*. IEEE, December 2007.
[19] K. Majid, Y. Herdiyeni, and A. Rauf. I-PEDIA: Mobile application for paddy disease identification using fuzzy entropy and probabilistic neural network. In *2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*. IEEE, September 2013.
[20] S. Iniyan, R. Jebakumar, P. Mangalraj, M. Mohit, and A. Nanda, Plant disease identification and detection using support vector machines and artificial neural networks, In: *Advances in Intelligent Systems and Computing*, Springer, Singapore, 15–27, 2020.
[21] M. W. Shi, Based on time series and RBF network plant disease forecasting, *Procedia Engineering*, 15, 2384–2387, 2011.
[22] S. P. Mohanty, D. P. Hughes, and M. Salathé, Using deep learning for image-based plant disease detection, *Frontiers in Plant Science*, 7, September 2016.
[23] M. H. Saleem, J. Potgieter, and K. M. Arif, Plant disease classification: A comparative evaluation of convolutional neural networks and deep learning optimizers, *Plants*, 9(10), 1319, October 2020.
[24] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, A comparative study of fine-tuning deep learning models for plant disease identification, *Computers and Electronics in Agriculture*, 161, 272–279, June 2019.
[25] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, C. Brian, V. Esesn, A. A. S. Awwal, and V. K. Asari. The history began from AlexNet: A comprehensive survey on deep learning approaches, 2018.
[26] J. G. A. Barbedo, Factors influencing the use of deep learning for plant disease recognition, *Biosystems Engineering*, 172, 84–91, August 2018.
[27] K. P. Ferentinos, Deep learning models for plant disease detection and diagnosis, *Computers and Electronics in Agriculture*, 145, 311–318, February 2018.
[28] V. Singh and A. K. Misra, Detection of plant leaf diseases using image segmentation and soft computing techniques, *Information Processing in Agriculture*, 4(1), 41–49, March 2017.
[29] M. Brahimi, M. Arsenovic, S. Laraba, S. Sladojevic, K. Boukhalfa, and A. Moussaoui, Deep learning for plant diseases: Detection and saliency map visualisation, In: *Human and Machine Learning*, Springer International Publishing, 93–117, 2018.

[30] T. Shanthi and R. S. Sabeenian, Modified AlexNet architecture for classification of diabetic retinopathy images, *Computers & Electrical Engineering*, 76, 56–64, June 2019.
[31] P. Saleem and M. Arif, Plant disease detection and classification by deep learning, *Plants*, 8(11), 468, October 2019.
[32] S. B. Jadhav, V. R. Udupi, and S. B. Patil, Identification of plant diseases using convolutional neural networks, *International Journal of Information Technology*, February 2020.
[33] M. Hussain, J. J. Bird, and D. R. Faria, A study on CNN transfer learning for image classification, In: *Advances in Intelligent Systems and Computing*, Springer International Publishing, August, 191–202, 2018.
[34] A. Abade., A. S. de Almeida, and F. Vidal. Plant diseases recognition from digital images using multichannel convolutional neural networks. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications – Volume 5: VISAPP*, 450–458. INSTICC, SciTePress, 2019.
[35] S. Zhang, W. Huang, and C. Zhang, Three-channel convolutional neural networks for vegetable leaf disease recognition, *Cognitive Systems Research*, 53, 31–41, January 2019.
[36] A. Darwish, D. Ezzat, and A. E. Hassanien, An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis, *Swarm and Evolutionary Computation*, 52, 100616, February 2020.
[37] A. Fuentes, D. H. Im, S. Yoon, and D. S. Park, Spectral analysis of CNN for tomato disease identification, In: L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada, editors, *Artificial Intelligence and Soft Computing*, Springer International Publishing, Cham, 40–51, 2017.
[38] A. Krizhevsky. One weird trick for parallelizing convolutional neural networks, 2014.
[39] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks, 2014.
[40] L. M. Abou El-Maged, A. Darwish, and A. E. Hassanien, Artificial intelligence-based plant diseases classification, In: *Studies in Big Data*, Springer International Publishing, 45–61, 2020.
[41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–9, 2015.
[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems*, 25, 1097–1105, 2012.
[43] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.
[44] A. K. Rangarajan and R. Purushothaman, Disease classification in eggplant using pre-trained VGG16 and MSVM, *Scientific Reports*, 10(1), February 2020.
[45] P. Bedi and P. Gole, Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network, *Artificial Intelligence in Agriculture*, 5, 90–101, 2021.
[46] Y. Yoo, Hyperparameter optimization of deep neural network using univariate dynamic encoding algorithm for searches, *Knowledge-Based Systems*, 178, 74–83, 2019.
[47] D. A. Pitaloka, A. Wulandari, T. Basaruddin, and D. Y. Liliana, Enhancing CNN with preprocessing stage in automatic emotion recognition, *Procedia Computer Science*, 116, 523–529, 2017.
[48] S. Tabik, D. Peralta, A. Herrera-Poyatos, and F. Herrera, A snapshot of image pre-processing for convolutional neural networks: Case study of MNIST, *International Journal of Computational Intelligence Systems*, 10(1), 555, 2017.

[49] J. Yim and K.-A. Sohn. Enhancing the performance of convolutional neural networks on quality degraded datasets. In *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, November 2017.
[50] Y. Lin, L. Fengjun, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang. Large-scale image classification: Fast feature extraction and SVM training. *CVPR 2011*. IEEE, June 2011.
[51] J. Miao and L. Niu, A survey on feature selection, *Procedia Computer Science*, 91, 919–926, 2016.
[52] D. Das, M. Singh, S. S. Mohanty, and S. Chakravarty. Leaf disease detection using support vector machine. In *2020 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, July 2020.
[53] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet classification with deep convolutional neural networks, *Communications of the ACM*, 60(6), 84–90, May 2017.
[54] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
[55] I. Z. Mukti and D. Biswas. Transfer learning based plant diseases detection using ResNet50. In *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*. IEEE, December 2019.
[56] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
[57] Z. Qiang, L. He, and F. Dai, Identification of plant leaf diseases based on inception v3 transfer learning and fine-tuning, In: *Communications in Computer and Information Science*, Springer, Singapore, 118–127, 2019.
[58] K. Smeda. Understand the architecture of CNN, November 2019.
[59] R. Thakur. Step by step VGG16 implementation in Keras for beginners, November 2020.
[60] A. Daniel. Understanding AlexNet: A Detailed Walkthrough, September 2020.
[61] P. Dwivedi. Understanding and Coding a ResNet in Keras, March 2019.
[62] B. Raj. A Simple Guide to the Versions of the Inception Network, July 2020.

Satish K. Jain, Shobha Jain

Artificial intelligence applied to multi- and broadband antenna design

Abstract: Looking into the multiband and broadband need of 5G wireless technology, researchers are trying hard to explore antennas design having multi- and broadband characteristics. Mostly, planar and vertical designs are suggested for these types of designs because various advantages like small size, low manufacturing cost, low profile, volume production, and conformability. It seems that stacked configurations, namely, antennas in vertical direction, may be one of the solutions for getting multiband and broadband features because the physique of the overall antenna does not expand horizontally. This is one of the most important features, which provides opportunity to construct a large array with limited space. However, the design process of stacked microstrip antenna using commercially available electromagnetic software is a cumbersome task because of "Generate and Test" approach involved. Furthermore, because stacked patch antenna involves large number of design parameters, therefore, the designer needs to optimize so many geometrical parameters. Hence, hundreds of simulations may require to reach at the final design. In this chapter, artificial intelligence algorithms such as artificial neural network, deep learning, and particle swarm optimization metaheuristic approach have been explored to systemize the entire design process of stacked patch antennas design. The proposed hybrid algorithms are stable and flexible computationally, which is able to provide accurate results. Developed antennas are useful for satellite, wireless local area network, and radar communication applications. The performance of the designed antennas has been verified through electromagnetic simulations done by IE3D software and experimental measurements accomplished through vector network analyzer. The close resemblance of the simulation and experimental results with the design specifications confirms the validity of the developed design methodology.

Keywords: artificial intelligence, particle swarm optimization, computer simulation technology, metaheuristic

Acknowledgments: This work was supported by the Indian Space Research Organization (ISRO, India) and Shri G.S. Institute of Technology and Science, Indore (MP), India, through the Ministry of Human Resource Development (MHRD) India, under the Project TEQIP-II (NPIU).

Satish K. Jain, Department of Electronics and Telecommunication Engineering, S.G.S. Institute of Technology and Science, Indore, India, e-mail: satishjain.jain@gmail.com
Shobha Jain, Department of Mathematics, S.V.V.V, Indore, India, e-mail: shobajain1@yahoo.com

https://doi.org/10.1515/9783110734652-004

1 Introduction

A considerable research and development effort is being consistently devoted to the design and development of planar microstrip patch antennas [1, 2]. At present, rapid expansion in 5G mobile network and miniaturization of mobile handset indicates that the demand for microstrip antennas and arrays will increase. In wireless communication area, there is a need of integrating different communication services in a single device, thereby requiring multiband and broadband operations. In microstrip antenna technology, stacked configuration is one of the common and viable solutions for obtaining wideband and multiband characteristics because the size of the structure does not increase in the planar direction, making them available for use in antenna arrays. As on today, a number of commercial and freeware simulators are available for the analysis of these antennas [3]. Despite the current level of design sophistication, from designer's view point, even today designers require to execute so many simulations numerically in order to extract the exact values of the antenna geometrical parameters. Hence, optimization becomes tedious numerically. Also, vender-provided electromagnetic software packages for antenna design occupy large memory even in gigabyte so computer resources get engaged heavily. In a nutshell, there is still a need for a tool for the design of radiating electromagnetic structures having a large number of design variables that can produce tailor-made structures in quickest possible time without compromising accuracy. This research is an effort to bridge the gap in this direction. Basically, the design task is approached as an optimization problem and instead of classical optimizers, artificial intelligence–based algorithms are used for the solution. Stacked patch antennas and their variants are taken as candidate structures to test the validity of the artificial intelligence–based developed methodology.

The entire design approach is based on the two artificial intelligence algorithms (AIAs): artificial neural network (ANN) and particle swarm optimization (PSO). For the design of proposed antenna structure, available PSO algorithms have been updated toward stacked patch antenna geometrical parameter optimization. Since artificial intelligence–based PSO algorithm requires fitness function, ANN-based black-box models have been developed and applied to develop the fitness function that is able to connect the geometrical parameters of the stacked patch antenna with its working frequencies and bandwidths. These trained neural networks are merged in an optimization loop of PSO. Finally, developed hybrid algorithms are able to provide geometrical parameter values of the antenna for user-specific frequencies within X–Ku band (8–12 and 12–18 GHz) and bandwidth till 40%. The performance of the proposed AIA-based methodology, that is, ANN-merged PSO, has been tested with respect to accuracy. One of the unique features of the developed methodology is that the design process is very much user-friendly as the designer has to give only the design frequencies and required bandwidths as inputs and the optimized numerical values of geometrical parameters are obtained as output.

1.1 An overview of stacked patch antennas

Stacked patch antenna is a suitable candidate for obtaining multiband/broadband operation [4] (Figure 1). In this type of antennas, two or more patches on different layers of the dielectric substrate are stacked on each other. The bottom patch can be excited either with coaxial feed, whereas the top parasitic patch is excited through electromagnetic coupling with the bottom patch. The patches can be fabricated on different substrates, and an air-gap or foam material can be introduced between these layers to increase the bandwidth. In place of coaxial feed, the bottom patch can be excited either by a microstrip line or through electromagnetic aperture coupling. The method of excitation influences the bottom patch characteristics and usually does not significantly affect the performance of the stacked patches. Although this method of stacking increases the overall height of the antenna, the size in the planar direction remains the same as that of the single patch antenna. Thus, these antennas may be suitable for use as array elements in applications like 5G mobile communications due to limited space occupying.

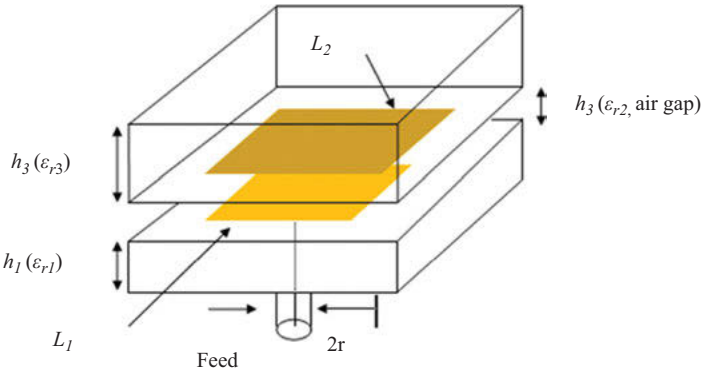


Figure 1: Stacked patch antenna layout. Adopted in a modified form with permission from ref. [87]. Copyright 2021, Springer Nature.

Antennas having number of patches in stacked form have been explored by researchers [5–8]. In these types of antennas, when patches are close to each other (critical coupling) and resonating at two adjunct frequencies, they act as broadband antennas; whereas they act as multiband antennas when patches are kept at sufficient distance to maintain under coupling between two different resonating frequencies [9–11]. Since the resonating frequency of antenna depends on its geometrical parameters, structure dimensions play a crucial role in the design process. Authors have proposed some antenna design procedures [12, 13]; still they feel the requirement of user-friendly approach computationally. Since the conventional stacked patch antenna is able to provide the limited bandwidth, it is found that the stacked patch antenna having

U- and *E*-shaped geometries, which are achieved by creating slots in normal patch, are able to provide broadband performance [18, 20]. Keeping this in mind, a lower copper patch in antenna is converted into *E*-shaped patch by creating two rectangular slots (Figure 2). In this way, the finally achieved structure has square-shaped patch at the top side and *E*-shaped patch at the bottom side. In the final structure of antenna, dielectric layer parameters (h_1, ε_{r1}, ε_{r2}, h_3, ε_{r3}) and both patch geometrical parameters (L_1, L_2, h_2, length of slot l_s, width of slot w_s, position of slot p_s, and feed location x_f) need to be optimized to get the desired broadband performance.

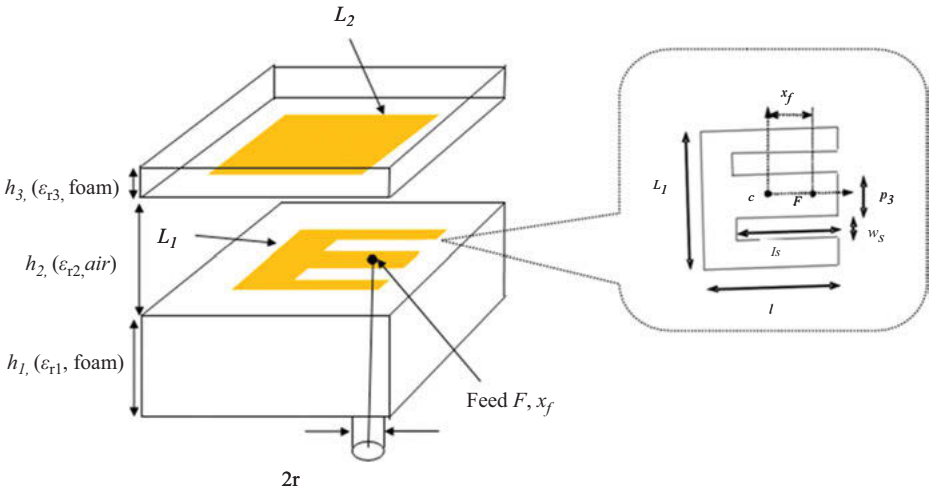


Figure 2: Broadband stacked patch antenna layout.

1.2 Literature review

1.2.1 Multi- and broadband stacked patch antennas

Researchers had started using stacked configuration in antenna design nearly at the end of 1970s, and they have explored this concept in various ways like various excitation techniques [11, 14, 15], various patch geometries [11, 16–18], impact of various dielectric layer joints [12], broadbanding [10–12, 17–21], miniaturizations [22, 24, 25], and multifrequency operation with different polarizations [9, 14, 23, 26–29]. Authors developed various techniques to analyze these [30–32]. The requirement of larger bandwidth in antenna applications has been realized and it is reflected in their research work, including papers and books [4, 6, 33–36]. Although various computational and measurement outcomes have been delivered in their research

activity, however, works leading to the analysis and design optimization activity of antennas are very much rare still [13, 33].

Depending on the design dimensions, stacked patch antenna can exhibit dual-band, multiband, or broadband behavior. Various analytical and numerical techniques have been developed for the analysis of stacked patch antennas [30–32, 39, 40, 55]. Analysis of stacked patch antenna through a cavity model is proposed in [58]. A method for calculation of input impedance of probe-fed stacked circular microstrip antenna has been proposed in [38], and resonance frequency of multilayer stacked patch antenna has been calculated in [37, 38].

1.2.2 Artificial intelligent algorithms in antenna domain

Due to some inherent advantages of AIAs as compared to their classical counterparts, the radio frequency (RF) and microwave community has recently been using them for the solution of many complex problems. These algorithms include ANN, genetic algorithm (GA), PSO, bacteria foraging optimization (BFO), simulated annealing (SA), ant colony optimization (ACO), and differential evolution (DE). GA, PSO, BFO, SA, ACO, and DE are metaheuristic approaches, whereas ANN is a part of deep learning. In this section, we have briefly reviewed the application of two AIAs like ANN (deep learning) and PSO (metaheuristic approach), in order to solve the design problems of antennas.

A thorough investigation of ANN technique in RF and microwave application has been done by Patnaik et al. [41]. Authors have described the applications of different ANN models like backpropagation, hopfield, and radial basis functions in microwave engineering. They clarified the use of ANN technique in areas where device physics is not fully understood, but device output for the specified inputs is known. A robust algorithm for automatic development of neural network models for microwave applications, namely, aperture feed antenna analysis, has been proposed in [42–44]. In [45], a generalized ANN model has been proposed to determine the resonance frequency of various microstrip antennas. Mishra et al. [46–48] proposed an ANN-based CAD model to solve the analysis and design problem of square and rectangular patch antennas with the error backpropagation algorithm. Application of ANN in finding the design parameters of the broadband multislot hole-coupled microstrip antenna is described in [46, 49]. In this work, tunnel-based ANN was developed to calculate the radiation patterns of the antenna. Multilayered perceptron-based ANN has been used for computing the resonant frequency of rectangular microstrip antenna with thin and thick substrates [50]. In this work, many deep learning algorithms like Levenberg–Marquardt, Bayesian regularization, scaled conjugate gradient, and conjugate gradient of Powell–Beale were applied and performance was compared. Application of ANN in characterizing multiband reconfigurable antennas was done in [51, 52]. In this work, multilayer perceptron (MLP) was used to locate the operational frequency bands of antenna at different

reconfigured conditions. Neurocomputational capability is utilized in finding the resonance frequency of equilateral triangular microstrip antenna in [53]. Design optimization of dual-band multilayer stacked patch antennas using ANN deep learning technique has been done in [54]. In this work, it has been demonstrated that the developed ANN model is capable of replacing repeated simulation work. In the same line, various authors have proposed the application of various heuristic approaches in order to address antenna optimization issues [59]. A heuristic approach-based PSO algorithm was introduced by Eberhart et al. [60–63] and applied first time in 1998 for optimization problems. Rahmat Samii et al. [57, 64] have proposed PSO algorithm in electromagnetic applications leading to antenna optimization in 2004. In order to improve the PSO performance, various boundary conditions were suggested by Xu and Samii [65]. Different versions of PSO like real number PSO, binary PSO, and single objective and multiobjective PSO were developed for antenna design [66, 67]. In order to improve the PSO performance further, time-varying acceleration coefficients and inertia weights were suggested in [68, 69]. In order to stabilize the velocity of particles during heuristic search, an analysis was done on various parameters of PSO [70, 71]. On the basis of reported results in the above literature, PSO has been applied for design optimization of rectangular patch antenna as well as equilateral triangular patch antennas [72, 73], where dielectric constant, thickness of the substrate, and operational frequency were given as inputs, and length and width of the patch were the outputs. Multiband and wideband patch antennas were designed using PSO in [74], where PSO and finite difference time domain (FDTD) technique were combined to achieve the optimum antenna satisfying a certain design criterion.

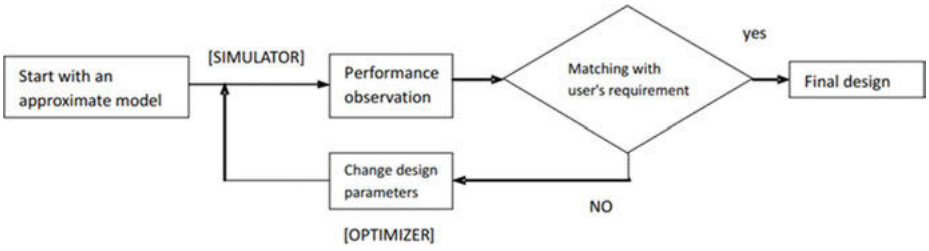


Figure 3: Traditional antenna design procedure.

The antenna geometric parameters to be optimized were extracted by PSO, and the fitness function was evaluated by the FDTD simulator software i.e. computer simulation technology (CST). PSO was tested on the rectangular patch antenna and *E*-shaped patch antenna design. Dual-band antenna has also been designed using basic PSO, Boolean PSO, and modified PSO [75]. Spline-shaped UWB antenna and multiband coplanar wave guide (CPW)-fed monopole antenna and antenna array for wireless local area network (WLAN) application have been designed using PSO [76–78]. The above brief review reveals that deep learning and heuristic approach–based artificial

intelligence techniques are able to provide better solution to fix all the design parameters of a stacked patch structure for specific frequency and bandwidth response.

1.3 Proposed hybrid approach

In this chapter, artificial intelligence–based optimization approach has been explored to design microstrip multipatch antenna. In order to make a computer-aided design module as a least time-consuming one, a simulator is proposed to be replaced by an optimizer comprising trained ANN model merged with PSO. The simulator-based procedure involving "generate and test" for any microwave component design is shown in Figure 3. The proposed strategy of optimization is shown in Figure 4. In general practice, the antenna designer uses simulator where geometrical parameters of antenna are fed and the response is observed. The parameters are kept on changing until and unless the designer gets the desired response. If it is seen computationally, it is nothing but an optimization process in which the designer keeps on reducing error between desired and computational responses by varying antenna geometrical parameters. The root of the proposed design methodology lies with this fact, where we have used heuristic-based PSO as the optimizer. But the most time-consuming part in the usual process of antenna designing, that restricts the use of interactive optimization, is the recording of the simulator response for each and every updated design. This difficulty has been circumvented in the proposed work by applying the deep learning–based trained ANNs before the optimization process. The developed ANN model has been merged with the optimizer algorithm loop for estimation of the design response instantly. The process of designing can be understood by the schematic shown in Figure 4.

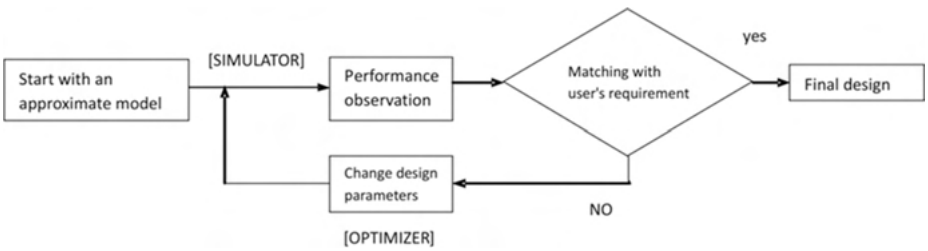


Figure 4: Artificial intelligence–based design methodology.

1.4 Artificial intelligence algorithms applied to proposed antennas

Researchers in RF engineering have been using deep learning and heuristic approach-based AIAs for more than a decade, and various complex and novel problems have been attempted and solved. The reasons for motivation of applying these algorithms are: the speed of commercial simulator is overcome, the demand of high computer resources is overcome, and the process is user-friendly.

1.4.1 Neuron-based deep leaning algorithm for stacked patch antenna analysis

Neural networks are having few special abilities like the ability to learn from data, ability to generalize patterns in data, and ability to model nonlinear relationships. Moreover, these algorithms have been applied since last few years in different aspects of antenna design [41–44]. In the proposed research activity, the artificial neuron-based deep learning algorithm helps in getting rid from the repetitive use of simulators during its antenna parametric optimization through evaluating of the fitness function required for the optimizer. An ANN can work electively in the data range in which it is trained. Furthermore, our main objective in this work is to propose a computational fast model, which should be designer-friendly for any complex antenna having any number of design variables. Based on these facts, four different ANN models were developed with varying inputs. This chapter justifies the development of ANN models for the analysis of antenna, their development procedure, and performance testing. Different ANN architectures can be proposed depending on the analysis problems to be handled. In order to analyze our specified problem, we have used multilayered feed-forward ANNs, which are trained in the supervised mode, and error backpropagation algorithm has been applied [79].

1.4.1.1 Neuron-based deep learning algorithm for stacked patch antenna analysis

Every neural network has at least one input and one output layer. Other layers in between are hidden layers. The number of layers and neuron depends on the model configuration. As per the literature, it is proved that at least three-layer ANN having an input layer, a hidden layer with a output layer consisting enough number of neurons can solve any nonlinear problem. However, for a complex mapping problem, more than one hidden layer has been proposed. The elements corresponding to the input layer (n_i) neurons keep all the input variables to be mapped with output in an explicit form. The number of neurons (n_O) at the output layer must match with the number of parameters to be outputted. Decision about the size and number of hidden layer is very tricky in ANN model formation. Unfortunately, no fixed rule exists for the selection of number of neurons (n_h) in a hidden layer. It is totally problem specific. In our

proposed work, decision about the number of neurons in hidden layer was based on the few criteria. Initially, the process started with only little number of neurons in the hidden layer, and the training error is calculated as follows (eq. (1)). On the basis of this error, number of neurons in the hidden layer increases:

$$E = \frac{1}{2}\sum_{k=1}^{m}\sum_{i=1}^{n_0}(y_{i,k} - \hat{y}_{i,k})^2 \tag{1}$$

where $y_{i,k}$ are the expected output and $\hat{y}_{i,k}$ are the associated estimates, which are delivered by the network with respect to the set (y_k, x_k), where $k = 1,2,3, \ldots, m$, m being the sample numbers in the training set, $x^*_k = [x_{1,k}, x_{2,k}, \ldots, x_{n,k}]$ are the network's inputs, and $y^*_k = [y_{1,k}, y_{2,k}, \ldots, y_{n,k}]$ are the associated expected outputs. If this network delivered error is well within the quoted training margin and further possibility of improvement is not there, then training is to be stopped; otherwise, the same procedure is to be followed with hidden layer neurons increased. When network delivered the minimum value of training error E, then this is assumed to be the expected model.

1.4.1.2 Neural network parameters, data generation, training, and testing

In the proposed research, iterative optimization process followed by the simulator is replaced by the trained ANN model. Trained ANN model is having the role of mapping antenna geometrical variables with its output parameters in terms of resonance frequency and/or bandwidth. The network training mechanism is shown in Figure 5.

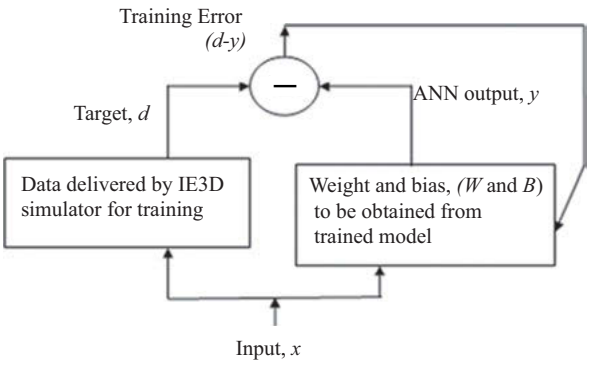


Figure 5: ANN training approach.

Two types of data are needed. The first dataset is called "training dataset," and other dataset is called testing dataset. Training data is needed to model the network and it consists of chosen input data and their corresponding output data. Whereas

the test dataset is applied to check the effectiveness of the developed trained model. The training and test dataset are different in values. The size of the training dataset depends on the size of the ANN and complexity of the relationship between input and output. While training network, input (x) is applied to ANN and response (y) is noted down. It is compared for same input, response from simulator (output d) is also to be recorded, and difference as error (d–y) is observed (Figure 5). This procedure is repeated until and unless the training error is minimized (difference, d–y). When this situation is met, ANN weight and bias values (W, B) are recorded [80–84].

In the literature, two types of training algorithms have been proposed: "online training" algorithm and "off-line training" algorithm. In the research work presented in this chapter, online training approach has been recommended as it is proved to be more efficient in most of the cases. The concept of backpropagation algorithm has been utilized to reduce the error [84].

The training efficiency of the network depends on few parameters, which are listed as follows:

- **Number of hidden layers**: It has been proved by researchers that any ANN with many layers, namely, minimum one hidden layer, is capable to model any complex nonlinear relationship. We have used three layers for first problems, whereas four layers for last two problems of design.
- **Number of hidden layer neurons**: The number of neurons in hidden layer plays an important role in determining the structure of the network. In order to model complex relationships, one needs to optimize the number of neurons in the hidden layer. Otherwise, too many neurons may lead to overstrain of the network.
- **Learning rate**: In order to increase the converge speed, one has to optimize the rate of learning while training.
- **Momentum**: The speed of convergence can also be enhanced by decreasing the number of adaptation cycles.
- **Training tolerance**: This is one of the most sensitive learning parameters. It helps in determining the accuracy of ANN outputs.

1.4.1.3 Development of artificial neural network models

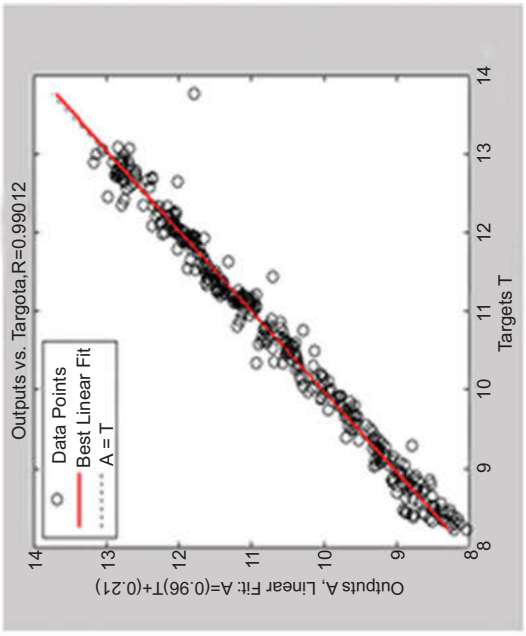
In this section, the procedure of ANN model development using MLP backpropagation algorithm has been described. A total of four ANN models have been developed. The very first ANN developed model (model-1) helps in predicting the frequencies of the stacked patch antenna working well within X–Ku band (8–18 GHz). Model-1 acts as a tool that can construct a relationship between the three input variable parameters of the stacked patch antenna geometry (size of the two microstrip patch L_1, L_2 and the air-gap h_2) and two resonating frequencies (f_{r1} and f_{r2}) as outputs. These are the same input parameters, those we have used to develop training data through simulator. The training error plot is shown in Figure 6. Training error plot in Figure 6(a)

converges within 120 numbers of epochs and Figure 6(b) refers to the closeness of training output and actual output for the first ANN model. The purpose of developing this second model (model-2) is to design stacked patch antenna at specific frequencies along with the associated bandwidth. In order to develop this model, number and type of input parameters will remain the same as in model-1, but the associated bandwidth (%) at each frequency is the additional output parameter. This ANN model is also able to work within the same frequency (X–Ku) band range. Model-3 was developed in 2–6 GHz frequency range. This model is helpful in designing stacked patch antennas for WLAN band applications [93–95]. The details of various WLAN license-free bands are available in literature and listed as follows: (1) 2.4 GHz: 2,403–2,483 MHz; (2) 5 GHz: 5,150–5,250, 5,250–5,350, 5,725–5,825, and 5,825–5,850 MHz. For this particular problem, we have taken one extra geometrical parameter, that is, feed position ($x_p = y_p$) along with L_1, L_2, and h_2,. In this problem also, totally four output parameters have been considered, that is, resonant frequencies (2 in number) and the associated bandwidth (2 in number). To get an effective performance, two hidden layers were used.

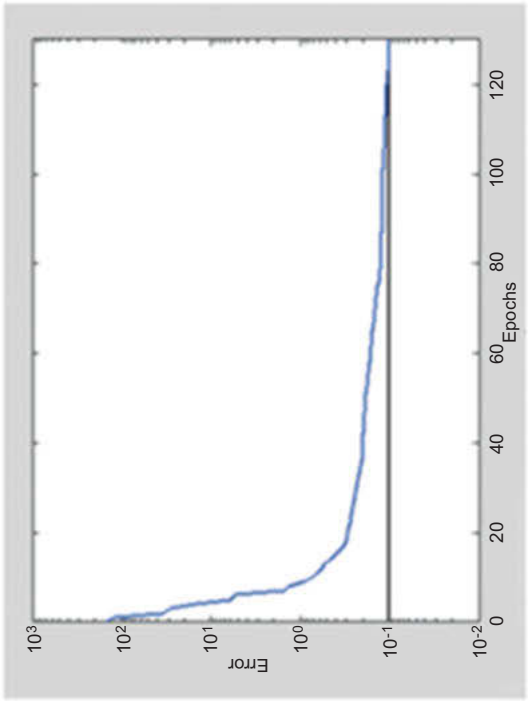
In order to design broadband (up to 40% bandwidth) resonance stacked patch antennas with a single resonance frequency (within 1–10 GHz), model-4 was helpful. These range of frequency and bandwidth are being needed in radar applications. So many authors have applied *E*-shaped patch antennas to get an antenna broadband performance [56]. In our work also, the same concept has been utilized. In the proposed structure, bottom patch is in E-shape (Figure 2) and upper patch is in square shape. In the parameters list, there are constant parameters and variable parameters. Constant parameters are as follows; $h_1 = 10.8$ mm, $\varepsilon_{r1} = 1.2$, $\varepsilon_{r2} = 1$, $h_3 = 5.4$ mm, $\varepsilon_{r3} = 1.2$, whereas the variable parameters are as follows: side length of *E*-shaped patch is L_1, length of upper square patch is L_2, height between two patches is h_2, length of slot is l_s, width of slot is w_s, position of slot is p_s, and feed location is x_f. All these parameters were utilized in simulation to develop training and testing data. The previously published work has helped a lot to decide the range of dimensions [20, 76]. The developed neural network models are shown in Figure 7 and the network parameters are listed in Table 1, respectively.

1.4.1.4 Results and discussion

After getting training and testing data from simulator, the network parameters like number of neurons and number of layers are fixed, and the network is made ready for training. The performance of ANN was estimated for each epoch as per the observed error curve. The error curve for one of the ANN models (model-1) is shown in Figure 6. Steps for ANN training are repeated till the goal of least error is achieved. When an error criterion is met, the values of weights and biases of the trained ANN are recorded.



(b)



(a)

Figure 6: ANN error performance plots.

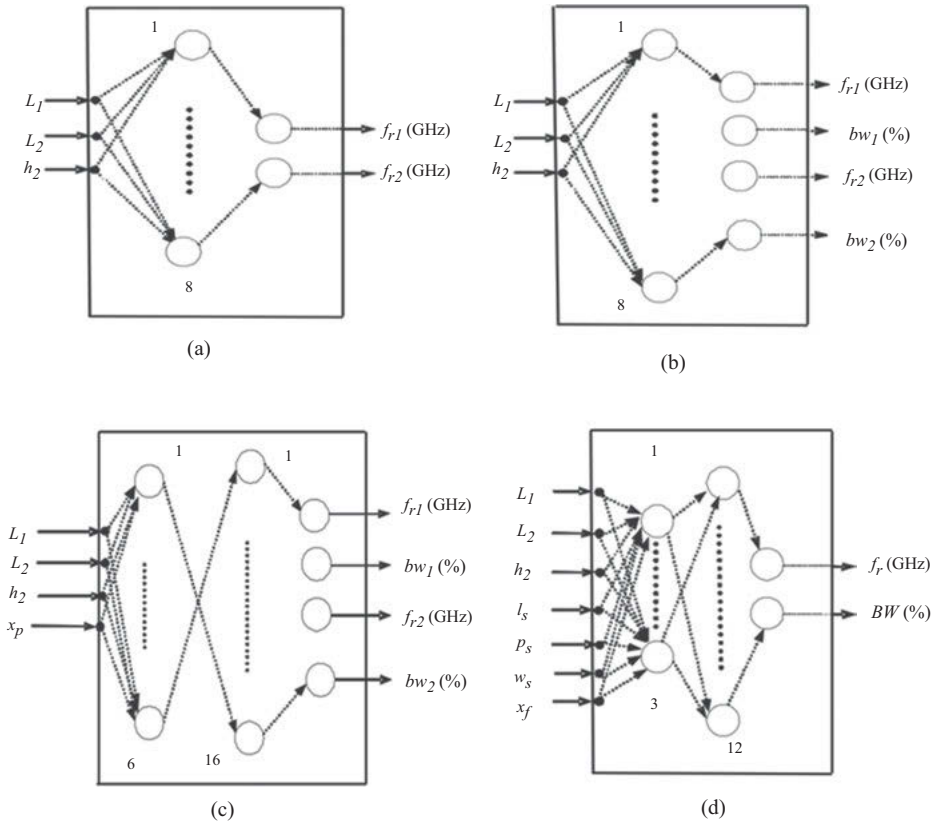


Figure 7: Developed ANN models.

In order to make the task easy and fast, trained ANN models that are able to reflect the relationship between the outputs and inputs were developed, namely, antenna frequency and/or bandwidth response with its geometrical variables. Depending on the requirement, four different models are developed. The first network was developed in the X–Ku band to give the position of the multiple frequencies for varying patch dimensions (L_1, L_2) and air-gap height (h_2). In the second case, bandwidth at the multiple resonant frequencies was added as another response of the network. The third network was developed in 2–6 GHz range to analyze the stacked patch antenna for WLAN applications. The fourth ANN model was developed with an aim to analyze single resonance broadband antennas. The validity of all ANNs was verified with simulation and experimental results. The ANN performance of all four models can be cross-checked from Table 2, where NP and NR stand for "no parameter" and "no result,"' respectively. In future, these trained ANNs are to be utilized to form the fitness function; those can be merged with optimizer algorithm to design custom-made stacked patch antennas for various applications.

Table 1: ANN model parameters.

S. no.	Parameters	ANN model parameter values			
		Model-1	**Model-2**	**Model-3**	**Model-4**
1	Size of network	3 × 8 × 2	3 × 8 × 4	4 × 6 × 16 × 4	7 × 3 × 12 × 2
2	Total training samples	220	371	300	300
3	Input parameter range	L_1 (6–11 mm) L_2 (4–10 mm) h_2 (0.3–10 mm)	L_1 (6–11 mm) L_2 (4–10 mm) h_2 (0.3–10 mm)	x_p (3–8 mm) h_2 (0.3–5 mm) L_2 (10–20 mm) L_1 (20–40 mm)	w_s (2–5) p_s (4–20) x_f (7–20) L_2 (20–50 mm) L_1 (20–80 mm) l_s (15–60 mm) h_2 (5–15 mm)
4	Total epochs	130	170	200	2,000
5	Rate of learning	0.72	0.72	0.72	0.7
6	Coefficient of momentum	0.72	0.72	0.72	0.9
7	Average training error	10×10^{-4}	1.2×10^{-4}	1.6×10^{-6}	3.2×10^{-4}

1.4.2 Heuristic approach-based particle swarm optimization algorithm

PSO is one of the simple and powerful heuristic approach-based algorithms that is stochastic in nature and is able to act as an optimizer to solve various design problems. It is being preferred over traditional methods because it is less prone to convergence to a weak local optimum. This optimization method has found useful applications in antenna engineering [57]. As the name indicates, PSO is inspired by the social behavior of a flock of birds and insect swarms. This nontraditional optimization algorithm is becoming popular in antenna engineering optimization problems as it proved to be more efficient for solving complex optimization problems. Use of conventional electromagnetic simulators is based on the trial-and-error approach for designing antenna structures. In our proposed work, the importance of PSO is that it can effectively remove shortcoming of simulators.

In this section, the use of PSO heuristic algorithm for the design of various stacked patch antennas has been discussed. The design task was formulated as an optimization problem and was solved successfully using PSO [85–87]. Stacked patch antennas have been designed for various applications using the developed formulation, and its validity has been cross-verified with simulation and experimental measurements. In this work, classical PSO algorithm has been used. Although the detailed description of the PSO algorithm is provided in Appendix A [57], in this section we

Table 2: Performance of developed ANN models.

Model no.	Ant. no.	ANN inputs (mm)								Outputs				Source
		L_1	L_2	h_2	x_p	l_s	w_s	p_s	x_f	f_{r1}	bw_1	f_{r2}	bw_2	
1	1	8.5	5.0	12.0	NP	NP	NP	NP	NP	11.63	NR	16.31	NR	ANN
										11.64		16.33		Simulator
										11.82		15.75		Measured
	2	8.0	9.50	5.40	NP	NP	NP	NP	NP	12.86	NR	17.44	NR	ANN
										12.93		17.49		Simulator
										12.67		16.99		Measured
2	3	8.5	6.5	2.0	NP	NP	NP	NP	NP	11.48	17.48	16.25	6.38	ANN
										11.49	17.50	16.23	6.40	Simulator
										11.73	15.83	16.10	5.25	Measured
	4	8.86	8.26	0.34	NP	NP	NP	NP	NP	10.40	14.15	14.43	6.14	ANN
										10.36	14.18	14.46	6.10	Simulator
										10.52	15.16	14.7	3.67	Measured
3	5	37.10	15.25	0.45	4.20	NP	NP	NP	NP	5.23	5.14	5.68	5.33	ANN
										5.23	5.56	5.70	5.31	Simulator
										5.47	6.30	5.73	5.66	Measured
4	6	40.0	30	10	NP	32	4	14	13	2.66	29.50	NR	NR	ANN
										2.65	28.72			Simulator
										2.57	27.62			Measured

have briefly discussed its implementation procedure. Then the design process of some typical stacked patch antennas have been discussed with the use of developed ANN merged with PSO methodology.

In order to resume design activity, expected frequencies and/or bandwidth are inputted to the optimizer (PSO). Every location of the search region of the PSO is an optimum solution. The trained ANN merged with PSO estimates the response for each candidate of the initial population. The merit of each candidate of the initial population is evaluated from the value of fitness. The "fitness" is a factor that helps in estimating the closeness of a particular solution point from the true answer of the problem. If the fitness value of a particular location of solution in a search region is more, then this point is assumed to be closed to the true solution and hence the possibility of its movement to the second population level is also more. The fitness value is estimated by evaluating a fitness function. Fitness function is a cost function that determines the fitness of a solution. When using the PSO algorithm for defect identification, the fitness function is always a critical factor for defect localization and quantification. A well-behaved fitness function can minimize the discrepancy between the measured and model-predicted data effectively. In the present problem of designing stacked patch antenna at user-defined frequencies, we have opted for the following type of fitness function (eq. (2)):

$$F = (\text{Desired frequency}_1 - \text{Instantaneous frequency}_1)^2 + (\text{Desired frequency}_2 - \text{Instantaneous frequency}_2)^2 \quad (2)$$

Expected output, that is, frequencies, are the user-defined inputs to the system, whereas the instantaneous frequencies are searched by the trained ANNs, merged with the PSO loop. After several iterations of the PSO, all candidates of the population converge almost to a single location and the fitness function touches to its minimum possible value within the tolerance limit. When this situation is met, PSO is assumed to give the optimized design parameters as the response.

1.4.2.1 PSO algorithm

PSO is one of the forms of AIA, which is characterized by the motion and reconnaissance of flock (swarm). In order to apply this algorithm, bunch of random solutions (population) is selected initially within the prespecified range of each variable to be optimized. In the terminology of proposed algorithm, each potential solution has been recognized as a "particle" (Appendix A, Figure 15). These particles are subjected to fly within the prespecified solution territory, where solution is expected to reside. As the iterative process continues, while moving each particle can keep a record in memory about its earlier best place (p_{best}) as per the their numerical values. The fitness function helps in fulfilling the objective of finding optimized variables. The best fitness value among all, viz. appropriate value of variable, is chosen as g_{best}.

The iterative process can be halted by fixing the specific number of iterations or by using suitable another stopping criteria. After iteration process is over, the variable numerical values in terms of g_{best} are assumed to be optimized parameters. While algorithm runs, all the particles are moved with certain fixed velocity which is updated using the following equation (eq. (3)):

$$v_{n+1} = w \times v_n \times c_1 \times \text{rand}(.) \times (p_{\text{best},n} - x_n) + c_2 \times \text{rand}(.) \times (g_{\text{best},n} - x_n) \tag{3}$$

where v_n is the velocity of the particle in the nth iteration and x_n is the coordinate of the particle in the nth iteration. The parameter w is known as the inertial weight, and this number (chosen to be between 0.0 and 1.0) determines as to what extent the particle remains along its original course unaffected by the pull of g_{best} and p_{best}; Two factors c_1 and c_2 determine the relative pull of the p_{best} and g_{best}, and rand(.) is a random function in the range [0,1]. Once the velocity has been determined, it is easy to move the particle to its next location. The velocity is applied for a given time step Δt as follows (eq. (4)):

$$x_{n+1} = x_n + \Delta t \times v_n \tag{4}$$

During this iterative process, the particles gradually settle down to an optimum solution. The PSO technique is simple to apply, easy to code, and is capable of solving difficult multidimensional problems efficiently. It has already been applied successfully for solving many electromagnetic problems [73, 74]. In the following sections, we have discussed the design procedure of a few typical antenna structures using the developed methodology.

1.4.2.2 Design of dual-band stacked patch antennas

To develop a dual-band stacked patch antenna, a structure under design consideration is shown in Figure 1. For the proposed problem, PSO through its artificial intelligence helps to locate the optimized values of antenna geometrical parameters (L_1, L_2, and h_2) so that it can resonate at end-user specified frequencies within X–Ku band just reducing the fitness function value till the minimum error is reached. That is done by trained ANN model-1 [87]. The most critical aspect of PSO execution is the construction of the fitness function so that it can fulfill the optimizer requirement. With the help of ANN model-1, the fitness function is constructed as eq. (5). The fitness function has been constructed based on the previous experience, and previously published literature has helped a lot in this regard:

$$F = \left[(F_{R1} - f_{r1})^2 + (F_{R2} - f_{r2})^2 \right] \tag{5}$$

In this equation, two desired resonant frequencies and PSO-located frequencies during the heuristic search in the solution territory are as follows: F_{R1}, F_{R2}, f_{r1}, and f_{r2}, respectively. In a case when a specified value of frequency difference (say

4.5 GHz) is required between the two resonant frequencies, then little modification in fitness function is done as follows (eq. (6)):

$$F = \left[(F_{R1} - f_{r1})2 + (F_{R2} - f_{r2})^2\right] \times \frac{(f_{r2} - f_{r1} - x)^2}{(f_{r2} - f_{r1})} \tag{6}$$

The role of first part in above equation is to locate the two resonant frequencies at two different desired locations within X–Ku band (8–18 GHz), whereas the second part locks the required gap within two frequencies till the predecided value ($x = 4.5$ GHz). As the iteration progresses, the PSO gradually settle down to the final optimized values of geometrical parameters of stacked patch antenna. The flowchart of the entire implementation process is shown in Figure 8. The very first step is to pick the geometrical parameters L_1, L_2, and h_2 that need to be optimized within a specified range, in which the optimal solution is to be searched. The range of each parameter is decided by the range of variable parameters, which are used in simulations to create a training dataset for neural network modeling. In this particular problem, three parameters (L_1, L_2, and h_2) need to be optimized; hence, the problem is three dimensional. The minimum and maximum values of each parameter will form the boundary for solution territory. Within this boundary, only all the particles will keep on moving. Various locations of particles in a solution territory are referred by numerical values of parameters (L_1, L_2, and h_2).

As a very first step to begin with, the population of particles is fixed, which helps to find out the optimized solutions. After that, position and velocity are initialized for each particle in a random manner. The initially fixed particle's position is its p_{best}. Out of these p_{best} values, the overall best and appropriate values (L_1, L_2, and h_2) are g_{best}. Then each particle is motivated to move within the predefined solution territory. The PSO algorithm starts working on each and every particle applying its contingence and cognitive intelligence and tries to find the fitness value using eq. (6), depending on the application. If the present fitness value is larger than the previous one, then the previous is replaced by the latest one. Particle's velocity of the whole generation is changed as per the updated values of p_{best} and g_{best}, applying eq. (3) in such a way that the particles jump toward the best fitness values. Once the velocity is fixed, the particle's next position is decided by using eq. (4). The whole process of jumping and moving is repeated till new p_{best} and g_{best} are found, and termination criteria are met. For this particular problem of dual-band antenna parameter optimization, the algorithm parameters are as follows: number of particles, $N = 20$ particles, numbers of iterations = 800. Minimum error expected was of the order of 10^{-4}. From Figure 9, it is clear that the AIA converged within 500 numbers of iterations.

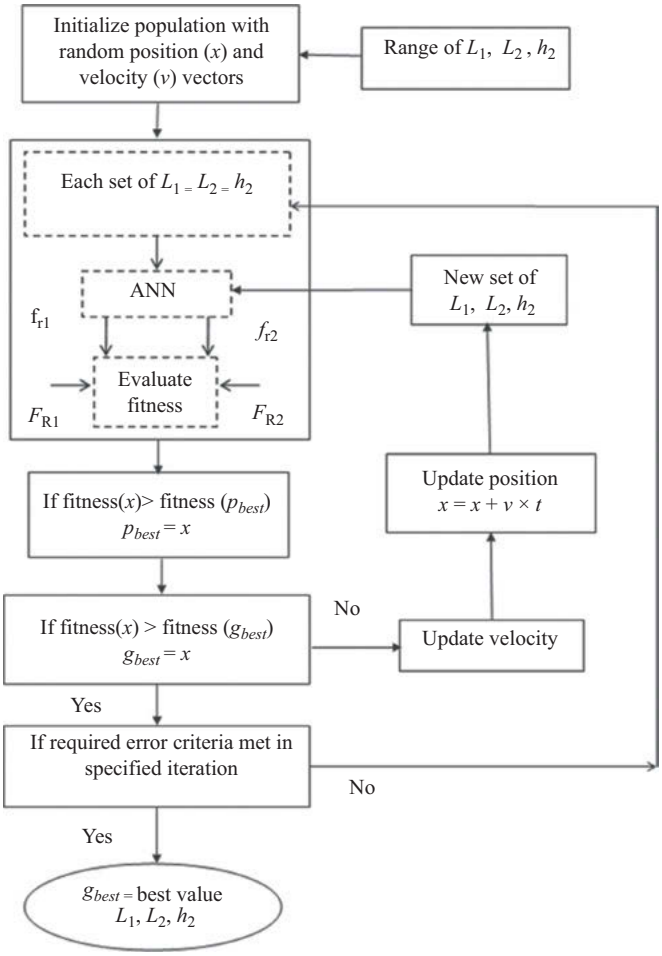


Figure 8: ANN-merged PSO algorithm. Reprinted with permission from ref. [87]. Copyright 2021, Springer Nature.

1.4.2.3 Results

The developed formulation was used to design five dual resonance stacked patch antennas in X–Ku band. Desired frequencies were given as input to the optimizer (PSO) and the corresponding outputs, which are the three optimized geometrical parameters, L_1, L_2, and h_2 of the stacked patch antenna were noted. These three optimized parameters (L_1, L_2, and h_2) along with other constant parameters (h_1 = 1.53 mm, $\varepsilon_{r1} = \varepsilon_{r3} = 2.2$, $\varepsilon_{r2} = 1$, $h_3 = 3.06$ mm, $x_p = y_p = 2.5$ mm) are used to simulate the stacked patch antenna structure and for antenna fabrication in the laboratory. Results for five typical structures are shown in the plots of Figure 10. Comparison of numerical values of the desired frequencies with the resonant frequencies of the

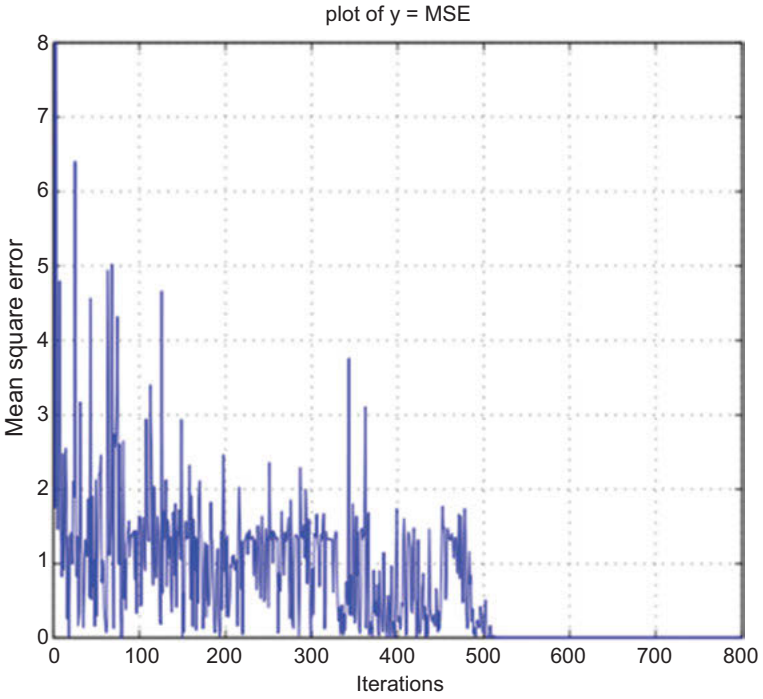


Figure 9: Error convergence performance plot.

obtained optimized structures is shown in Table 3. From the $|S_{11}|$ (dB) plots, very close similarity is reflected between measurement results and simulation results. The simulated and experimental results check the validity of the developed formulation and reflect the accuracy of the proposed method. Here, it has to be kept in mind that we have not taken the feedpoint as a variable parameter in the present application. It is fixed at, $x_p = y_p = 2.5$ mm. In the $|S_{11}|$ plots, although there is a bit discrepancy in resonance behavior of measurement plot and simulation plot, they have same resonating frequencies. Some of the discrepancies can be attributed to fabrication errors, especially the difficulty in achieving the specified value of air gap which was realized by using Teflon spacers.

1.4.2.4 Design of dual-band stacked patch antennas with specified bandwidth

There are situations where in a multiband antenna, the specific amount of bandwidth is required at each operational frequency. In this section, we have discussed the design of such type of antennas with the help of developed trained ANN model-2. This ANN model was trained for multiple frequencies with the associated bandwidth (%) of the stacked patch antenna in X–Ku band. The design implementation procedure for this case is same as discussed in the previous section. The only

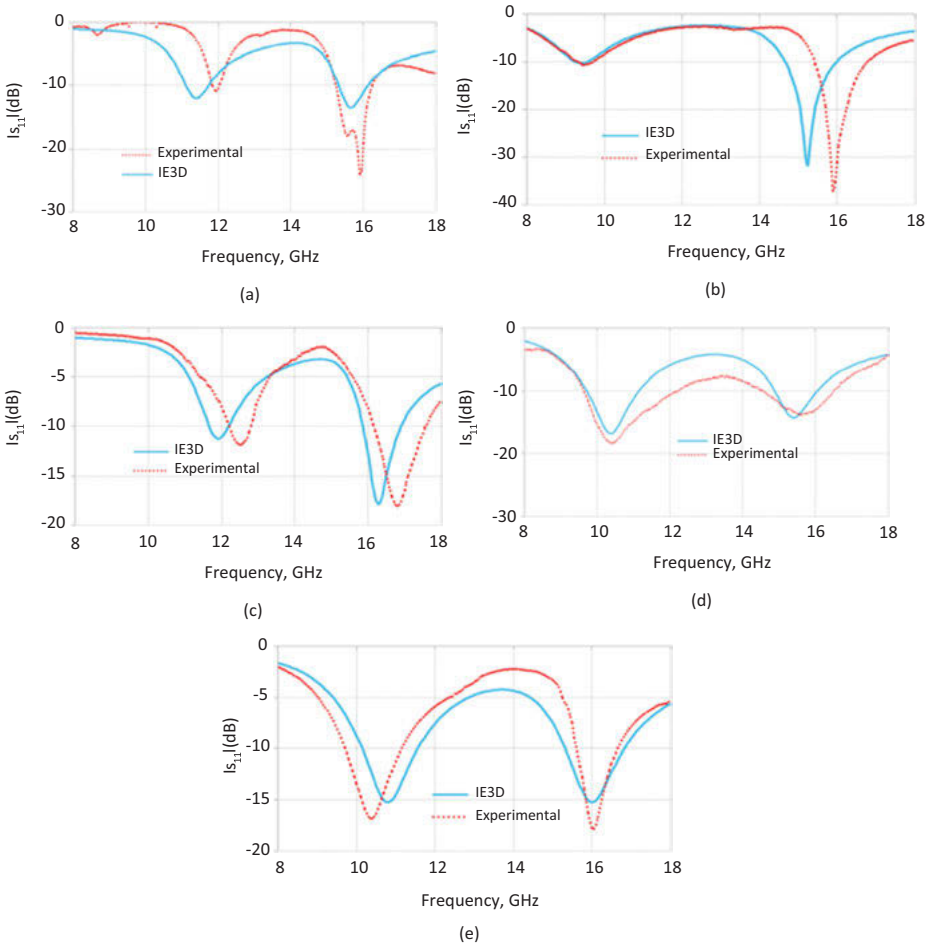


Figure 10: Comparison of simulation and measured $|S_{11}|$ for the optimized stacked patch antennas 1–5 at specified frequencies: Reprinted with permission from ref. [87]. Copyright 2021, Springer Nature.
(a) PSO-optimized parameters for antenna 1: $L_1 = 8.7$ mm, $L_2 = 7.08$ mm, $h_2 = 9.42$ mm
(b) PSO-optimized parameters for antenna 2: $L_1 = 7.75$ mm, $L_2 = 9.46$ mm, $h_2 = 1.15$ mm
(c) PSO-optimized parameters for antenna 3: $L_1 = 8.44$ mm, $L_2 = 7.01$ mm, $h_2 = 8.38$ mm
(d) PSO-optimized parameters for antenna 4: $L_1 = 8.35$ mm, $L_2 = 8.30$ mm, $h_2 = 0.79$ mm
(e) PSO-optimized parameters for antenna 5: $L_1 = 8.00$ mm, $L_2 = 7.96$ mm, $h_2 = 0.9$ mm

change occurs in framing the fitness function. The following fitness function (eq. (7)) was framed for the design of stacked patch antennas at desired frequencies with specified bandwidth:

$$F = \left[M \times (F_{R1} - f_{r1})^2 \times (BW_1 - bw_1)^2\right] + \left[N \times (F_{R2} - f_{r2})^2 \times (BW_2 - bw_2)^2\right] \quad (7)$$

where "f_r," "bw," "F_R," and "BW" are the heuristically searched parameters by the proposed algorithm (instantaneous frequency and bandwidth) and user input parameters (design frequency and bandwidth), respectively. There are two tuning constants, that is, "M" and "N"; these help in convergence within a limited number of iterations, so that computer resources can be utilized effectively. There is no specific method available to fix the values of these constants whose value usually lies in [0, 1] range. A guess on the values of these constants can be made by observing the contribution of each term of the fitness function on F after certain number of iterations. If the contribution from one term is less, then that term can be penalized by multiplying it with a higher value [88]. In the proposed work, the values of tuning constants, that is, M and N, have been fixed as 0.8 and 0.2, respectively. Here it may be emphasized that, for a particular problem, the fitness function is not unique and may vary from user to user. This fitness function was minimized with respect to the antenna design parameters L_1, L_2, and h_2. After several iterations, on meeting the stopping criterion, the iterative process of the PSO settles down to the optimized solution.

1.4.2.5 Results

The ANN model-2 was developed so that it can work well within the frequency X–Ku band. Therefore, the present design process is applicable only for the analysis and design of antenna that can work only within the specified frequency range. Similarly the highest specified bandwidths were restricted to 17% and 8% for the first (lower) and second (upper) band of frequency, respectively, because of the few constants and fixed value decided in geometrical parameters like substrate etc. Total six numbers of multiband antennas with specified resonance frequency and bandwidths are designed using the proposed approach. The optimized design parameters are L_1, L_2, and h_2, and fixed parameter values are $h_1 = 1.53$ mm, $\varepsilon_{r1} = \varepsilon_{r3} = 2.2$, $\varepsilon_{r2} = 1$, $h_3 = 3.06$ mm, and $x_p = y_p = 2.5$ mm. These optimized and fixed parameters are used to simulate the structures using electromagnetic simulator software. After that antennas are designed in the RF and microwave laboratory. The reflection performance plots for the total six antennas are shown in Figure 11. The computational and experimental measured results can be compared from these performance plots. The numerical values of optimizer response, along with the experimentally measured and simulation results have been listed in Table 3. Expected two parameters, that is, resonance frequencies and bandwidth, can be easily compared with experimentally measured and simulation results. The comparison helps in validation of the models.

1.4.2.6 Design of WLAN dual-band stacked patch antennas

In this design application, we have applied the same strategy used earlier to design stacked patch antennas for license-free dual-band (2.4 and 5 GHz) listed as follows:

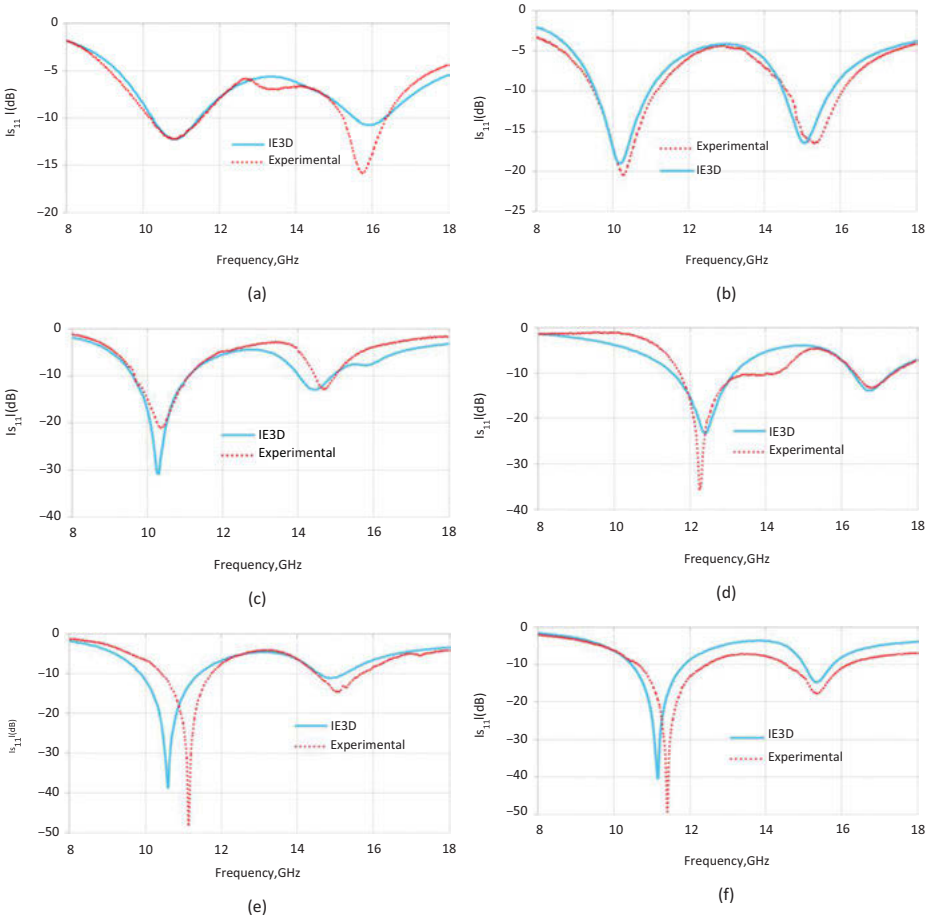


Figure 11: Comparison of simulation and measured $|S_{11}|$ for the optimized stacked patch antennas 1–5 at specified frequencies with the specified bandwidth: Reprinted with permission from ref. [89]. Copyright 2021, Cambridge University Press.
(a) PSO-optimized parameters for antenna 1: $L_1 = 8.25$ mm, $L_2 = 7.55$ mm, $h_2 = 1.3$ mm
(b) PSO-optimized parameters for antenna 2: $L_1 = 8.35$ mm, $L_2 = 8.66$ mm, $h_2 = 0.64$ mm
(c) PSO-optimized parameters for antenna 3: $L_1 = 8.86$ mm, $L_2 = 8.26$ mm, $h_2 = 0.34$ mm
(d) PSO-optimized parameters for antenna 4: $L_1 = 8.33$ mm, $L_2 = 6.2$ mm, $h_2 = 3.62$ mm
(e) PSO-optimized parameters for antenna 5: $L_1 = 8.98$ mm, $L_2 = 4.28$ mm, $h_2 = 0.67$ mm
(f) PSO-optimized parameters for antenna 6: $L_1 = 9$ mm, $L_2 = 5.5$ mm, $h_2 = 4$ mm

2,403–2,483 and 5,150–5,250 MHz; and quad-band listed as follows: 5,150–5,250, 5,250–5,350, 5,725–5,825, and 5,825–5,850 MHz. These are useful for WLAN applications [90–93]. For a quad-band stacked patch antenna design, we have merged four bands into dual broadband (5,150–5,350 and 5,725–5,850 MHz) and PSO was implemented for this. ANN model-3 is also helpful to develop the fitness function

required to solve this particular problem through the PSO. In addition to the patch dimensions (L_1, L_2) and air-gap height (h_2), the feeding point ($x_p = y_p$) was also optimized. The same fitness function (eq. (7)) was used here to design stacked patch WLAN antennas [94, 95].

1.4.2.7 Results

A typical dual-band WLAN antenna was designed for 2.44 and 5.2 GHz center frequencies. Similarly, other antennas were designed for license-free frequency band (5,150–5,350 and 5,725–5,850 MHz) with 5.25 and 5.79 GHz center frequencies. In order to cross-verify the results, the optimizer outputs (L_1, L_2, h_2, $x_p = y_p$) along with fixed parameters were used to simulate the proposed antenna in simulator software. Two antennas were fabricated in the laboratory, and measurements were done for reflection coefficient. The graphical comparison between the $|S_{11}|$ (reflection coefficient) plots of simulated and measured results are shown in Figure 12. Table 3 compares the simulated and experimental results with the desired values that show the validity of our methodology.

1.4.2.8 Design of broadband stacked patch antennas

Proposed stacked patch antennas will act as a single resonance broadband antenna or dual-band antenna. It will depend on the size and distances between used patches. In the same structure when slots are created in lower patch, then antennas will have one *E*-shaped radiating patch and it will perform as a single broadband antenna [76, 77]. Single resonance broadband stacked patch antennas of structure shown in Figure 12 have been designed using the ANN model-4 and developed methodology. Different dimensions of *E*-shape and other patches like side length L_1, length L_2, height h_2, slot length l_s, slot width w_s, slot position p_s, and location of feed x_f were optimized by the ANN-merged PSO for the design of single broadband antenna structure. The following fitness function (eq. (8)) was framed for this particular problem:

$$F = \left[(F_R - f_r)^2 \times (\mathrm{BW} - \mathrm{bw})^2 \right] \tag{8}$$

where "f_r," "bw," "F_R," and "BW" are the instantaneous values of the resonance frequency, bandwidth searched by the optimizer algorithm within solution territory, user-defined single resonance frequency, and associated bandwidth, respectively. Because the ANN model-4 was developed to work within 1–10 GHz, so broadband antennas in this range have been designed, with the restriction of maximum 45% bandwidth because of the constant values of some of the design parameters chosen in our work.

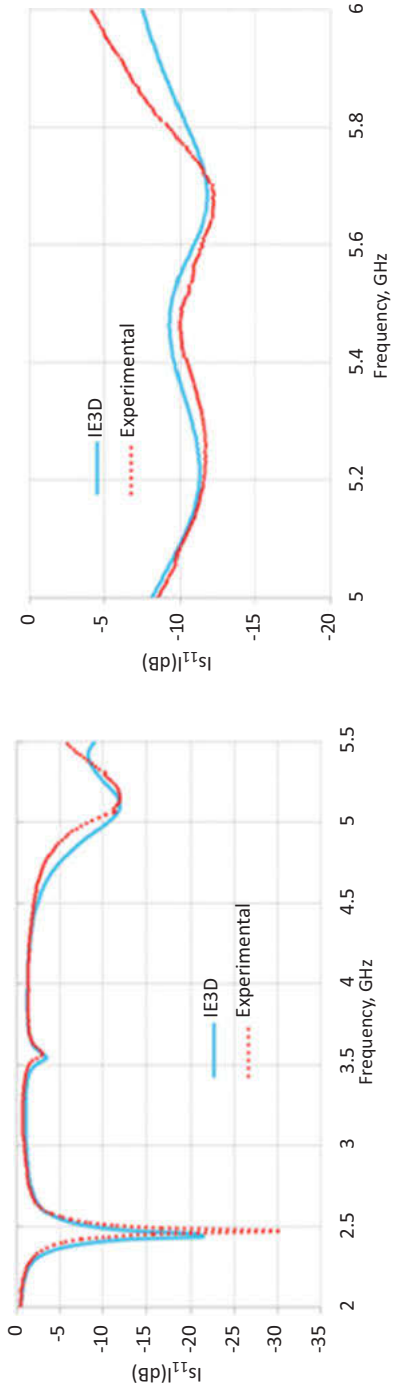


Figure 12: Comparison of simulation and measured $|S_{11}|$ for the optimized dual-band stacked patch antenna for WLAN applications.
(a) PSO-optimized parameters for antenna 1: $L_1 = 38.12$ mm, $L_2 = 13.52$ mm, $h_2 = 1.75$ mm, $x_p = y_p = 4.98$ m.
(b) PSO-optimized parameters for antenna 2: $L_1 = 37.1$ mm, $L_2 = 15.25$ mm, $h_2 = 0.45$ mm, $x_p = y_p = 4.2$ m.

1.4.2.9 Results

Two single resonance broadband antennas were designed using the same proposed methodology. With the help of optimizer outputs (L_1, L_2, h_2, l_s, w_s, p_s, x_f) and constant parameter ($h_1 = 10.8$ mm, $\varepsilon_{r1} = \varepsilon_{r3} = 1.2$, $\varepsilon_{r2} = 1$, $h_3 = 5.4$ mm) simulations were done using software. For the same parameters, antennas were fabricated in the laboratory and measurements were carried out. The simulation and measurement results can be used for performance comparison. $|S_{11}|$ (reflection coefficient) plots of these structures are shown in Figure 13. The numerical values of the simulation and experimental results for these structures are listed in Table 3.

1.5 Summary

The classical method of designing stacked patch antenna using commercial simulators was attempted as an optimization problem and solved using AIAs (ANN merged with PSO). The most required fitness function was constructed with the support of trained ANN. This approach helps in discarding contiguous use of simulator and whole design process becomes fast. Several custom-made stacked patch antennas were designed and developed through the advised approach. The prototypes of designed and developed antennas are shown in Figure 14. These designed and developed antennas can be applied for different applications depending on the training range of the neural network. Five different dual-band stacked patch antennas were designed to work in X–Ku band range. The size of the radiating patches along with the air-gap height between them was optimized parameters. In order to extend the idea of design, two additional bandwidth parameters were added in the resonance frequency outputs. Three different geometrical parameters (length of two patches and air-gap height) were optimized in X–Ku band. A total of six antennas were designed using the same approach to verify the design methodology. In continuation of this, dual-band and quad-band (dual broadband) WLAN antennas to work at 2.4 and 5 GHz bands were also designed. In that design process, the feed-point position ($x_p = y_p$) was also optimized in addition to other antenna design parameters like lengths of two patches and air-gap height. At last, two antennas having *E*-shaped patch were designed to work at single broadband (till 45%) frequency band (1–10 GHz). The results obtained from the AIA-based methodology were cross-checked with experimentally measured and simulation results. The experimentally measured and simulated results support the validity of the developed methodology. The major reason of using these AIA is that the number of variables in a stacked patch antenna is large as compared to those in a single-layer simple patch antenna. The unique nature of the AIA helps in handling large number of variables.

The main objective behind the proposed work is to evolve a user-friendly device for analysis and design of the multipatch antennas through artificial intelligence–based optimization. Although this objective has been achieved by developing a novel

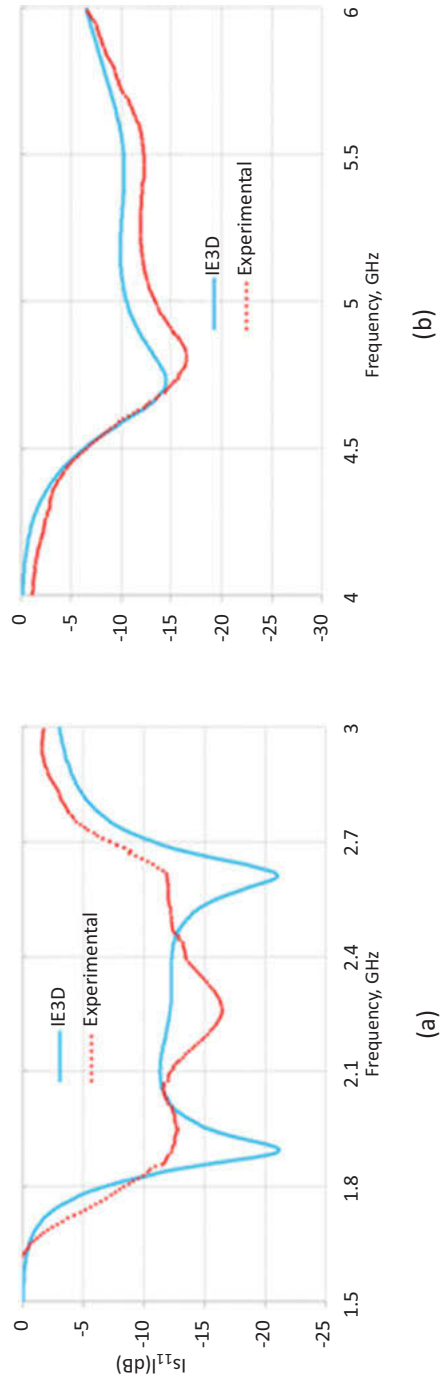


Figure 13: Comparison of simulation and measured $|S_{11}|$ for the optimized single resonance broadband stacked patch antenna-1. PSO-optimized parameters.
(a) $L_1 = 52.23$ mm, $L_2 = 41.13$ mm, $h_2 = 16.03$ mm, $l_s = 41.32$ mm, $w_s = 5.23$ mm, $p_s = 15.03$ mm, $(x_f, y_f) = (18.40, 0)$
(b) $L_1 = 20.07$ mm, $L_2 = 15.23$ mm, $h_2 = 17.2$ mm, $l_s = 14.97$ mm, $w_s = 2.52$ mm, $p_s = 5.22$ mm, $(x_f, y_f) = (6.50, 0)$

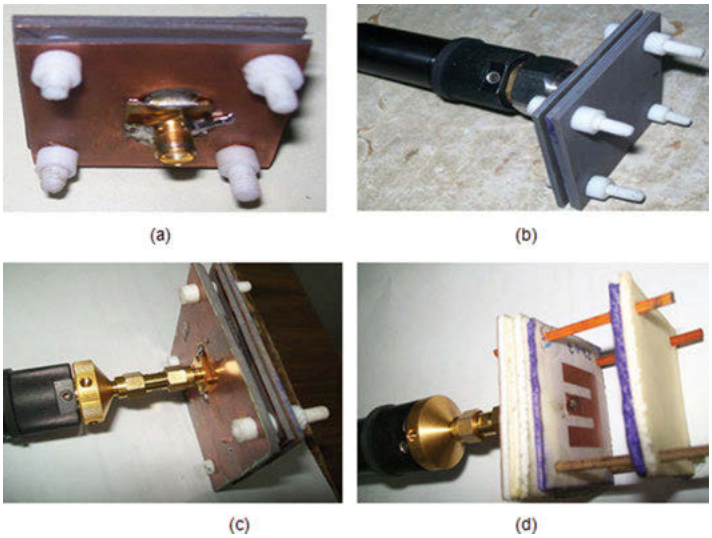


Figure 14: Prototype design of antennas.
(a) Dual-frequency PSO-optimized antenna, (b) dual-frequency broadband PSO-optimized antenna, (c) WLAN PSO-optimized antenna, and (d) single resonance broadband PSO-optimized antenna.

design method by embedding PSO in a neural network, the methodology needs further refinement:

- The developed design methodology can be tried for the design of other microwave components. Because the classical methods of designing of microwave components are cumbersome, the developed design methodology can be tried for reducing the computational burden considerably.
- Efforts should be made to embed the developed design method in the commercially available antenna simulators to avoid the present demerits of adjusting the parameters in a trial-and-error way. This may drastically reduce the computation time of the simulators.
- Depending on the space available for the antenna inside a communication system, the designer might opt for radiating structures of other regular and irregular shapes besides the rectangular or square one. Efforts should be made to expand the scope of the present design procedure for other regular radiating patches. In addition to that, there should be provisions to compare the performance of stacked patch antennas with varied radiating patch structures, depending on the stringent requirement of the user. Fuzzy rule-based systems may be tried for this type of problems.

Table 3: Performance of AIA-based optimizer.

Ant. no.	Desired frequencies (GHz) and associated bandwidth (%)				PSO-optimized dimensions of the antenna (mm)								Resonant frequencies and bandwidth of optimized antenna structure (GHz and %) (simulated) (measured)			
	F_{R1}	BW_1	F_{R2}	BW_2	L_1	L_2	h_2	x_p	l_s	w_s	p_s	x_f	f_{r1}	bw_1	f_{r2}	bw_2
1	11.75	NP	15.5	NP	8.70	7.08	9.42	NP	NP	NP	NP	NP	11.52 11.95	NR	15.72 15.73	NR
2	9.80	NP	15.5	NP	7.75	9.46	1.15	NP	NP	NP	NP	NP	9.65 9.70	NR	15.32 15.90	NR
3	12.0	NP	16.5	NP	8.44	7.01	8.38	NP	NP	NP	NP	NP	11.92 12.33	NR	16.25 16.82	NR
4	10.5	NP	15.5	NP	8.35	8.30	0.79	NP	NP	NP	NP	NP	10.42 10.70	NR	15.46 15.53	NR
5	11.0	NP	16.0	NP	8.0	7.96	0.90	NP	NP	NP	NP	NP	10.86 10.76	NR	16.04 16.01	NR
6	11.0	12.0	16.0	8.0	8.25	7.55	1.30	NP	NP	NP	NP	NP	10.85 10.85	11.49 11.59	15.92 15.89	6.34 7.40
7	10.25	14.5	15.5	8.5	8.35	8.66	0.64	NP	NP	NP	NP	NP	10.28 10.34	13.8 15.16	15.12 15.30	10.60 11.14
8	10.50	15.0	15	6.0	8.86	8.26	0.34	NP	NP	NP	NP	NP	10.36 10.52	14.18 16.16	14.46 14.70	8.54 6.59

(continued)

Table 3 (continued)

Ant. no.	Desired frequencies (GHz) and associated bandwidth (%)				PSO-optimized dimensions of the antenna (mm)								Resonant frequencies and bandwidth of optimized antenna structure (GHz and %) (simulated) (measured)			
	F_{R1}	BW_1	F_{R2}	BW_2	L_1	L_2	h_2	x_p	l_s	w_s	p_s	x_f	f_{r1}	bw_1	f_{r2}	bw_2
9	12.35	14.0	16.80	7.0	8.33	6.20	3.62	NP	NP	NP	NP	NP	12.36 12.66	13.23 14.76	16.82 17.0	6.14 6.09
10	10.50	14.0	15.0	7.0	8.98	4.28	0.67	NP	NP	NP	NP	NP	10.65 11.07	14.65 11.9	14.90 15.10	6.2 7.8
11	11.25	14.5	15.5	5.5	9.0	5.5	4.0	NP	NP	NP	NP	NP	11.15 11.31	13.58 16.41	15.40 15.15	5.04 6.1
12	2.44	3.27	5.2	1.92	38.12	13.52	1.75	4.98	NP	NP	NP	NP	2.45 2.483	3.08 3.03	5.11 5.25	2.89 3.84
13	5.25	6	5.79	5	37.1	15.25	0.45	4.2	NP	NP	NP	NP	5.23 5.47	5.56 6.30	5.70 5.73	5.31 5.66
14	2.5	45	NP	NP	52.23	41.13	16.03	NP	41.32	5.23	15.03	18.40	2.27 2.21	NR	38.54 34.84	NR
15	5	22	NP	NP	20.07	15.23	17.20	NP	14.97	2.52	5.22	6.50	5.11 5.14	NR	20.2 21.30	NR

Appendix A Algorithm: particle swarm optimization algorithm

Step 1: *Define the solution space*: Number of parameters are chosen which need to be optimized, and the range is to be decided in terms of minimum and maximum values in each dimension of solution space.

Step 2: *Define a fitness function*: A fitness function is framed as per the analytical approach using trial-and-modify basis in such a way that it represents the single value of goodness of a solution. For different optimization of different problems, different solution spaces with different fitness function are to be chosen.

Step 3: *Initialize particles with random location and velocities*: Each particle moves at a random location with random velocity initially.

Step 4: *Fly particles with random velocity and position*: Each particle is moved in a random location with random velocity in a predefined range of solution space. Then the algorithm acts on each and every particle one by one and covers the whole swarm.

a) *Evaluate the particle's fitness and compute* g_{best} *and* p_{best}: The fitness value of each particle is calculated on each and every location. When its current fitness value found better than its p_{best}, then p_{best} is replaced by this and it becomes p_{best}. When its current fitness value found better than its g_{best}, then g_{best} is replaced by this and it becomes g_{best}.

b) *Update particle velocity*: The velocity of the particle is changed as per the location of g_{best} and p_{best}. Particles move in the direction of greatest fitness as per the velocity update in eq. (3).

c) *Update particle position*: After the velocity is determined, move the particle in a next location as per eq. (4).

Step 5: *Repeat*: After the procedure is performed on each particle and if the desired error criteria get satisfied or fitness value is achieved, then terminate the procedure; otherwise, go to the above step and repeat step-4(a)–step-4(c).

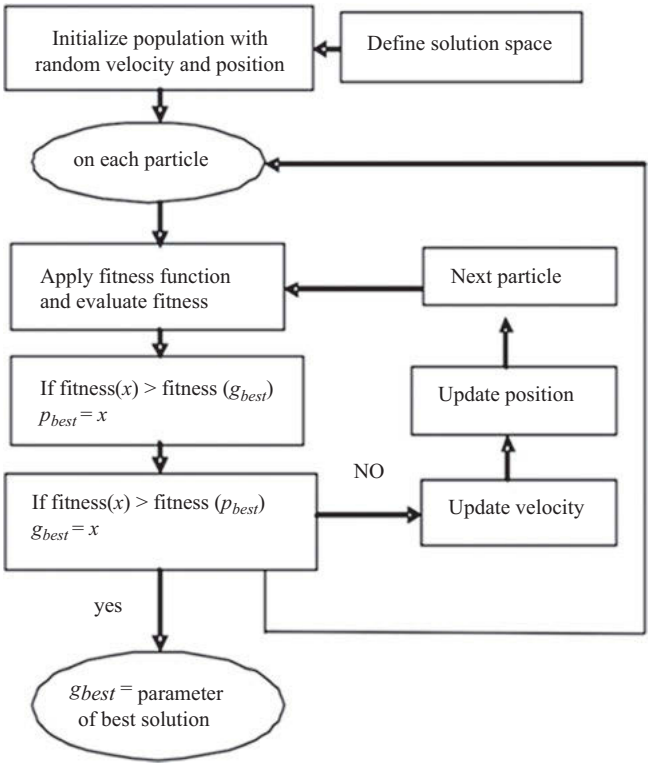


Figure 15: PSO flowchart.

References

[1] I. J. Bahl and P. Bhartia, *Microstrip Antennas*, Artech House, Dedham, MA, 1980.
[2] D. M. Pozar and D. H. Schaubert, *Microstrip Antennas: The Analysis and Design of Microstrip Antennas and Arrays*, IEEE Press, New York, 1995.
[3] IE3D Zeland Software, Incorporated, IE3D.
[4] G. Kumar and K. P. Ray, *Broadband Microstrip Antenna*, Artech House, Boston London, 2003.
[5] S. K. Jain and S. V. Charhate Stacked patch antennas design: Latest trend in research, *Proceedings of the International Conference on Microwaves, Antenna Propagation and Remote Sensing (ICMARS-2015), Jodhpur, Rajasthan, India*, 2015, 238–244.
[6] R. Bancroft, *Microstrip and Printed Antenna Design*, Prentice-Hall India, New Delhi, 2006.
[7] J. R. James and P. S. Hall, *Handbook of Microstrip Antennas*, Peter Peregrinus Ltd, Dedham MA London, 1989.
[8] R. Garg, P. Bhartia, I. Bahl, and A. Ittipiboon, *Microstrip Antenna Design Handbook*, Artech House, Dedham MA London, 2001.

[9] J. Anguera, G. Font, C. Puent, C. Borja, and J. Soler, Multi frequency microstrip patch antenna using multiple stacked elements, *IEEE Microwave Wireless Component Letter*, 13(3), 123–124, March 2003.
[10] K. S. Fong, H. F. Pues, and M. J. Withers, Wideband multilayer coaxial fed microstrip antenna element, *Electronics Letter*, 21(11), 497–498, April 1985.
[11] S. D. Targonski, R. B. WaterHouse, and D. M. Pozar, Wideband aperture coupled stacked patch antenna using thick substrates, *Electronics Letter*, 32(21), 1941–1942, October 1996.
[12] R. B. Waterhouse, Stacked patches using high and low dielectric constant material combinations, *IEEE Transactions on Antennas and Propagation*, 47(12), 1767–1771, December 1999.
[13] A. Mitchell, M. Leech, D. M. Kokotoff, and R. Waterhouse, Search for high-performance probe-fed stacked patches using optimization, *IEEE Transactions on Antennas and Propagation*, 51 (2), 249–255, February 2003.
[14] J. Wang, R. Fralich, C. Wu, and J. Litva, Multifunctional aperture coupled stack patch antenna, *Electronics Letter*, 26(25), 2067–2068, December 1990.
[15] D. M. Pozar and S. M. Duffy, A dual-band circularly polarized aperture coupled stacked microstrip antenna for global positioning satellite, *IEEE Transactions on Antennas and Propagation*, 45(11), 1618–1624, November 1997.
[16] J. Anguera, C. Puente, C. Borja, N. Delbene, and J. Soler, Dual-frequency broad-band stacked microstrip patch antenna, *IEEE Antennas Wireless Propagation Letter*, 2(25), 36–39, 2003.
[17] P. S. Bhatnagar, J. P. Daniel, K. Mahdjoubi, and C. Terret, Experimental study on stacked triangular microstrip antenna, *Electronics Letter*, 22(16), 864–865, July 1986.
[18] B. L. Ooi and C. L. Lee, Broadband air-filled stacked U-slot patch antenna, *Electronics Letter*, 35(7), 515–517, April 1999.
[19] J. P. Damiano, J. Bennegueouche, and A. Papiernik, Study of multilayer microstrip antennas with radiating elements of various geometry, *Proceedings of the IEE Conference*, 137(3), 163–170, June 1990.
[20] F. Croq and D. M. Pozar, Millimeter wave design of wide-band aperture-coupled stacked microstrip antenna, *IEEE Transactions on Antennas and Propagation*, 39(12), 1770–1776, December 1991.
[21] J. B. Ooi, S. Qin, and M. S. Leong, Novel design of broad-band stacked patch antenna, *IEEE Transactions on Antennas and Propagation*, 50(10), 1391–1395, October 2002.
[22] R. B. Waterhouse, Broadband stacked shorted patch, *Electronics Letter*, 35(2), 98–100, January 2002.
[23] J. Ollikainen, M. Fischer, and P. Vainikainen, Thin dual-resonant stacked shorted patch antenna for mobile communications, *Electronics Letter*, 35(6), 437–438, March 1999.
[24] R. Chair, K. M. Luk, and K. F. Lee, Measurement and analysis of miniature multilayer patch antenna, *IEEE Transactions on Antennas and Propagation*, 50(2), 244–250, February 2002.
[25] R. Chair, K. M. Luk, and K. F. Lee, Miniature multilayer shorted patch antenna, *Electronics Letter*, 36(1), 3–4, Jan 2000.
[26] H. K. Kan, R. B. Waterhouse, J. Lee, and D. Pavlickovski, Dual frequency stacked shorted patch antenna, *Electronics Letter*, 41(11), 10–11, May 2005.
[27] T. Fortaki, L. Djouane, F. Chebara, and A. B. Halia, On the dual-frequency behavior of stacked microstrip patches, *IEEE Transaction Wireless Propagation Letter*, 7(11), 310–313, April 2008.
[28] K. Ghorbani and R. B. Waterhouse, Dual polarized wide-band aperture stacked patch antennas, *IEEE Transactions on Antennas and Propagation*, 52(8), 2171–2174, August 2004.
[29] R. Q. Lee, T. Talty, and K. F. Lee, Circular polarization characteristics of stacked microstrip antennas, *Electronics Letter*, 26(25), 2109–2110, December 1990.

[30] J. G. Tagle and C. G. Christodoulou, Extended cavity model analysis of stacked microstrip ring antennas, *IEEE Transactions on Antennas and Propagation*, 45(11), 1626–1635, 1997.
[31] G. G. Gentiti, L. E. G. Castillo, M. S. Palma, and F. P. Martinez, Green's function analysis of single and stacked rectangular microstrip patch antennas enclosed in a cavity, *IEEE Transactions on Antennas and Propagation*, 45(4), 573–579, November 1997.
[32] T. M. Sze, Y. Mingwu, C. Yinchao, and R. Mitra, Finite difference time-domain analysis of a stacked dual-frequency microstrip planar inverted-F antenna for mobile telephone handsets, *IEEE Transactions on Antennas and Propagation*, 49(3), 367–376, 2001.
[33] W. S. T. Rowe and R. B. Waterhouse, Theoretical investigation on the use of high permittivity materials in microstrip aperture stacked patch antennas, *IEEE Transactions on Antennas and Propagation*, 51(9), 2484–2486, 2003.
[34] J. F. Zurcher and F. E. Gardiol, *Broadband Patch Antennas*, Artech House, Norwood, M.A, 1995.
[35] K. L. Wong, *Compact and Broadband Microstrip Antennas*, Wiley, 2002.
[36] J. M. Johnson and Y. R. Samii, A systematic design method to obtain broadband characteristics for singly-fed electromagnetically coupled patch antennas for circular polarization, *IEEE Transactions on Antennas and Propagation*, 51(12), 3239–3248, 2003.
[37] M. Edimo, K. Mahdjoubi, A. Sharaiha, and C. Terret, Simple circuit model for coax-fed stacked rectangular microstrip patch antenna, *Proceedings of the IEE Microw. Antennas Propagation*, 145(3), 268–272, June 1998.
[38] A. N. Tulintseff, S. M. Ali, and J. A. Kong, Input impedance of a probe-fed stacked circular microstrip antenna, *IEEE Transactions on Antennas and Propagation*, 39(3), 381–390, 1991.
[39] S. S. Zhong, G. Liu, and G. Qasim, Closed form expressions for resonant frequency of rectangular patch antennas with multidielectric layers, *IEEE Transactions on Antennas and Propagation*, 42(9), 1360–1363, September 1994.
[40] M. Kirschning and R. H. Jansen, Accurate model for effective dielectric constant of microstrip with validity up to millimeter-wave frequencies, *Electronics Letter*, 18(6), 272–273, March 1982.
[41] A. Patnaik and R. K. Mishra, ANN techniques in microwave engineering, *IEEE Microw. Mag.1*, 1, 55–60, March 2000.
[42] V. K. Devabhaktuni, M. C. Yagoub, and Q. J. Zhang, A robust algorithm for automatic development of neural network models for microwave applications, *IEEE Transactions on Microwave Theory Techniques*, 49(12), 2282–2291, December 2001.
[43] J. Chakrawarty, S. K. Jain, and S. Sharma Review on artificial neural network applied to RF and microwave domain, *Proceedings of the International Conference on Microwaves, Antenna Propagation and Remote Sensing (ICMARS-2017)*, Jodhpur, Rajasthan, India, 51–56, 2017.
[44] J. Chakrawarty, S. K. Jain, and S. Sharma Artificial neural network applied to aperture coupled microstrip patch antenna analysis, *Proceedings of the International Conference on Microwaves, Antenna Propagation and Remote Sensing (ICMARS-2017)*, Jodhpur, Rajasthan, India, 204–208, 2017.
[45] K. Guney, S. Sagiroglu, and M. Erler, Generalized neural methods to determined resonant frequencies of various microstrip antennas, *International Journal of RF and Microwave Computer-Aided Engineering*, 12(1), 131–139, December 2001.
[46] Y. Kim, S. Keely, J. Ghosh, and H. Ling, Application of artificial neural networks to broadband antenna design based on a parametric frequency model, *IEEE Transactions on Antennas and Propagation*, 55(3), 669–674, March 2007.
[47] R. K. Mishra and A. Patnaik, Neural network – based CAD model for the design of square – patch antennas, *IEEE Transactions on Antennas and Propagation*, 46(12), 1890–1891, December 1998.

[48] R. K. Mishra and A. Patnaik, Neurospectral computation for complex resonant frequency of microstrip resonators, *IEEE Microwave Guided Wave Letter*, 9(9), 351–353, September 1999.
[49] D. K. Neog, S. S. Patnaik, D. C. Panda, S. Devi, B. Khuntia, and M. Dutta, Design of wideband microstrip antenna and the use of artificial neural networks in parameter calculation, *IEEE Antennas and Propagation Magazine*, 47(3), 60–65, June 2005.
[50] K. Guney and S. S. Gultekin, Artificial neural networks for resonant frequency calculation for rectangular microstrip antennas with thin and thick substrates, *International Journal of Infrared Millimeter Waves*, 25(9), 1383–1391, September 2004.
[51] A. Patnaik, D. Anagnostou, C. G. Christodoulou, and J. C. Lyke, Neurocomputational analysis of multiband reconfigurable planar antenna, *IEEE Transactions on Antennas Propagation*, 53 (11), 3453–3458, November 2005.
[52] A. Patnaik, D. Anagnostou, C. G. Christodoulou, and J. C. Lyke, Modeling frequency reconfigurable antenna array using neural networks, *Microwave and Optical Technology Letters*, 44(4), 351–354, February 2005.
[53] S. Sagiroglu and K. Guney, Calculation of resonant frequency for an equilateral triangular microstrip antenna with the use of artificial neural networks, *Microwave and Optical Technology Letters*, 14(2), 89–93, February 1997.
[54] N. P. Somasiri, X. Chen, and A. A. Rezazadeh, Neural network modeler for design optimization of multilayer patch antennas, *Proceedings of the IEE Microwaves, Antennas Propagation Conference*, 151(6), 514–518, December 2004.
[55] S. K. Jain and S. Jain, Performance analysis of coaxial fed stacked patch antennas, *Frequenz Journal of RF-Engineering and Telecommunications*, 2014, 1–12.
[56] S. K. Jain, Analysis of multivariable patch antenna using artificial neural network. *Proceedings of the International Conference on Microwaves, Antenna Propagation and Remote Sensing (ICMARS-2014)*, Jodhpur, Rajasthan, India, 251–254, 2015.
[57] J. Robinson and Y. R. Samii, Particle swarm optimization in electromagnetic, *IEEE Transactions on Antennas and Propagation*, 52(2), 397–407, February 2004.
[58] C. A. Balanis, *Antenna Theory: Analysis and Design*, Wiley Inc., New York, 2003.
[59] V. Devaraj, K. K. Ajayan, and M. R. Baiju A novel optimization technique for a stacked patch antenna, *Proceedings of the Asia-Pacific Microwave Conference*, 111–114, 2007.
[60] J. Kennedy and R. Eberhart Particle swarm optimization, *Proceedings of the IEEE Int. Conf. Neural Networks IV*, Lecture Notes in Computer Science 4628, Piscataway, NJ, 1942–1948, 1995.
[61] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm Intelligence*, Morgan Kaufmann Publishers, San Francisco, CA, 2001.
[62] R. C. Eberhart and Y. Shi, Particle swarm optimization: Developments, applications and resources, *Proceedings of the Evolutionary Computation*, 1, 2001, 81–86.
[63] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, John Wiley and Sons, South Africa, 2001.
[64] N. Jin and Y. R. Samii, Particle swarm optimization for antenna designs in engineering electromagnetic, *Journal of Artificial Evolution and Applications*, 1–10, 2008.
[65] S. Xu and Y. R. Samii, Boundary conditions in particle swarm optimization revisited, *IEEE Transactions on Antennas and Propagation*, 55(3), 760–765, March 2007.
[66] N. Jin and Y. R. Samii, Advances in particle swarm optimization for antenna designs: Real number, binary, single-objective and multiobjective implementations, *IEEE Transactions on Antennas and Propagation*, 55(3), 556–566, 2007.
[67] X. Hu and R. Eberhart Solving constrained nonlinear optimization problems with particle swarm optimization, *Proceedings of the 6th World Multi-Conf. Systemics, Cybernetics Informatics*, Orlando, USA, 203–206, July 2002.

[68] A. Ratanveera, S. K. Halgamuge, and H. C. Watson, Self hierarchical particle swarm optimizer with time varying acceleration coefficients, *IEEE Transactions on Evolutionary Computing*, 8 (3), 240–254, June 2004.
[69] R. C. Eberhart and Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, *Proceedings of the Congress Evolutionary Computation*, San Diego, CA, 84–88, 2000.
[70] I. C. Trelea, The particle swarm optimization algorithm: Convergence analysis and parameter selection, *Information Processing Letter*, 85(6), 317–325, December 2002.
[71] M. Clerc and J. Kennedy, The particle swarm: Explosion stability and convergence in a multi-dimensional complex space, *IEEE Transactions on Evolutionary Computing*, 6(1), 58–73, February 2002.
[72] A. E. Yilmaz and M. Kuzuoglu, Calculation of optimized parameters of rectangular microstrip patch antenna using particle swarm optimization, *Microwave and Optical Technology Letters*, 49(12), 2905–2907, December 2007.
[73] V. S. Chintakindi, S. S. Pattnaik, O. P. Bajpai, S. Devi, P. K. Patra, and K. M. Bakwad Resonant frequency of equilateral triangular microstrip patch antenna using particle swarm optimization techniques, *Proceedings of the Recent Advances in Microwave* Theory *and Applications* Microwave *2008 Conference,* 20–22, 2008.
[74] N. Jin and Y. R. Samii, Parallel particle swarm optimization and finite time-domain (PSO/FDTD) algorithm for multiband and wide-band patch antenna designs, *IEEE Transactions on Antennas and Propagation*, 53(11), 3459–3468, November 2005.
[75] A. Z. Hood and E. Topsakal, Particle swarm optimization for dual-band implantable antennas, *IEEE Transactions on Antennas and Propagation*, 52(10), 3209–3212, November 2007.
[76] A. Bzeih, S. A. Chahine, K. Y. Kabalan, A. E. Hajj, and A. Chehab Empirical formulation and design of a broadband enhanced E-patch antenna, *Proceedings of the National Radio Science, Egypt NRSC* 07, 1–9, March 2007.
[77] F. Yang, X. X. Zhang, X. Xiaoning, and Y. R. Samii, Wideband E-shaped patch antennas for wireless communications, *IEEE Transactions on Antennas and Propagation*, 49(7), 1094–1100, July 2001.
[78] Z. Ma, V. Volski, and G. A. Vandenbosch, Optimal design of highly compact low-cost and strongly coupled 4 element array for WLAN, *IEEE Transactions on Antennas and Propagation*, 59(3), 1061–1065, March 2011.
[79] S. Haykin, *Neural Networks – A Comprehensive Foundation*, Prentice Hall of India, New Delhi, 2004.
[80] P. M. Watson and K. C. Gupta, EM-ANN models for design of CPW patch antennas, *Proceedings of the IEEE Antennas Propagation International Symposium*, 2(12), 648–651, June 1998.
[81] Q. J. Zhang and K. C. Gupta, *Neural Networks for RF and Microwave Design*, Artech House, London, 2000.
[82] C. Christodoulou and M. Georgiopoulos, *Application of Neural Networks in Electromagnetics*, Artech House, London, 2001.
[83] S. K. Jain, S. N. Sinha, and A. Patnaik, Analysis of coaxial fed dual patch multilayer X-Ku band antenna using artificial neural networks, *Proceedings of the International Symposium Biologically Inspired Computing Applications*, Bhubaneswar, India, 1111–1114, December 2009.
[84] J. M. Zurada, *Introduction to Artificial Neural Systems*, Jaico Publishing House, Mumbai, 2006.
[85] S. K. Jain, A. Patnaik, and S. N. Sinha Neural network based particle swarm optimizer for design of dual resonance X-Ku band stacked patch antenna, *Proceedings of the IEEE Antennas and Propagation Symposium*, Spokane, Washington, USA, 2932–2935, July 2011.

[86] D. P. Rini, S. M. Samsuddin, and S. S. Yuhaniz, Particle swarm optimization: Techniques, system, and challenges, *International Journal of Computer Applications*, 14(1), 19–27, January 2011.
[87] S. K. Jain, A. Patnaik, and S. N. Sinha, Design of custom-made stacked patch antennas – A artificial intelligent approach, *International Journal of Artificial Intelligence and Cybernetics (Springer)*, 4(3), 189–194, March 2012.
[88] K. Deb, *Optimization for Engineering Design Algorithms and Examples*, Prentice-Hall India, New Delhi, 2003.
[89] S. K. Jain, Bandwidth enhancement of patch antennas using neural network dependent modified optimizer, *International Journal of Microwave and Wireless Technology, Cambridge University Press and the European Microwave Association*, 4, 1–9, 2015.
[90] J. M. Ju, G. T. Jeong, J. Han, K. Y. Won, and K. S. Kwak, Design of multiple U-shaped slot microstrip patch antenna for 5-GHz band WLANs, *Microwave and Optical Technology Letters*, 43(6), 487–488, December 2004.
[91] R. K. Gupta and G. Kumar, High gain multilayered antenna for wireless application, *Microwave and Optical Technology Letters*, 50(7), 1923–1928, July 2008.
[92] K. G. Thomas and M. Sreenivasan, A simple dual-band microstrip-fed printed antenna for WLAN applications, *Journal of* IET Microwaves Antennas & Propagation, 3(4), 687–693, 2009.
[93] WLAN License-Free Bands, (Accessed July 2018, at http://en.wikipedia.org/wiki/Listof-WLAN-channels)
[94] S. K. Jain and S. Jain, Neurocomputational analysis of coaxial fed stacked patch antennas for satellite and WLAN applications, *International Journal of Progress in Electromagnetic Research (PIERC)*, 42, 125–135, 2013.
[95] S. Mohine and S. K. Jain, Neural network analysis of WLAN dual band miniaturized stacked patch antenna, *Proceedings of the IEEE Workshop on Computational Intelligence Theories, Applications and Future Directions*, I.I.T. Kanpur India, 18–22, 2013.

Shreeyansh Singh Yadav, Abhinav Sharma, Abhishek Sharma, Arpit Jain

Direction of arrival estimation using Lévy flight-based moth flame optimization algorithm

Abstract: In the field of 4G/5G communication, an important area of research is estimating the direction of incoming signals. The direction of narrow band sources can be determined using different spectral and eigenstructure techniques. When the signal-to-noise ratio (SNR) remains minimal and the channel is coherent, these methods fail to predict signal direction. Maximum likelihood (ML) is a statistical direction of estimation technique that overcomes the limitations of conventional algorithm and precisely discoveries signals in adverse conditions. ML approximation is estimated by minimalizing the complex log-likelihood function through indeterminable parameters. In this chapter, author proposed the modified Lévy flight mechanism-based moth flame optimization algorithm (LVMFO) to estimate the signal direction in low SNR environment. Moth flame optimization is a swarm intelligence algorithm that has good exploitation capability but has poor exploration capability; therefore, Lévy flight mechanism is incorporated in MFO to improve the exploration capability. The proposed improved LVMFO algorithm outperforms CAPON, MUSIC, and sine–cosine algorithm in terms of root mean square error and probability of resolution.

Keywords: ML, MFO, LVMFO, SCA, RMSE, PR

1 Introduction

Optimization is the act of finding the finest result or optimal solution underneath the given condition. Many flaws in the reality can be viewed as optimization challenges. Over the last several decades, various approaches have been developed to

Shreeyansh Singh Yadav, Department of Electrical and Electronics Engineering, Development, University of Petroleum and Energy Studies, Dehradun, India, e-mail: 500069251@stu.upes.ac.in
Abhinav Sharma, Department of Electrical and Electronics Engineering, University of Petroleum and Energy Studies, Dehradun, India, e-mail: abhinavgbpuat@gmail.com
Abhishek Sharma, Research and Development, University of Petroleum and Energy Studies, Dehradun, India, e-mail: abhishek15491@gmail.com
Arpit Jain, WhiteHat Education Technology Pvt. Ltd., India, e-mail: arpit.eic@gmail.com

https://doi.org/10.1515/9783110734652-005

provide global optimum solution for diverse set of problems. "Metaheuristic" [1] are high-level algorithms that discover optimal solution and provide a satisfactory explanation for an optimization problem, especially when computational complexity is high. Metaheuristic algorithms are categorized as single solution and population-based algorithms. It has minimal assumptions about how the optimization delinquent will be solved; therefore, it might be useful for a wide range of problems. In comparison to the optimization algorithm, metaheuristics do not guarantee that for a given collection of problems, a globally optimal solution can be discovered. Metaheuristics conduct a thorough investigation of the search area in order to identify the most effective solutions. These algorithms are approximate and nondeterministic and efficiently discover the optimal solution in search space. Metaheuristics use some sort of optimization to ensure that the solution is based on a collection of randomly generated variables. Metaheuristics may frequently identify good answers with less computing work than optimization approaches by looking across a broad collection of solutions. In recent years, metaheuristic optimization algorithms have grown in prominence as a technique of optimization. Genetic algorithms [2], particle swarm optimization [3], ant colony optimization [4], ant lion optimization [5], differential evolution [6], evolutionary programming [7], cuckoo search [8], firefly algorithm [9], and dragonfly algorithm [10] are some of the most popular algorithms in this domain.

Moth flame optimization (MFO) [11] is a swarm intelligence algorithm ended up replicating the transverse orientation steering mechanism used by moths in nature. Moths fly in the dark by upholding a static angle with orientation to the Moon, which is a strangely operative tactic for drifting extensive distances in a conventional line. Since the Moon is far away from the moth, this methodology assumes that the moth will fly in a conventional line. The method, however, is simple, flexible, and precise to implement, but it has a slower rate of convergence and hence has a natural tendency to become trapped in a local optimum solution. An improved MFO algorithm that incorporates Lévy flight mechanism [12] is proposed in this chapter. Lévy flight technique has high ability to attract the performance of the optimization algorithms in exploration and exploitation stages as well as absconding from local optima in the search space.

In the domain of wireless communication, direction of arrival (DOA) estimation [13–14] and adaptive beamforming [15–16] are two important aspects that play a vital role in smart antenna [17–18] and multiple-input–multiple-output systems [19]. In this chapter, authors analyzed the performance of the proposed algorithm by optimizing the deterministic maximum likelihood (ML) [20] function for estimating the direction of signals in varying signal-to-noise ratio (SNR) environment. The summary of chapter is outlined as follows: the uniform linear array (ULA) data model for the designed issue is presented in Section 2. The traditional DOA estimation techniques are described in Section 3. MFO and the Lévy-flight-based MFO algorithm are briefly addressed in Section 4. The mathematical model of sine–cosine

algorithm (SCA) is explained in Section 5. Section 6 shows and examines the simulation results, and Section 7 concludes the chapter and highlights the scope of future work.

2 Data model

Let us assume an ULA with M sensing components, where the spacing between consecutive sensors is d which is defined as half the wavelength of the received signals, as shown in Figure 1. Assume that D narrow-band far-field signal sources with distinct DOAs imposed on ULA.

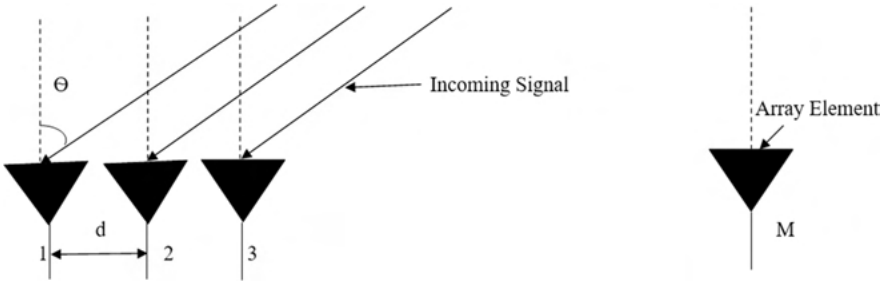


Figure 1: Uniform linear array.

The phase source is the first element of the ULA, as well as the array steering vector of size $M \times 1$ represents the ith user's direction as follows:

$$a(\theta i) = \left[1, z, \ldots, z^{M-1}\right]^T,\ z = e^{\left(\frac{j2\pi d}{\lambda}\right)\sin\theta i} \tag{1}$$

where the angle measured from the array broadside is denoted as θ_i and λ represents the carrier wavelength of the source signal. At tth time sample, the total signal received by the array elements is represented as

$$\bar{x}(t) = \bar{A}(\theta)\,{}^{*}\bar{s}(t) + \bar{n}(t) \tag{2}$$

where $\bar{s}(t)$ denotes the incident complex monochromatic signals, $\bar{n}(t)$ denotes the noise vector, and $\bar{A}(\theta)$ is the array steering matrix, which is composed of array steering vectors and is defined as follows:

$$\bar{A}(\theta) = [\bar{a}(\theta_1) \cdots \bar{a}(\theta_D)] \tag{3}$$

For DOA estimation in array signal processing, correlation matrices are utilized instead of the real array output $x(t)$; and the correlation matrix is defined by

$$\bar{R}_{xx} = E\left[x^{*}\bar{x}^{H}\right] = \bar{A}\bar{R}_{ss}\bar{A}^{H} + \bar{R}_{nn} \tag{4}$$

where $\bar{R}_{ss}$ represents the source and $\bar{R}_{nn}$ represents the noise correlation matrix, and $E[]$, $()^{H}$ denotes the expectation and Hermitian operation, respectively. The authors speculated the deterministic method for estimating the direction of arriving signals in this case. As a result, for the given unknown configurations, the signal vector is predictable. In the deterministic model, ML estimates the incoming signals by optimizing the multimodal function defined as follows:

$$f_{\mathrm{DML}} = \mathrm{tr}\left[\left(I_M - \bar{A}\left(\bar{A}^{H}\bar{A}\right)^{-1}\bar{A}^{H}\right)\bar{R}\right] \tag{5}$$

where tr[] represents the trace and I_M is the identity matrix of order $M \times M$.

3 Conventional DOA estimation algorithm

The customary angle estimation algorithms are well mathematically structured algorithm which shows substantial estimation results in deterministic channel environment. Nonsubspace algorithms assess the acquired signal direction by observing the peaks of the pseudospectrum; and the subspace algorithm, also called eigenstructure algorithms, estimates the direction of signals through eigendecomposition into signal and noise subspaces. Nonparametric spectral estimation algorithms such as Capon and Bartlett functionalities depend on how the weights are obtained for producing the pseudospectrum. MUSIC, Root-MUSIC, matrix pencil, Pisarenko harmonic decomposition, min-norm estimate, and ESPRIT are some of the key eigenstructure approaches that use orthogonal signal subspaces for DOA estimation.

3.1 CAPON

The Capon DOA estimate is also termed as minimum variance distortionless response. The goal is to utilize the signal-to-interference ratio (SIR) by transmitting the signal of interest in phase and amplitude without distortion. The source correlation matrix ($\overline{R_{ss}}$) is anticipated to be diagonal. This maximized SIR is achieved using a set of array weights ($\bar{w} = [w_1 w_2 \cdots w_M]^{T}$), where the array weights are

$$\bar{w} = \frac{\bar{R}_{xx}^{-1}\bar{a}(\theta)}{\bar{a}^{H}(\theta)\overline{R_{xx}}^{-1}\bar{a}(\theta)} \tag{6}$$

and the array correlation matrix $\bar{R}_{xx}^{-1}$ is used.

The pseudospectrum is as follows:

$$P_C(\theta) = \frac{1}{\bar{a}^H(\theta)\bar{R}_{xx}^{-1}\bar{a}(\theta)} \tag{7}$$

3.2 MUSIC

MUSIC is a subspace algorithm that exploits the eigenvalues of signal and noise subspaces to efficiently estimate the incoming signal direction. To determine the direction of incoming signals, one must know the number of signals. Let us assume there are D signals with D eigenvalues and eigenvectors and $M–D$ noise eigenvalues and eigenvectors for M number of array elements.

The mathematical model for MUSIC is as follows:

Step A: Estimate the array correlation matrix ($\overline{R_{xx}}$) considering the uncorrelated noise with equal variances.

Step B: For $\bar{R}_{xx}$, determine the eigenvalues and eigenvectors. The signals are represented by D eigenvectors, while the noise is represented by $M–D$ eigenvectors. The eigenvectors with the lowest eigenvalues are chosen. The lowest eigenvalues for uncorrelated signals are equal to the noise variance. The $M \times (M–D)$ dimensional subspace covered by the noise eigenvectors may then be constructed as follows:

$$\bar{E}_N = [\bar{e}_1\ \bar{e}_2 \cdots \bar{e}_{M-D}] \tag{8}$$

Step C: At the angles of arrival $\theta_1, \theta_2, \ldots, \theta_D$ the noise subspace eigenvectors are orthogonal to the array steering vectors. Because of this orthogonality, the Euclidean distance may be expressed as follows:

$$d^2 = \bar{a}(\theta)^H \bar{E}_N \bar{E}_N^H \bar{a}(\theta) = 0 \tag{9}$$

Step D: Find the pseudospectrum by placing the Euclidean distance expression in the denominator. The MUSIC pseudospectrum is given as follows:

$$P_{\mathrm{MU}}(\theta) = \frac{1}{\left|\bar{a}(\theta)^H \bar{E}_N \bar{E}_N^H \bar{a}(\theta)\right|} \tag{10}$$

Step E: Estimate the peaks of pseudospectrum to determine the direction of incoming signals.

4 Moth flame optimization

S. Mirjalili proposed MFO, a population-based stochastic optimization method, in 2015. Moths are opulent insects that are closely related to the butterfly family. This insect has over 160,000 distinct species all across the planet. Their lives are divided into two stages: larvae and adults. The larvae are transformed into moths by cocoons. The most fascinating aspect about moths is their unique night navigation capabilities. They have developed to fly at night with the help of moonlight. They navigate using a technique known as transverse orientation. Along these lines, a moth flies by keeping a consistent point with the Moon, which is an exceptionally successful technique for voyaging significant distances in an orderly manner. Since the Moon is so distant, this strategy ensures that the moth will fly in an orderly manner. Individuals can use the same navigating approach. Despite the effectiveness of transverse lighting, moths frequently fly in a spiral path around the light. In actuality, artificial lights deceive moths, leading them to behave in this manner. This is due to the insufficiency of the transverse orientation, which can only be used to move in a straight line when the light source is extremely far away. When moths see a man-made artificial light, they try to fly in a straight line by keeping their angle with the light the same. Because the light source is so close to the Moon, maintaining a similar angle to it produces a useless or dangerous spiral flight path for moths.

The moth is supposed to be the candidate solution to the issue in the MFO method, and the variable to be solved is the moth's position in space. Moths can fly in one, two, three, and even higher dimensional space by altering their position vectors. Because the MFO approach is fundamentally a global optimization methodology, the moth species may be expressed in the matrix as follows:

$$M = \begin{bmatrix} m_{1,1} & \cdots & m_{1,d} \\ \vdots & \ddots & \vdots \\ m_{n,d} & \cdots & m_{n,d} \end{bmatrix} \tag{11}$$

where n denotes the moth's number to solve and the number of control variables to solve is denoted by d. We further presume there is an array for storing the proper fitness values for each moth, as illustrated below:

$$OM = \begin{bmatrix} OM_1 \\ OM_2 \\ \cdot \\ \cdot \\ \cdot \\ OM_n \end{bmatrix} \tag{12}$$

where n addresses the complete number of moths.

MFO method needs every moth to adjust its area exclusively through the select flame that has a place with it to keep away from the strategy slipping into the neighborhood ideal arrangement. The framework's overall pursuit abilities are essentially improved over the time. Subsequently, the fire's and moth's areas in the pursuit cycle are both changing grids of a similar measurement:

$$F = \begin{bmatrix} f_{1,1} & \cdots & f_{1,d} \\ \vdots & \ddots & \vdots \\ f_{n,1} & \cdots & f_{n,d} \end{bmatrix} \tag{13}$$

In flames, it is expected that a persistent column of fitness value vectors exists, as shown below:

$$OF = \begin{bmatrix} OF_1 \\ OF_2 \\ \cdot \\ \cdot \\ \cdot \\ OF_n \end{bmatrix} \tag{14}$$

The appropriate approach of the variables in two matrices differs during the iteration phase. Moths are essentially searchers who move throughout the exploration interplanetary, and also flame is an ideal area for moths to return steadily. Each moth is encircled by a flame that matches its color, and the moths' locations are updated toward the flame with each generation. Using this method, the calculation can track down the worldwide most fitting answer. To direct out the exact demonstration for the flying conduct of moth drawn to, the following expression may be used to explain the updation procedure for the area inside every moth compared with a flame:

$$M_i = I\left(M_i, F_j\right) \tag{15}$$

where the *i*th moth is represented by M_i, the *j*th flame is represented by F_j, and the helical function is represented by *I*. The helical function is represented by the *j*th flame and *I*. The helical function fulfills the accompanying rules:

a. Primary argument of the function is determined by moth's starting planetary location.
b. Spiral's final point is the spatial location that corresponds to a modern flame.
c. The variation range of the spiral really should not surpass its search space.

The function is defined as follows under the given conditions:

$$I(M_i, F_j) = D_i e^{bt} \cos(2\pi t) + F_j \tag{16}$$

$$t = (x-1) * \text{rand} + 1 \tag{17}$$

$$x = -1 + \text{Iteration} * \left(\frac{-1}{T_{\max}}\right) \tag{18}$$

wherever ith moth's distance from the jth flame is represented by D_i, b is a constant, and t is a random number in the range [−1,1]. D_i is represented as follows:

$$D_i = \left|F_j - M_i\right| \tag{19}$$

The above flame position update system ensures the moth's potential to pursue locally round the flame. To upgrade the chances of tracking down a serviceable methodology, the ideal choice found in the cutting-edge age is utilized as the area of the following emphasis of moths encompassing the flame. The situation of the principal moth consistently is by all accounts refreshed regarding the flame with the most noteworthy wellness esteem, while the situation of the last moth has consistently been refreshed according to the flame with the least wellness esteem. On the off chance that each position update of n moths depends on n various areas in the hunt space, the calculation's neighborhood arranging potential will be diminished. To tackle this issue, a versatile procedure for diminishing the quantity of blazes is given, which permits the quantity of flames to be decreased iteratively during the iterative interaction, orchestrating the calculation's worldwide and nearby inquiry abilities in the hunt space. The method is as per the following equation:

$$\text{flame}_{\text{no}} = \text{round}\left(A - 1 * \frac{A-1}{F}\right) \tag{20}$$

where the current iteration count is l, the maximum number of flames is A, and the total iteration count is F.

4.1 Lévy flight mechanism

Lévy flight is a non-Gaussian stochastic approach that is identified with the Lévy stable dispersion and is a kind of arbitrary walk approach. Lévy flight is portrayed by a progression of little yet occasionally enormous advances that guarantee that moving components do not ceaselessly investigate a similar spot changing the system behavior over the course of time. In spite of the fact that its mobility direction is arbitrary, its movement step size exhibits exponential distribution. The following equation is the mathematical model of the Lévy flying mechanism:

$$L(s) \sim |s|^{-1-\beta} \tag{21}$$

where s represents the step length and β is the index which lies in the range [0, 2].

The step length is calculated as follows:

$$s = \frac{u}{|v|^{1/\beta}} \tag{22}$$

where u and v are normally distributed parameters and are specified as follows:

$$u \sim N\left(0, \sigma_u^2\right) \tag{23}$$

$$v \sim N\left(0, \sigma_v^2\right) \tag{24}$$

where

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\beta\Gamma[(1+\beta)/2]*2^{(\beta-1)/2}} \right\}^{1/\beta} \tag{25}$$

$$\sigma_v = 1 \tag{26}$$

where $\Gamma()$ represents the Gamma function and is defined as follows:

$$\Gamma(1+\beta) = \int_0^\infty t^\beta \exp^{-t} dt \tag{27}$$

If the value of β is an integer, then the Gamma function is defined as follows:

$$\Gamma(1+\beta) = \beta! \tag{28}$$

4.2 Lévy-flight-based MFO

Acquainting the Lévy flying strategy with the moth's search capability may adequately expand the moth's hunt space and can enhance moth global search capability. The MFO procedure combined with the Lévy flight mechanism can enhance the algorithm's area of service, enhance populace diversity, and allow the algorithm to escape the local optimal solution. Figure 2 portrays the Lévy-flight-based MFO algorithm.

5 Sine–cosine algorithm

In 2015, S. Mirjalili introduced the SCA algorithm, a stochastic optimization approach. The algorithm is inspired from the mathematical model of sine and cosine functions. In SCA, a random initial solution is produced in the search space which fluctuates over the ideal solution using sine and cosine functions. As a result, the

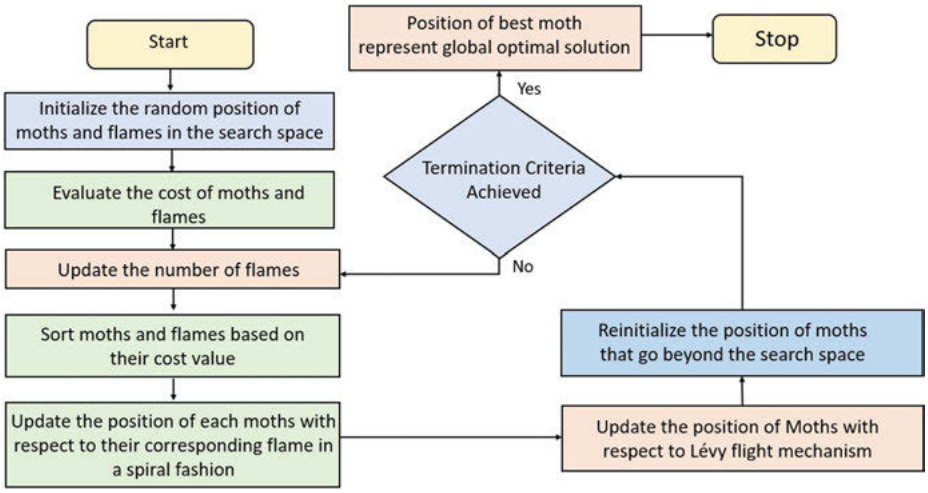


Figure 2: Process flow diagram of Lévy-flight-based MFO algorithm.

final optimal result is found by simply varying the sine and cosine functions. The position of the random solution is updated with respect to the following equations:

$$P_i^{t+1} = \begin{cases} P_i^t + r_1 * \sin(r_2) * \left| r_3 R_i^t - P_i^t \right|, r_4 < 0.5 \\ P_i^t + r_1 * \cos(r_2) * \left| r_3 R_i^t - P_i^t \right|, r_4 \geq 0.5 \end{cases} \tag{29}$$

where current iteration is denoted as t, R_i^t is the ith dimensional destination point position, and P_i^t is the ith dimensional position of current solution at iteration t. The random numbers are r_1, r_2, r_3, and r_4 . r_1 and r_3 have a uniform distribution between 0 and 2, r_2 obeys uniform distribution between 0 and 2π, and r_4 has uniform distribution between 0 and 1. The control parameter r_1 decreases linearly and helps the algorithm to switch from the global exploration to the local exploitation and adaptively changes over the course of iterations and is defined as follows:

$$r_1 = a\left(1 - {}^{t}/_{T}\right) \tag{30}$$

where a is a constant, t is the current iteration, and T is the maximum number of iterations. SCA is successfully used for finding the global optimal solution of constraint and unconstraint optimization problems.

6 Simulation results and discussion

This section shows and discusses the simulation results of DOA estimation using ML-LVMFO (Lévy flight mechanism-based moth flame optimization), ML-SCA CAPON

and MUSIC algorithms. Figure 3 depicts the procedure of LVMFO and SCA for DOA estimation. Two narrow band plane waves with arrival angles of 35° and 38° with 100 snapshots are directed toward 10 elements. ULA is considered for simulation in MATLAB software. Since the model is deterministic, the uncorrelated BPSK signals are simulated in the virtual environment.

With respect to the two conventional algorithms CAPON and MUSIC, the pseudospectrum's peaks are positioned in the [–90°, 90°] range and they signify the direction of incoming signals. LVMFO and SCA are stochastic algorithms; therefore, 100 Monte Carlo runs are considered to obtain the optimized value of the direction of incoming signals. The parameters considered for LVMFO and SCA are mentioned in Table 1.

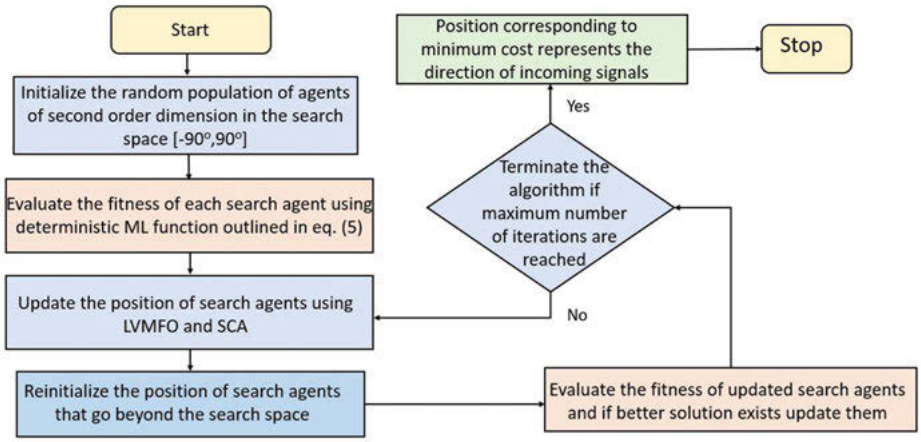


Figure 3: Implementation of LVMFO and SCA for estimating the direction of signals.

Table 1: Parameters considered for LVMFO and SCA.

Name	Population size	Parameters
MFOLevy	30	Constant parameter (b) = 1 Beta (β) = 1.4
SCA	30	Controlling parameter ($r1$) = [0,2]

The performance of the four mentioned methods is assessed by using RMSE and PR criteria, and since the ML-LVMFO and ML-SCA are iterative algorithms, their efficacy is also evaluated in terms of boxplot and rate of convergence.

6.1 Root mean square error

The difference between the actual direction and the estimated direction of signals is defined as the root mean square error (RMSE), which is stated as follows:

$$RMSE = \sqrt{\frac{1}{N_n N_{runs}} \sum_{x=1}^{N_{runs}} \sum_{y=1}^{N_n} \left[\hat{\theta}_y(x) - \theta_y\right]^2} \tag{31}$$

where N_{runs} represents the Monte Carlo runs, N_n denotes the total number of sources, $\hat{\theta}_y(x)$ represents the yth direction of signal in the xth run and θ_y denotes the true direction of the yth signal.

Figure 4 illustrates the variation of RMSE for CAPON, MUSIC, ML-LVMFO, and ML-SCA with respect to SNR in the range [−30, 20] dB. The result shows that ML-LVMFO algorithm presents best results with respect to ML-SCA, and conventional algorithms till 16 dB SNR while at SNR higher than 16 dB MUSIC algorithm outperforms all three algorithms. Therefore, ML-LVMFO algorithm produces the best result and can be utilized for estimating the direction of signals in low SNR environment.

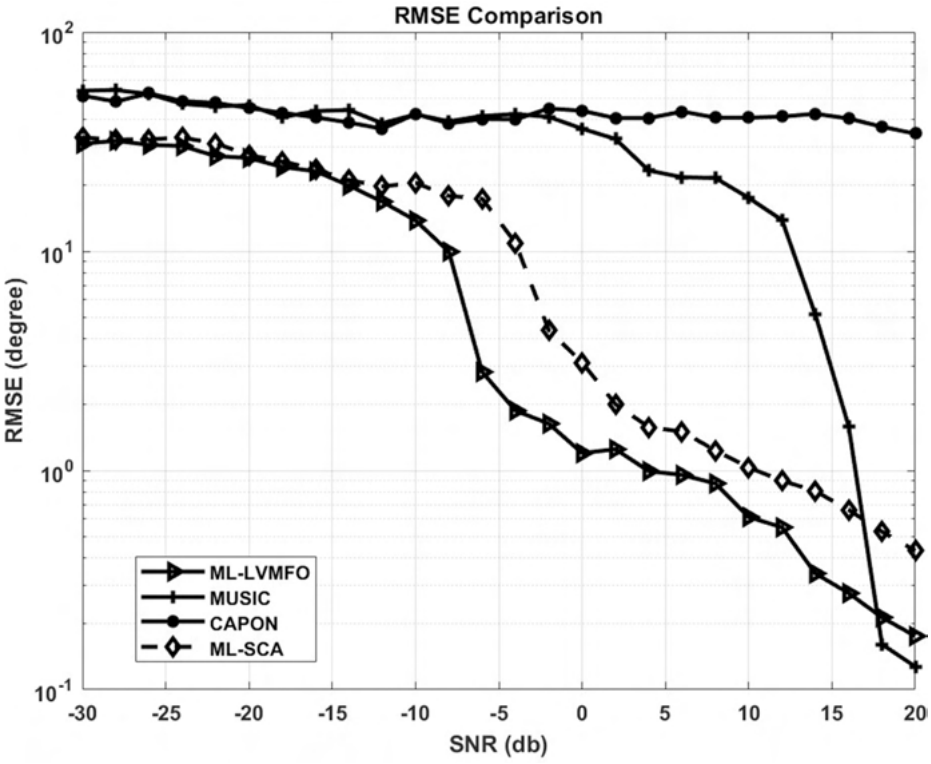


Figure 4: RMSE plot of DOA estimation with respect to SNR for MUSIC, CAPON, ML-LVMFO, and ML-SCA.

6.2 Probability of resolution

Probability of resolution (PR) is defined as the ability of the algorithm to resolve the sources that are closely spaced. In the multisource estimation criteria if the difference between the estimated and the true angle of signals is minimum than the half of the difference between the two signals, then it is assumed that the signals are determined; therefore, it is a critical parameter to judge the resolving capability of the algorithm.

Figure 5 demonstrates the variation of PR with respect to SNR for CAPON, MUSIC, ML-LVMFO, and ML-SCA. The result shows that ML-LVMFO algorithm presents best results and at 14 dB completely resolve the two signals while MUSIC and ML-SCA completely resolve the two signals at 18 and 20 dB. CAPON did not resolve the two signals till 20 dB. With these results, we can conclude that the ML-LVMFO algorithm can be used in situations where the two signals have narrow angular separation.

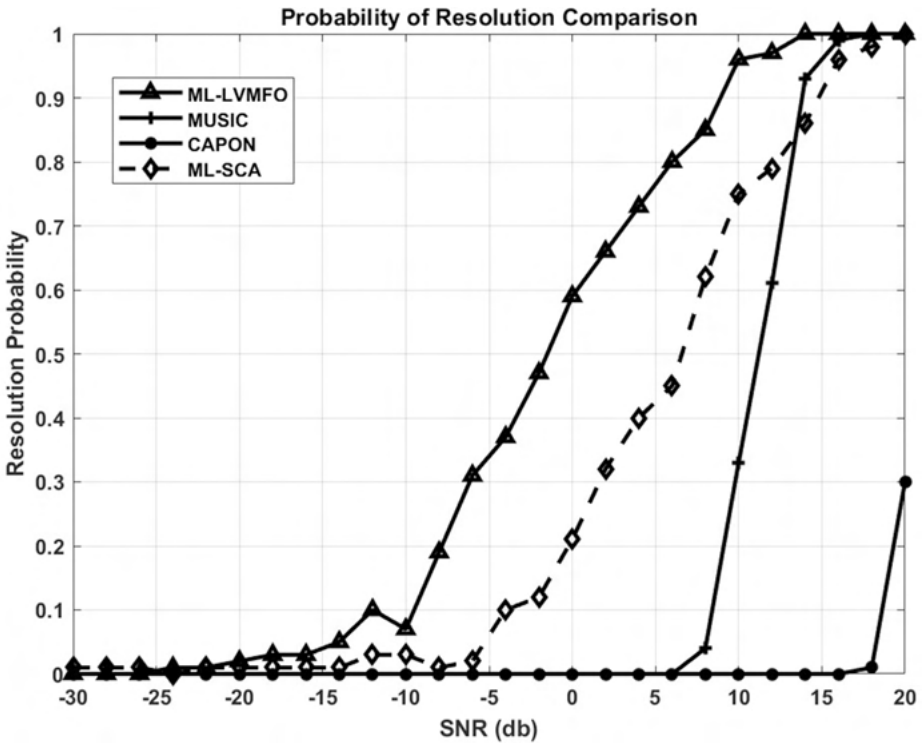


Figure 5: Probability of resolution plot with respect to SNR for MUSIC, CAPON, ML-LVMFO and ML-SCA.

6.3 Rate of convergence and boxplot

In the field of optimization, the rate of convergence is an essential metric for predicting an algorithm's performance. The RMSE convergence curve of the ML-SCA and ML-LVMFO algorithms is shown in Figure 6. The result shows that the LVMFO algorithm outperforms SCA and converges around 150 iterations while SCA converges around 740 iterations.

In DOA estimation, the rate of convergence is critical because once the direction of signals is estimated, thereafter the major lobe is adaptively oriented in the direction of desired user through adaptive algorithms. For 100 Monte Carlo runs, Figure 7 shows a boxplot for ML-LVMFO as well as ML-SCA. It can be predicted from the results that ML-LVMFO technique precisely discovers the direction of narrow spaced signals in comparison to ML-SCA.

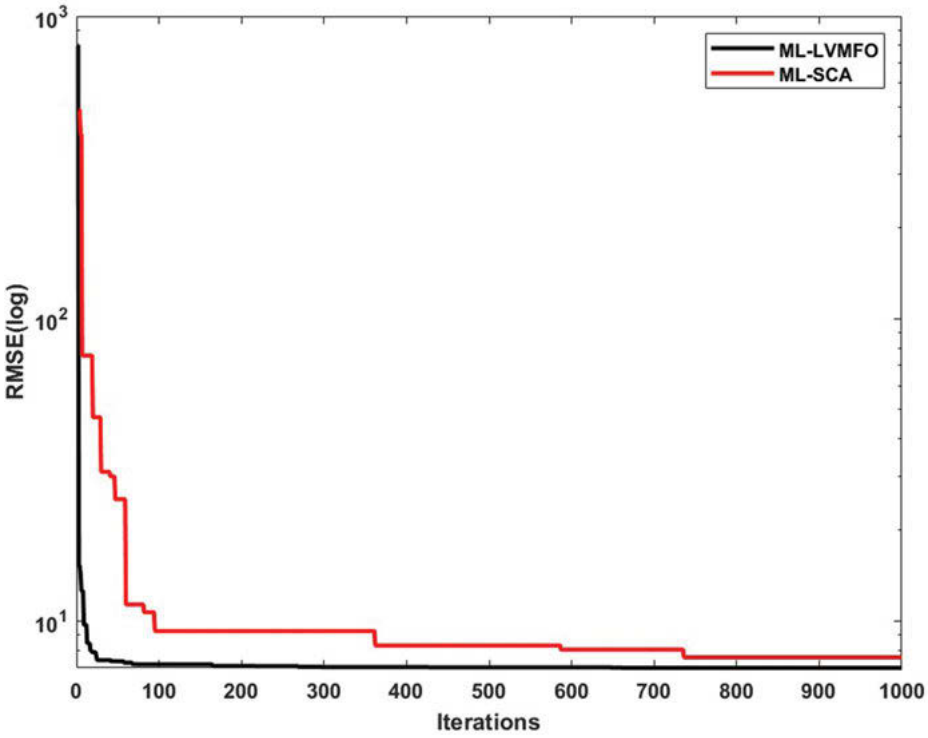


Figure 6: RMSE convergence plot of LVMFO and SCA.

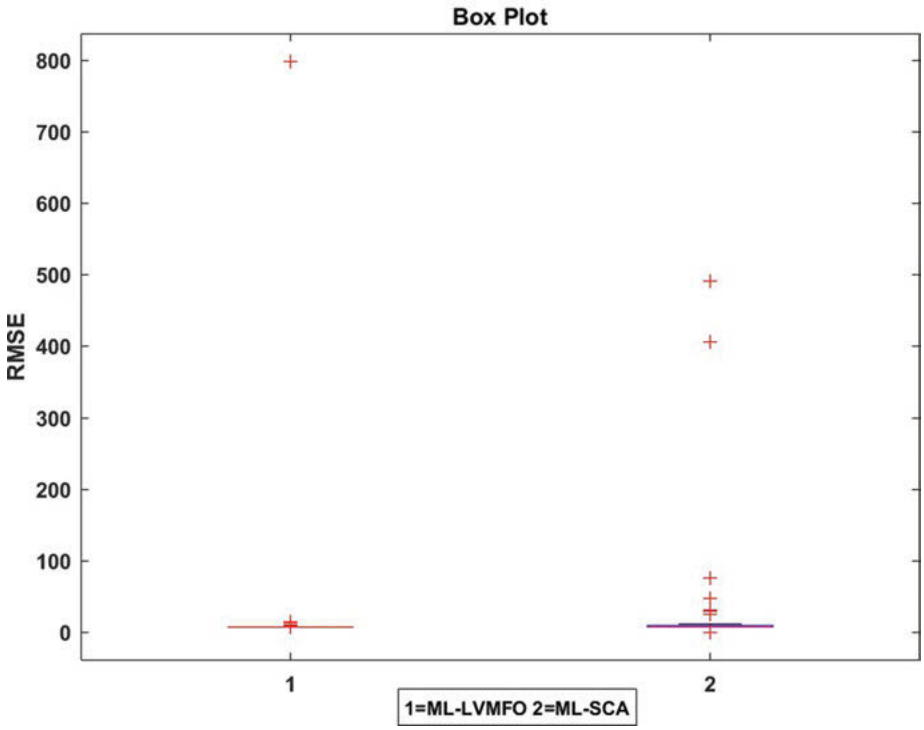


Figure 7: Boxplot of ML-LVMFO and ML-SCA.

7 Conclusion

In this chapter, a novel improved version of MFO algorithm based on Lévy flight mechanism is proposed for estimating the direction of arriving signals to linear array by optimizing the deterministic ML function. The recital of the procedure is analyzed with respect to RMSE and PR and is compared with ML-SCA, MUSIC, and CAPON algorithms. The result shows that ML-LVMFO outperforms other approaches in low SNR environment, produces best results, and accurately estimates the direction of signals having narrow angular separation. Furthermore, the performance of the proposed algorithm is analyzed in terms of rate of convergence and distribution of RMSE values. As a result, ML-LVMFO converges considerably faster over ML-SCA and produces consistent results.

The suggested metaheuristic technique may be investigated for DOA estimation for other antenna arrays as well as in various channel environments in future study. The proposed LVMFO algorithm can also be utilized for solving other NP-hard optimization problems.

References

[1] X. S. Yang, *Nature-inspired Metaheuristic Algorithms*, Luniver press, United Kingdom, 2010.

[2] D. E. Goldberg and J. H. Holland, 1988. Genetic algorithms and machine learning.

[3] J. Kennedy and R. Eberhart, 1995, November). Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks, 4, 1942–1948). IEEE.

[4] M. Dorigo, M. Birattari, and T. Stutzle, 2006. Ant colony optimization. IEEE computational intelligence magazine, 1(4),28–39.

[5] S. Mirjalili, The ant lion optimizer, *Advances in Engineering Software*, 83, 80–98, 2015.

[6] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, Springer Science & Business Media, Berlin, Heidelberg, 2006.

[7] X. Yao, Y. Liu, and G. Lin, Evolutionary programming made faster, *IEEE Transactions on Evolutionary Computation*, 3(2), 82–102, 1999.

[8] X. S. Yang and S. Deb, Cuckoo search via Lévy flights, In: *2009 World Congress on Nature & Biologically Inspired Computing (Nabic)*, 2009 December, 210–214, IEEE, Coimbatore, India.

[9] X. S. Yang and X. He, Firefly algorithm: Recent advances and applications, *International Journal of Swarm Intelligence*, 1(1), 36–50, 2013.

[10] S. Mirjalili, Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, *Neural Computing & Applications*, 27(4), 1053–1073, 2016.

[11] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, *Knowledge-based Systems*, 89, 228–249, 2015.

[12] P. Barthelemy, J. Bertolotti, and D. S. Wiersma, A Lévy flight for light, *Nature*, 453(7194), 495–498, 2008.

[13] A. Sharma and S. Mathur, Deterministic maximum likelihood direction of arrival estimation using GSA, In: *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, 2016March, 415–419, IEEE, Chennai, India.

[14] A. Sharma and S. Mathur, Comparative analysis of ML-PSO DOA estimation with conventional techniques in varied multipath channel environment, *Wireless Personal Communications*, 100(3), 803–817, 2018.

[15] A. Sharma and S. Mathur 2018. A novel adaptive beamforming with reduced side lobe level using GSA. COMPEL-The international journal for computation and mathematics in electrical and electronic engineering.

[16] A. Sharma, S. Mathur, and R. Gowri, Adaptive Beamforming for Linear Antenna Arrays Using Gravitational Search Algorithm, In: *Intelligent Communication, Control and Devices*, Ed. Rajesh Singh, Sushabhan Choudhury, Anita Gehlot, Springer, Singapore, 1159–1169, 2018.

[17] F. Gross, *Smart Antennas for Wireless Communications*, McGraw-Hill Professional, United States of America, 2005.

[18] L. C. Godara, *Smart Antennas*, CRC press, United States of America, 2004.

[19] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, Massive MIMO for next generation wireless systems, *IEEE Communications Magazine*, 52(2), 186–195, 2014.

[20] P. Stoica and K. C. Sharman, Maximum likelihood methods for direction-of-arrival estimation, IEEE Transactions on Acoustics, *Speech Signal Processing*, 38, 1132–1143, 1990.

Preeti Malik, Varsha Mittal, Lata Nautiyal, Mangey Ram

NLP techniques, tools, and algorithms for data science

Abstract: Natural language processing (NLP) is the subdomain of artificial intelligence that explores different methods of using computers to understand, analyze, and process the huge amount of human language or text. Technically, NLP focuses on knowledge acquisition and manipulation of the natural language in order to develop tools and techniques to help the computer understand and accomplish the desired task. The process of NLP is done in two phases: data preprocessing and development of an algorithm. This chapter summarizes these techniques. The tools used for NLP are also included in the discussion. NLP is used in text classification, extraction, summarization, and many other areas, which are also discussed in the chapter.

Keywords: NLP, tools and techniques, data science, sentiment analysis, syntactic analysis, NLTK, Apache OpenNLP

1 Introduction

Everything we say (verbally or in writing) has a great deal of information. Everything we do (from the topic we choose, to the tone we use, to the language we use) adds some type of information that can be analyzed and value taken. In principle, we can use such data to understand and even anticipate human behavior [1]. The data generated from discussions, pronouncements, and even tweets are the examples of unstructured data. Unstructured data, which makes up the great majority of data in the real world, do not fit cleanly into the usual row-and-column structure of relational databases. It is clumsy and difficult to work with. Nonetheless, because of advancements in areas such as machine learning (ML), a major revolution in this field is underway [2]. Nowadays, comprehending the meaning behind such words is more important than trying to interpret a text or speech based on its keywords (the old-fashioned mechanical technique) (the cognitive way). This allows for the detection of figures of speech such as irony, as well as sentiment analysis [3].

Preeti Malik, Graphic Era (Deemed to be) University, Dehradun, India, e-mail: preetishivach2009@gmail.com
Varsha Mittal, Graphic Era (Deemed to be) University, Dehradun, India, e-mail: var.aadi@gmail.com
Lata Nautiyal, University of Bristol, Bristol, UK, e-mail: lata.nautiyal1903@gmail.com
Mangey Ram, Graphic Era (Deemed to be) University, Dehradun, India, e-mail:drmrswami@yahoo.com

https://doi.org/10.1515/9783110734652-006

Natural language processing (NLP) is the subdomain of artificial language (artificial intelligence, AI) that explores the different methods of using computers to understand, analyze, and process the huge amount of human language or text. Technically, NLP focuses on knowledge acquisition and manipulation of the natural language in order to develop tools and techniques to help the computer understand and accomplish the desired task [4]. Different disciplines like computer and information technology, AI, mathematics, psychology, and linguistic provide a strong foundation to NLP.

Computational linguistics – rule-based human language modeling – is combined with statistical, ML, and deep learning models in NLP. These technologies, when used together, allow computers to process human language in the form of text or speech data and "understand" its full meaning, including the speaker or writer's intent and mood [5]. Different application areas of NLP include machine translation (MT), speech recognition, text summarization, multilingual information retrieval, development of expert systems in the medical field, and business intelligence.

Starting with the fundamentals of NLP, this chapter seeks to summarize the significant research activity in this field. This chapter discusses the tools and techniques established for developing NLP systems, as well as the specific areas of application for which they are constructed. Despite the fact that MT is a vital tool, this topic is not just a part of NLP study, but it is also its origin because it is such a large area, it requires its own treatment.

1.1 How it works?

At the core of any NLP problem, the prominent issue that exists is the understanding of the natural language in the same manner as humans do. AI is used by NLP to take input and to process it in a way so that it can be understood by the computers. The process of NLP is done in two phases: data preprocessing and development of an algorithm (see Figure 1).

1.1.1 Data preprocessing

Data preprocessing is an integral and important part of building an ML model for NLP tasks. In this phase, the basic units of text like characters, words, and sentences are extracted and are used in all other different phases of processing. The data are converted into the form that can be processed efficiently by different algorithms. The various methods used for data preprocessing are described further.

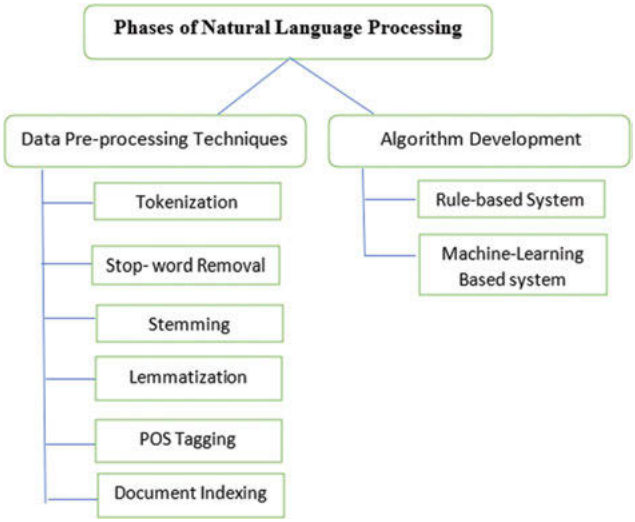


Figure 1: Phases of NLP.

Tokenization

It is a method of splitting the text into meaningful elements also known as tokens. Tokens include all words, symbols, and phrases and are later used as an input in the processing stage. It is beneficial both in linguistics as a form of text segmentation as well as in lexical analysis. Initially, all the textual data are simply the block of characters. Different information retrieval processes require the words; therefore, there is a requirement for tokenization of the textual data. It may be trivial since the data are already stored in machine-readable format; however, the problems like removal of punctuation marks and processing of special characters like brackets are still need to be taken care of. Maintaining the consistency of documents while parsing is important for tokenizer. Handling abbreviations and transforming acronyms into the standard form is another problem of tokenizer [6].

Tokenization challenges differ depending on the type of language. Space delimited languages, such as English and French, are so named because most of the words are separated by white spaces. Unsegmented languages, such as Chinese and Thai, are defined by the lack of unambiguous word boundaries. Tokenizing unsegmented language phrases necessitates the inclusion of additional lexical and morphological data. The writing system and the typographical structure of the words have an impact on tokenization.

Stop word removal

Many words appear frequently in text, although they are virtually meaningless because they are employed to connect words in a sentence. Stop words (widely

accepted) do not contribute to the context or content of literary works. Their existence in text mining creates a barrier to interpreting the content of the documents due to their high frequency of recurrence [7].

Stop words, such as "and," "are," and "this," are frequently used words. They are useless for document classification. As a result, they must be deleted. However, compiling a list of stop words is difficult and varies among literary sources. This technique also minimizes the amount of text data in the system and enhances its speed. Every text document deals with these nonessential terms which are not required in any text mining applications.

Challenges that exist in stop words filtering process includes the difficulty in building a list of stop words due to inconsistency between different textual sources, and their high frequency of occurrences also poses difficulty in processing the textual data. Four methods that are used for stop word filtering are specified below [7]:

(i) **Classic method:** The traditional method involves deleting stop words from precompiled lists.

(ii) **Method using Zipf's law (Z-technique):** In addition to the conventional stop list, we employ three stop word creation methods, including omitting the most common words (TF-High) and deleting words that only appear once, that is, singleton words (TF1). We also think about getting rid of words that have a low inverse document frequency [8].

(iii) **Mutual information (MI) method:** The MI is a supervised method that calculates the MI between a certain term (i.e., positive and negative) and a certain class of documents and gives an evaluation of how much data the term can say on that kind of information**.** Because the term has low MI, it should be removed, as the discrimination power is low.

(iv) **Term-based random sampling:** Stop words in documents have long been manually identified using this method, as proposed by Lee et al. (2005) [9]. This method selects random sections of data, one chunk at a time. As shown in eq. (1), Kullback–Leibler divergence is used to order terms within each chunk based on their in-document values:

$$d_x(t) = P_x(t) \cdot \log_2 \frac{PX(t)}{P(t)} \tag{1}$$

where $P(t)$ is the normalized term frequency of t in the entire collection, and $P_x(t)$ is the normalized term frequency of a term t within a mass x. The final stop list is generated by selecting the words that are the least descriptive across all chunks and eliminating any potential duplications.

Stemming

The process of stemming is the discovery of a root word that is represented in various forms. For example, the words "directed," "directions," and "directing" could

all be stemmed by the root word "direct." The aim of this approach is to eliminate numerous suffixes, minimize the number of words, ensure that stems are correctly matched, and save time and memory space [10].

Challenge of stemming exists in mechanism used for controlling error. Stemming errors include over-stemming (false positive) and under-stemming (false negative). Over-stemming occurs when two different stemmed words are stemmed to the same root. Under-stemming occurs when two words that should have been stemmed to the same root are instead stemmed to distinct terms. The Porter stemmer, for example, turns "universal," "university," and "universe" into "univers." This is an over-stemming case: although those three words are linguistically related, their modern significance is very much different, so it is likely to decrease the importance of the search results by treating them as synonyms within a search engine. Similarly, for the example of under-stemming, take the words "data" and "datum." Some algorithms can reduce these words to the letters "dat" and "datu," which is obviously incorrect. All of these must be boiled down to a specific stem "datum." The majority of stemming studies till date have been conducted in English and other west European languages [11].

Lemmatization

The goal of stemming and lemmatization is to reduce a word's inflectional forms and related forms to a single stem. Stemming is a fundamental heuristic that truncates the ends of sentences. Derivational affixes are removed as part of this process [12].

Lemmatization is the process of removing inflectional endings from a word and returning it to its base form, which is called a lemma, using a vocabulary and morphological analysis. When faced with the token "saw," stemming would only return "s." Lemmatization, on the other hand, would try to return "see" or "saw" based on whether the token was employed as a verb or a noun. When it comes to derivationally related words, stemming often fails. In the different inflection forms of a lemma, lemmatization fails.

Part-of-speech tagging

Part of speech (POS) enriches the "word" and its "meaning" with a massive amount of data about itself and its surroundings. The use of terms like noun, pronoun, verb, adverb, adjective, article, preposition, and conjunction in a sentence aids in inferring potential information about neighboring words and syntactic structure weaving around the term. As a result, POS labeling becomes an inextricable part of syntactic parsing.

POS tagging is the process of identifying each word in an input text and applying a POS marker to it. A tagging algorithm takes a list of words and a set of tags as input and outputs a list of tags and a single best tag for each word [13].

Tagging is a disambiguation task; since words with multiple POSs are used in sentences, the aim is to tag the word with the correct POS. The word book, for example, can be a verb (book the ticket of movies) or a noun (this book has beautiful pictures), and "that" can be a determiner (Does that flight serve dinner) or a complementizer resolution. The delinquent with POS tagging is resolving these ambiguities by selecting the appropriate tag for the situation [14].

Document indexing

"Indexing a text is done by using a list of words that will be included in the document. It works by taking a suitable collection of keywords from a huge corpus of papers and assigning weights to them for each document, essentially turning each document into a vector of keyword weights. The weight is determined by the frequency in which the word appears in the document and the number of documents in which it appears." Index words are chosen using a variety of methods. The identification of noun groups is one such process. "It makes sense to combine two or three nouns in one component in close proximity to the text into a single indexing component (e.g., Information Technology, European Union, United Arab Emirates). A noun category is a set of nouns in a text whose syntactic distance does not exceed a predetermined limit" [15].

1.1.2 Algorithm development

Once the text is preprocessed, an algorithm is designed to process the data. There are a variety of NLP algorithms, but there are two key types that are widely used and are specified below:

Rule-based system

The rule-based systems are the most traditional approach of NLP. This method employs linguistic rules that have been carefully crafted. The rule-based approach traditionally involves a human being in developing and improving the system gradually. The most significant benefit of formal grammar is that it can often be determined whether or not a device can process a user's question and how it can do so. And, since all of the rules were written by humans, any identified bug is simple to locate and resolve by modifying the rules in the relevant module [16].

Grammar rules can be created in a variety of ways, such as by extending the base of translation rules and synonyms, and they can be easily modified with new functions and data types without requiring major changes to the core system. This technique to query analysis, unlike ML-based alternatives, is based on the construction and expansion of existing rules; hence, the algorithm does not require a large

training corpus. The important points about rule-based approaches are summarized below:

1. Tend to concentrate on pattern recognition or parsing
2. Are sometimes referred to as "fill in the blanks" methods
3. Are low precision and high recall, implying that they can perform well in particular use cases but experience performance degradation when applied more broadly

The most obvious drawback of the rule-based approach is that it necessitates the use of trained experts: manually encoding each rule in NLP necessitates the use of a linguist or a knowledge engineer. Rules must be manually designed and improved on a regular basis. Furthermore, the scheme can become so complex that certain rules begin to conflict with one another [12].

Overall, a rule-based method is effective at capturing a particular linguistic phenomenon: it will decipher the linguistic associations between words in order to understand the sentence. As a result, it excels at sentence-level tasks like parsing and extraction. As a result, rule-based methods are best suited to query analysis in general.

Machine learning–based approaches

NLP makes extensive use of ML. Algorithms that learn to "understand" language without being specifically programmed underpin this approach. This is accomplished through the application of statistical methods, in which the algorithm analyzes the training set (annotated corpus) in order to generate its own information, laws, and classifiers [17].

The ML method leaves substantially fewer formal guarantees because it is dependent on probabilistic results [18]. It suffers from the butterfly effect, just like any other complicated mechanism that a person cannot completely observe: even a small quantity of new data that is used for learning will greatly change the model, and the novel "amended" version of the model can behave unpredictably even to its creator.

The obvious benefit of ML is its "learnability," which is why no manual rule/grammar coding, which demands high expertise, is required: A low-skilled labor can annotate the corpus. Since there are several data points (e.g., keywords) in both cases, ML is good at tasks like text classification and word clustering from a corpus. This makes it easy for the machine to learn statistical hints of the terms for a given task.

When good training datasets are available, ML approaches can be used to substantially speed up the development of some NLP systems' performance. In reality, however, it is not always so easy [19].

The lack of training data (which helps the system learn how to translate plain English into SQL) is the biggest challenge with using the ML approach to construct an NLP framework for query analysis; thus, you should have lots of parsed messages (and preferably all the parsed message should belong to the same domain, such as "transportation enquiry system"). But what if you did not already have a similar app where users could input their questions in simple English? What are your plans for obtaining them? One option is to start brainstorming and manually develop them, which can take a long time because datasets must be large enough for the "qualified" ML-based framework to deliver highly accurate results in recognizing the query and providing the appropriate information. Furthermore, once generated and labeled, the corpus is often unable to be reused on new data schemas, necessitating new data "preparation" each time [20].

The difference between rule-based approaches and ML approaches are discussed briefly in Table 1.

Table 1: Difference between rule-based systems and machine leaning–based system.

Rule-based approaches	Machine learning approaches
These models are deterministic.	These models are probabilistic and use statistical laws. Machine learning systems are continuously evolving, developing, and adapting their performance in response to training data streams.
The projects developed using the rule-based approach are not scalable.	These systems are highly scalable.
Rule-based AI models can work with very basic data and knowledge.	The system developed using machine learning approaches require more demographic data for training.
Rule-based systems have high precision.	Machine learning systems have high recall.
To develop the rule-based system there is a requirement of understanding the basic language phenomenon.	Machine learning systems do not require the basic understanding of language phenomenon.
These AI models are immutable objects.	Machine learning models are mutable objects that allow companies, using mutable coding languages such as java, to transform data or value.

2 Techniques of NLP

Syntactic and semantic analyses are two methods for breaking down natural language into machine-readable chunks that are used in many NLP tasks. Classification of NLP techniques is shown in Figure 2.

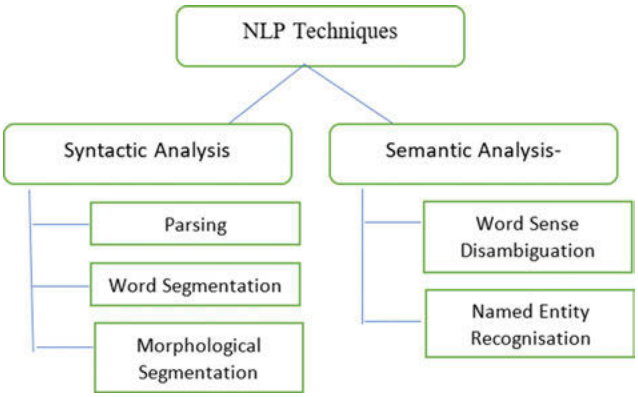


Figure 2: Classification of NLP techniques.

2.1 Syntactic analysis

It is also known as parsing or syntax analysis, and determines a text's syntactic structure and the dependence relationships between words, which are illustrated on a parse tree diagram. NLP assesses the significance of a language based on grammatical principles using syntax. Following techniques come under the category of syntax analysis.

2.1.1 Parsing

Parsing examines each word in a sentence to deduce its structure from its components. It accomplishes this by combining two components: a parser and a grammar. A parser is a procedural component computer program. It follows the same protocol regardless of the language used. Depending on the language being used, the grammar will vary. As a result, a machine can parse any language by altering its grammar. The grammar tree of a sentence is generated by an NLP parser, which develops the grammatical structure of the sentence, for example, a group of words that is a series of words (words) and which word is the subject or objects of a verb. Several types of grammar rules are used during parsing [6]: (a) context-free grammar and (b) lexicalized context-free grammar [7].

Context-free grammar is characterized as a collection of production rules that generate all possible patterns of strings in a given language in formal language and automata theory. Its components include terminal symbols, nonterminal symbols, productions, and a start symbol. Every rule in context-free grammar is one-to-one, one-to-many, or one-to-none. Lexicalized context-free grammar is the result of combining context-free grammar's parsing efficiency with a constrained, elegant type of

lexical sensitivity from lexicalized tree adjoining grammar. It is made up of two types of trees: initial trees and auxiliary trees.

2.1.2 Word segmentation

Word tokenization is the process of breaking down a string of written language into its constituent words. It is also known as word segmentation. Space is a reasonable approximation of a word divider in English and many other languages that use some kind of Latin alphabet [12].

However, if we just break by space to achieve the desired results, we can still run into issues. Some English compound nouns are written in a variety of ways, and they may or may not contain a room. To overcome this issue, the dictionary-based segmentation techniques can also be used.

2.1.3 Morphological segmentation

The smallest significant elements of a word are called morphemes. Boy, dog, girl, toy, and compute, for example, are all morphemes. They have a clear sense and cannot be broken down into smaller pieces. A single word may often contain several morphemes. Consider the word "unstructured," which is made up of three morphemes: "un," "structure," and "ed."

Morphemes are used in a number of different linguistic situations. They aid in the comprehension of word structure and development. Morphology is used in text preprocessing tasks (word stemming and lemmatization) and in generating vector-space representations of words in NLP.

Two models were created: one that used context and one that did not use context. The context-insensitive model was shown to over-account for some morphological structures. Words with the same stem, even though they were antonyms, were grouped together in particular. The context-sensitive model outperformed the others because it took into account not just the relationships between the stems but also other factors such as the prefix "un." The model was also tested on a number of other well-known datasets [21–23], and it outperformed previous embedding models on all of them.

Many NLP tasks necessitate the use of a strong morphological analyzer. As a result, Belinkov et al. [24] conducted a study to see how much morphology was learned and utilized by various neural MT models. Several translation models were created, all of which translated from English to French, German, Czech, Arabic, or Hebrew.

Universal morphology is a relatively new branch of morphology research. This task examines the relationships between different languages' morphologies and how they interact, with the ultimate objective of creating a single morphological

analyzer. However, to the best of the authors' knowledge, only one study has used deep learning in this field [25], and even then, only as a supporting task to universal parsing. Several datasets, including the one from a CoNLL shared task [26], are already available for those interested in applying deep learning to this task.

Aside from universal morphology, the creation of morphological embeddings, which take word structures into consideration, may help with multilanguage processing. They may be utilized in a variety of cognate languages, which is beneficial because some languages have more resources than others. Morphological constructs may also be critical in dealing with specialized languages like the literature of biomedical sciences. Better treatment of morphological components is anticipated to enhance the overall model efficiency, given how deeply deep learning has gotten established in NLP.

2.2 Semantic analysis

The aim of semantic analysis is to figure out what language means. Semantic tasks look at the structure of sentences, word interactions, and related concepts in order to figure out what words mean and what a text is about. It is one of the most difficult fields of NLP because language is polysemic and vague. Some of the NLP techniques that fall under the semantic analysis are discussed further.

2.2.1 Word sense disambiguation (WSD)

Words can have diverse meanings according to the context in which it is used. For example, the word "book" can be used as a noun where it refers to the book to read. It can also be used as a verb like "to book a ticket". It can also be used by accountants to refer the ledger books. Disambiguation is the process of assigning a specific context to each term in the corpus. Word sense disambiguation (WSD) is the task of determining the meaning of a word in context. It has been a longstanding research objective for NLP. WSD is contextual, which means that a collection of words that come next to each other in the same context appear to have similar meanings [4].

Natural language has a built-in complexity that allows it to express an idea or concept in a variety of ways. When a single word is used to refer to multiple concepts, the term becomes confusing. There are many synonyms for each word. The context of a word changes its meaning. Sublanguages provide a context that is restricted to specific domains [5], which reduce ambiguity. For WSD, there are two basic techniques: knowledge-based (or dictionary) approach and supervised approach. The first seeks to infer meaning from ambiguous keywords in a text by

looking up dictionary meanings, whereas the latter uses NLP algorithms to train the model from the training data.

WSD is modeled as a classification problem in supervised techniques, with each classifier dealing with one target word. Each classifier is trained independently using all annotated examples related to a certain target term. Despite the high demand for a large labeled corpus, most techniques in this area outperform knowledge-based alternatives [7]. However, there has been no major performance increase in this category of techniques in recent years.

The primary goal behind knowledge-based (KB) techniques is to fully utilize the knowledge contained in KBs like WordNet [8] and BabelNet [9]. In this category, there are primarily two study streams. One method is to look for overlap, or resemblance, between the context of a word that has to be disambiguated and relevant information from a KB, such as the definition of a prospective sense and its associated sense. The predicted sense is then determined by the most similar sense.

Knowledge-based techniques have grown rapidly in recent years due to their lack of reliance on an expensive sense-annotated corpus. As a result, the performance difference between the two approaches has reduced. Even in the most recent WSD dataset, certain knowledge-based approaches outperform supervised alternatives.

2.2.2 Named entity recognition (NER)

The task of identifying named entities in text, such as a person, location, organization, drug, time, clinical treatment, biological protein, is known as named entity recognition (NER). Question answering, information retrieval, coreference resolution, topic modeling, and other tasks frequently use NER systems as the first step. Handcrafted rules, lexicons, orthographic features, and ontologies were used in early NER systems.

In terms of NER methodologies, there are three main streams:

(1) Rule-based techniques rely on handcrafted rules rather than the annotated data; domain-specific gazetteers [9] and syntactic-lexical patterns [27] can be used to create rules. For speech input, Kim [28] advocated using the Brill rule inference technique. Based on Brill's POS tagger, this system constructs rules automatically. Hanisch et al. [29] proposed ProMiner in the biomedical area, which uses a preprocessed synonym dictionary to find protein references and possible genes in the biomedical material. For NER in electronic health records, Quimbaya et al. [30] developed a dictionary-based solution. The strategy enhances memory while having a minor impact on precision, according to the findings of experiments. Due to domain-specific rules and inadequate dictionaries, such systems frequently exhibit high precision and low recall, and they are unable to be ported to other domains.

(2) Unsupervised learning approaches rely on unsupervised algorithms rather than hand-labeled training instances. Clustering is a common unsupervised learning strategy. Clustering-based NER systems use context similarity to extract named entities from clustered groups. The key principle is that named entity mentions can be inferred using lexical resources, lexical patterns, and statistics generated on a huge corpus. Collins et al. [31] found that using unlabeled data reduces supervision requirements to just seven simple "seed" rules.

(3) Feature-based supervised learning techniques rely on supervised algorithms rather than hand-labeled training instances, which are based on supervised learning algorithms with close attention in detail. NER is used to a multiclass classification or sequence labeling task using supervised learning. Features are meticulously created to represent each training example given annotated data samples. The model is then trained using ML methods to recognize similar patterns in previously unknown data.

In supervised NER systems, feature engineering is crucial. A feature vector representation is a text abstraction in which a word is represented by one or more Boolean, numeric, or nominal values [32]. Various supervised NER systems have commonly employed word-level features [33], Wikipedia gazetteer and DBpedia gazetteer have employed list-based lookup features [34, 35], and the concept of local syntax and multiple occurrence used document and corpus features [36, 37].

Deep learning–based NER models have risen to prominence in recent years, achieving state-of-the-art outcomes. Deep learning is superior to feature-based techniques in terms of automatically discovering hidden features.

3 NLP tools

3.1 Apache OpenNLP

Jason Baldridge and Gann Bierner founded OpenNLP in 2000 as graduate students at the University of Edinburgh's Division of Informatics [38]. It is library is an ML-based NLP toolset. It can perform language identification, chunking, tokenization, coreference resolution, POS tagging, parsing, named entity extraction, and sentence segmentation, among other NLP tasks. These tasks are frequently required in the development of increasingly advanced text processing systems.

Library structure of Apache OpenNLP

The library includes a number of components that may be used to create a complete NLP pipeline. Some of the components are sentence detector, chunker, tokenizer,

parser, name finder, POS tagger, and coreference resolution. Components are parts that allow you to complete an NLP task, train a model, and so on. The OpenNLP Java API was finalized in March 2003, and the new OpenNLP Toolkit was produced using the API and the Grok text processing system. Since then, the OpenNLP Toolkit and OpenCCG have grown separately, with primarily active developer and user communities. OpenCCG is mostly utilized in academics, but OpenNLP is utilized extensively in both academia and industry. A search on Google Scholar (done in March 2010) yielded over 650 papers referencing OpenNLP as evidence of its academic significance.

3.2 Apertium

Apertium is an open-source rule-based MT platforms, licensed under the GNU General Public License. For all lexical changes, Apertium uses finite state transducers, and hidden Markov models for POS tagging and disambiguation of word categories. Currently available MT systems are mostly commercial, which make them challenging to familiarize to new uses. Moreover, different technologies have been used across language pairs, making it hard to incorporate them in a single multilingual content management system, for example. To make it easier for people to contribute to the development of Apertium, and overall growth, the project employs a language-independent specification.

Apertium has 51 stable language pairings available as of December 2020, giving quick translation with relatively understandable output. As an open-source initiative, this tool gives potential developers the tools they need to create their custom language pair and mitigate the project.

Apertium pipeline

The Apertium's technique for translating a source-language text into a target-language text is depicted in Figure 3.

Step 1: Apertium accepts text in the source language as input.

Step 2: The deformatter eliminates arranging markup that ought to be kept set up yet not deciphered.

Step 3: The morphological analyzer fragments the content (growing elisions, stamping set expressions, etc.), and look into sections in the language word references including labels for all matches.

Step 4: The morphological disambiguator settles equivocal sections by picking one match. It utilizes the Visual Interactive Syntax Learning Constraint Grammar Parser [39].

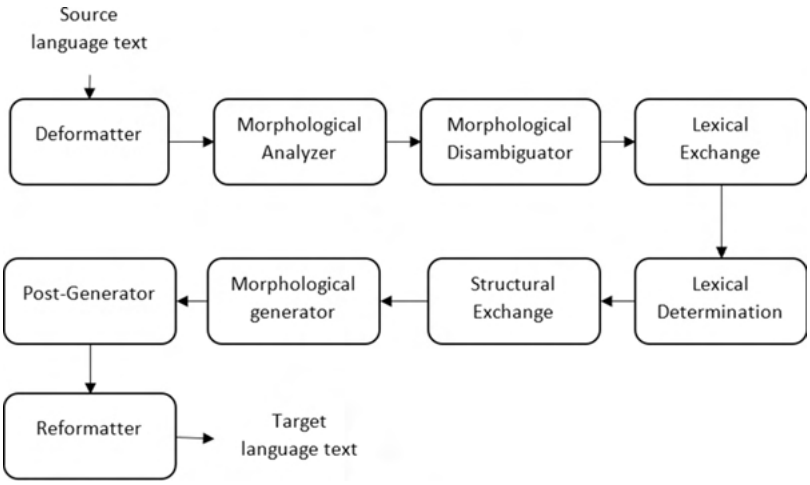


Figure 3: Apertium pipeline.

Step 5: Lexical exchange searches for objective language counterparts of disambiguated source-language basewords. For lexical exchange, it utilizes an XML-based word reference design called bidix [40].

Step 6: Lexical determination picks between elective interpretations when the input text word has elective implications. It does lexical choice using an XML-based technology called apertium-lex-tools. [41].

Step 7: Structural exchange can either include one-venture move or a three-venture move module. It banners syntactic contrasts amid the input and output language by making a succession of pieces that includes the markers for these banners. At that point, it adjusts pieces to deliver a syntactic interpretation in the objective language. It can also be done with the help of lttoolbox.

Step 8: The morphological generator utilizes the labels to convey the right objective language surface structure. It is a morphological transducer, which works in the same way as a morphological analyzer [42].

Step 9: Because of the contact of words, the postgenerator makes any significant orthographic alterations.

Step 10: The reformatter replaces the organizing markup that was removed in the previous phase by the deformatter.

Step 11: Finally, Apertium expresses the objective language interpretation.

3.3 ChatScript

ChatScript is a blend natural language motor and discourse the executives framework planned at first for making chatbots; however, it is at present additionally utilized for different types of natural language preparing. It is written in C++. It is an open source and available at SourceForge [43] and GitHub [44]. It was written by Bruce Wilcox and first performed in 2011, after Suzette had won the 2010 Loebner Prize by deceiving one of the four human judges. Suzette is written in ChatScript.

Features

Overall, ChatScript plans to writer very briefly, since the restricting versatility of hand-composed chatbots is how a lot/quick one can compose the content. Since ChatScript is intended for intuitive discussion, it consequently keeps up the client state across volleys. A volley is quite a few sentences the client contributes without a moment's delay and the chatbots reaction.

The fundamental component of prearranging is the standard. A standard comprises a sort, a name (discretionary), an example, and a yield. Three different types of rules are taken into consideration. When a chatbot has control of the conversation, it may say rules. Responses are decisions that are made in response to a client comment made in response to what the chatbot has just said. Responders are decision-makers who react to aggressive client feedback that is not necessarily related to what the chatbot just stated. Examples depict conditions under which a standard may fire. Examples range from amazingly oversimplified to profoundly perplexing (comparable to Regex yet focused on NL). Weighty use is commonly made of idea sets, which are arrangements of words sharing an importance. ChatScript contains approximately 2,000 predefined ideas, and scripters can undoubtedly compose their own. Yield of a standard intermixes strict words to be shipped off the client alongside basic C-style programming code.

Rules are packaged into assortments called subjects. Points can have watchwords, which permit the motor to consequently scan the theme for pertinent guidelines dependent on the client input.

3.4 General architecture for text engineering (GATE)

General architecture for text engineering (GATE) is a Java setup of apparatuses. It was developed at the University of Sheffield in 1995 and now it is being used worldwide by researchers, organizations, and understudies for some, normal language handling assignments, remembering data extraction for some languages [45].

Features

ANNIE (A Nearly New Information Extraction System) is the framework for extracting data incorporated by GATE. ANNIE contains modules; tokenizer, gazetteer, sentence splitter, grammatical form tagger, named elements transducer and a coreference tagger. GATE can handle many languages: English, Chinese, Hindi, Italian, Arabic, French, Cebuano, Bulgarian, Romania, Danish, and Russian.

ML plugins are incorporated with Weka, RASP, SVM Light, MAXENT just as a LIBSVM coordination, for overseeing ontologies like WordNet, for questioning web indexes like Google or Yahoo, for grammatical form labeling with Brill or TreeTagger, and some more. Numerous outside modules are additionally accessible, for dealing with, for example, tweets [46].

3.5 Gensim

Gensim library is the available open source. It is utilized for unsupervised topic modeling and NLP with today's quantifiable AI. Gensim is executed in Python and Cython. It is a text processing system that uses streaming of data and gradual online algorithms to handle enormous text collections. The feature handling large text makes Gensim tool different from other packages that focus only on in-memory processing. The open-source code of Gensim is available on GitHub [47]. The firm rare-technologies.com commercially supports Gensim. This firm also provides student mentorship programs for Gensim through their Student Incubator program [48].

Features

FastText, word2vec, and doc2vec algorithms, and inert semantic examination (LSA, LSI, SVD), nonnegative framework factorization, idle Dirichlet designation (latent Dirichlet allocation), tf-idf, and arbitrary projections, are all included in Gensim [49].

3.6 LinguaStream

LinguaStream is a nonexclusive stage for NLP, in view of gradual improvement of electronic records. LinguaStream is created at the GREYC (French: Groupe de recherche en informatique, picture, automatique et instrumentation de Caen) software engineering research bunch (Université de Caen) since 2001. LinguaStream is accessible for nothing for personal usage and examination purposes.

LinguaStream permits complex preparing streams to be planned and assessed, gathering investigation segments of different sorts and levels: grammatical feature,

sentence structure, semantics, talk, or factual. Each phase of the preparing stream finds and delivers new data, on which the ensuing advances can depend. Toward the finish of the stream, a few instruments permit dissected records and their explanations to be helpfully envisioned.

Technology

As a stage, LinguaStream gives a broad Java API. For instance, very well it may be incorporated with Java EE workers to foster web applications dependent on preparing streams. It is additionally utilized for instructing and gives explicit modules committed to understudies.

3.7 Natural Language Toolkit (NLTK)

The Natural Language Toolkit (NLTK) is a collection of symbolic and statistical NLP libraries and tools for English. Python is used to design the libraries and programs. It was invented by Steven Bird and Edward Loper at the University of Pennsylvania [50].

NLTK is expected to help research and educating in NLP and AI, cognitive science, and ML [51]. NLTK has been successfully used as a training tool, as a study tool for individuals, and as a stage for prototyping and developing research frameworks. Classification, tokenization, stemming, labeling, parsing, and semantic reasoning are all supported by NLTK.

3.8 spaCy

The tool spaCy is an open-source programming library for cutting edge NLP. It is written in Python and Cython [52, 53]. The library is distributed under the MIT permit, and its primary engineers are Matthew Honnibal and Ines Montani.

In contrast to NLTK, spaCy spotlights on delivering software to production usage [51]. spaCy likewise upholds deep learning work processes that permit connecting statistical models prepared by well-known AI libraries like TensorFlow, PyTorch, or MXNet through its own AI library Thinc. SpaCy emphasizes convolutional neural network models for POS tagging, NER, text categorization, and dependency parsing using Thinc as its backend. It provides nondestructive tokenization. For more than 65 languages, the Alpha tokenization feature of spaCy is used [54]. It supports for custom models in PyTorch, TensorFlow, and different systems.

3.9 SparkNLP

Spark NLP is an open-source library for cutting edge NLP for Python, Java, and Scala languages [55–57]. Spark NLP library is based on Apache Spark and its Spark ML library [58]. Its aim is to give an API to NLP. The library provides pre-trained neural network models, pipelines, and embeddings, as well as assistance with the custom model training [58]. **Spark NLP** is equipped toward creation use in programming frameworks that grow out of more seasoned libraries like spaCy, NLTK, and CoreNLP.

4 Usage of NLP

NLP is the most useful tool of the recent era to help small-scale to large-scale businesses to deeply analyze their important information by dig deep inside and to save the time along with cost. This gives them the great benefits to compete them with their competitors. Following are the important applications of NLP.

4.1 Sentiment analysis

The biggest drawback of NLP is to understand the behavior, characteristics, and features of a human being. But the sentiment analysis is the method to help NLP to undershaft the sentiments of humans just as a real human being. It also determines how positive or negative is the characteristics of a human.

Let us talk about analyzing the sentiment in real time, at that time we can monitor social media like Facebook, LinkedIn and twitter (and deal with them before negative reviews escalate), you can also measure the reaction and positive or negative response of the customer regarding the latest marketing campaigns or product launches and get a rough idea of whether the product will be liked to buy the customer or not.

Sentiment analysis can also be performed on a day-to-day or a frequent basis to understand the likes, dislikes, approaches, and other factors regarding the product we are going to sell them. Customer may be very happy or dissatisfied by the new product. By showing you what needs to be improved, these insights can help you make more informed choices.

4.2 Text classification

Another category of sentiment analysis where text is an important factor to deal with is regarded as a text classification. It involves the automatic comprehension,

processing, and classification of unstructured or the semistructured text. Assume we have a large number of replies to analyze from your recent NPS survey, and we have a large number of open-ended responses. Manually doing it will take a lot of your time and cost you a lot of money. Imagine if you could provide a training just like a human being to an NLP model which will automatically tag that mass of words within fraction of seconds, by applying some predefined techniques, or by any new techniques given by you. Would that be great? Yes of course, it will be very rapid and of course cost cutting.

For NPS survey replies, you can use a topic classifier, which automatically tags your data by themes like customer service, features, simplicity of use, and price. Give it a shot and see how it goes!

4.3 Chatbots and virtual assistance

In this recent ML age, some of the AI-based programs automatically provide solutions to the queries to understand the natural languages and make available a well-suitable response or the prompt reply by using the natural languages called as Chatbots and virtual assistants. There are some predefined questions, and their responses are already stored in some form of rules and regulations. AI-powered chatbots and virtual assistants manage all those rules and apart from their management of responses those have the ability to improve the process to help the stakeholders in a better way.

These intelligent machines are increasingly available on customer support wires because the team can take 80% of all routine queries and help the human agent through more complex problems. Available 24 h/24 and 7d/7, ChatBots and virtual assistants are able to provide the quick responses and mitigate the agent of a query that is repeated and off.

4.4 Text extraction

The innovative technique that is able to extract or dig the most relevant information from the given text, for example, name, address, organization, city, and country is called as text extraction. Sometimes it is also referred to as entity recognition. It is able to dig or extract keywords from the long paragraphs, as well as prebuilt characteristics like bar code details of the product and product model.

This method is very much useful for sorting the pool of data, for example, serial number, email address, name of the organization, and current location without even opening or reading each file of the organization. Data can also be imported by using the text extraction. Important information as per the specific need and

configure a trigger to automatically import that information into the database can also be extracted by this method.

Apart from this, text extraction is a meaningful method to see the deep insight into text content, with the help of NLP; further, it is combined with sentiment analysis. This method can add an extra layer of information, letting you know which terms are frequently used by the customer most often to express negative feelings about the services or product offered.

4.5 Machine translation

The conventional application of NLP is ML, although claims provided by renowned social media like Facebook declared ML is having the capabilities of a superhuman. Though, if for many years, someone is using Google Translate, then it is clear to him that it took a long way to finally came in that form, which is more accurate and precise. And its credit goes to the advances in neural networks and the increasing availability of large amounts of data.

MT is especially playing an important role and helping big business houses because it assists in communication, allowing businesses to reach a wider audience and understand the important documents quickly and economically.

4.6 Text summarization

Automatic summaries are fairly easy to understand. It sums up the text along with extracting the highly important information to fulfill the organizations' needs. The ultimate aim of the text summarization is to make the process simple to review large amounts of data, for instance, the scientific articles, news content, or legal documents.

There are two ways of text summarization.

Extractive text summarization: This method finds out the important keywords from the given text and produces them again as part of the summary. In this approach, only the existing text is used in the summarization, and no new text is a generated process.

Abstractive text summarization: This method is considered to be very powerful when compared to the **extractive text summarization**. It interprets text and produces new summary text. It uses its own interpretation rather than picking the words from the given paragraph.

4.7 Market intelligence

NLP helps the market to know about their present and future potential customers to recommend more and more products and to launch new products based on the current feedback and the day-to-day browsing history of the customers. NLP approaches that analyze browsed themes, moods, searched keywords, and intents in unstructured data can significantly improve your market research, revealing trends and business prospects. You can also analyze data to clearly find out the customer weaknesses and track your competitors (see what works for them and what does not).

4.8 Autocorrect

A method that automatically corrects our spelling and grammatical mistakes by using NLP is called as AutoCorrect. It is playing an important role in writing tasks. Tools like Grammarly, for example, employ NLP to help improve your writing by detecting mistakes in syntax, spelling, and sentence structure.

4.9 Intent classification

Intent classification includes the identification of the purpose or persistence behind a text. In addition to chatbots, intent detection can be beneficial in areas of sales and customer support. You can identify clients who are ready to buy by analyzing customer interactions such as emails, discussions, and social media posts. The more quickly you can identify and categorize these leads, the more likely they are to become customers. Organize the responses to this email into categories such as Interested, Uninterested, and Unsubscribe.

Finally, looking into the customer intent in your support tickets or social media posts can help you identify customers who are on the verge of unsubscribing, allowing you to implement a successful approach to retain them.

4.10 Urgency detection

NLP approaches can also aid in the detection of emergency situations in text. You can train an emergency detection model to distinguish particular words and phrases that signify seriousness or discontent based on your chosen criteria. This can assist you in prioritizing your most critical requests and ensuring that they are not buried beneath a mountain of unanswered votes. These detections assist you in increasing response time and efficiency, which improves customer happiness.

4.11 Speech recognition

NLP is used in speech recognition technologies to translate spoken language into a machine-readable format. Virtual assistants like Siri, Alexa, and Google Assistant rely on these recognition technologies to function. However, there are an increasing number of business applications for voice recognition. Businesses can, for example, use text-to-speech capabilities in their business software to automatically record calls, send emails, and even translate.

Jason Baldridge and Gann Bierner founded OpenNLP in 2000 as graduate students at the University of Edinburgh's Division of Informatics [38]. Its library is an ML-based NLP toolset. It can perform language identification, chunking, tokenization, coreference resolution, POS tagging, parsing, named entity extraction, and sentence segmentation, among other NLP tasks. These tasks are frequently required in the development of increasingly advanced text processing systems.

5 Conclusion

Starting with the fundamentals of NLP, this chapter seeks to summarize the significant research activity in this field. This chapter discusses the tools and techniques established for developing NLP systems, as well as the specific areas of application for which they are constructed. Despite the fact that MT is a vital tool, this topic is not just a part of NLP study, but it is also its origin because it is such a large area, it requires its own treatment. The process of NLP is done in two phases: data preprocessing and development of an algorithm. Data preprocessing includes various phases like tokenization, stop-word removal, and stemming, and algorithms of NLP are developed by two methods: rule-based methods and ML-based methods.

References

[1] S. Jusoh and H. M. Alfawareh, Techniques, applications and challenging issue in text mining, *IJCSI International Journal of Computer Science Issues*, 9(6), No 2, 431–436 November 2012.
[2] N. Ranjan, K. Mundada, K. Phaltane, and S. Ahmad, A survey on techniques in NLP, *International Journal of Computer Applications*, 134(8), 6–9, Jan 2016.
[3] T. Young, D. Hazarika, S. Poria, and E. Cambria, Recent trends in deep learning based natural language processing, *IEEE Computational Intelligence Magazine*, 13(3), 55–75, 2018.
[4] A. Gelbukh, Special issue: Natural Language Processing and its Applications, Institut Politécnico Nacional Centro de Investigaciónen Computación México, Mexico, 2010.
[5] S. Jusoh and H. M. Alfawareh, Natural language interface for online sales, in Proceedings of the International Conference on Intelligent and Advanced System (ICIAS2007). Malaysia: IEEE, November 2007, 224–228.

[6] E. M. Sibarani, M. Nadial, E. Panggabean, and S. Meryana, A Study of parsing process on natural language processing in Bahasa Indonesia, International Conference on Computational Science and Engineering, 309–316 2013.
[7] M. F. Porter, An Algorithm for Suffix Stripping, *Program*, 14(3), 130–137, 1999.
[8] S. Vijayarani and R. Janani, Text mining: Open source tokenization tools–an analysis, *Advanced Computational Intelligence*, 3(1), 37–47, 2016.
[9] J. Lee and J. M. Lee, Approximate dynamic programming-based approaches for input–output data-driven control of nonlinear processes, *Automatica*, 41, 1281–1288, 2005.
[10] C. Ramasubramanian and R. Ramya, Effective pre-processing activities in text mining using improved Porter's stemming algorithm, *International Journal of Advanced Research in Computer and Communication Engineering*, 2(12), 4536–4538, 2013.
[11] M. Massimo and O. Nicola, A novel method for stemmer generation based on hidden Markov models. Proceedings of the twelfth international conference on Information and knowledge management. 2003, 131–138.
[12] P. Joel, L. Nada, and M. Dunja, A rule based approach to word lemmatization Proceedings C of the 7th International Multi-Conference Information Society IS. 2004.
[13] K. Nikita and M. S. Chaudhari, Text preprocessing for text mining using side information, *International Journal of Computer Science and Mobile Applications*, 3(1), 01–05, 2015.
[14] P. C. Gaigole, L. H. Patil, and P. M. Chaudhari, Preprocessing techniques in text categorization, National Conference on Innovative Paradigms in Engineering & Technology (NVIPET-2013), Proceedings published by International Journal of Computer Applications (IJCA), 2013.
[15] S. Vaidya and J. Aher, Natural language processing preprocessing techniques, *International Journal of Computer Engineering and Applications*, Volume XI Special Issue 2017, www.ijcea.com ISSN 2321-3469, Retrieved 2021-06-10.
[16] S. Charanyaa and K. Sangeetha, Term frequency based sequence generation algorithm for graph based data anonymization, *International Journal of Innovative Research in Computer and Communication Engineering, (An ISO 3297: 2007 Certified Organization)*, 2(Issue 2), 3033–3040 February 2014.
[17] R. Collobert and J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in Proceedings of the 25th international conference on machine learning, 160–167, ACM, 2008.
[18] Y. Liu and M. Zhang, Neural network methods for natural language processing by Yoav Goldberg, *Computational Linguistics*, 44(1), 193–195, Mar 2018.
[19] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, Very deep convolutional networks for text classification, in Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, Valencia, Spain, Apr. 2017, 1107–1116.
[20] Y. Kim, Convolutional neural networks for sentence classification, Proc. of 14th International Conference on Empirical Methods of Natural Language Processing, Doha, Qatar, 1746–1751, July 2014.
[21] A. Akbik, D. Blythe, and R. Vollgraf, Contextual string embeddings for sequence labeling, in Proceedings of the 27th International Conference on Computational Linguistics, 1638–1649, 2018.
[22] B. Bohnet, R. McDonald, G. Simoes, D. Andor, E. Pitler, and J. Maynez, Morphosyntactic tagging with a Meta-BiLSTM model over context sensitive token encodings, arXiv preprint arXiv:1805.08237, 2018.
[23] J. Legrand and R. Collobert, Joint RNN-based greedy parsing and word composition, arXiv preprint arXiv:1412.7028, 2014.

[24] J. Legrand and R. Collobert, Deep neural networks for syntactic parsing of morphologically rich languages, in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 573–578, 2016.
[25] E. Kiperwasser and Y. Goldberg, Simple and accurate dependency parsing using bidirectional LSTM feature representations, arXiv preprint arXiv:1603.04351, 2016.
[26] C. Dyer, M. Ballesteros, W. Ling, A. Matthews, and N. A. Smith, Transition-based dependency parsing with stack long short-term memory, arXiv preprint arXiv:1505.08075, 2015.
[27] S. Zhang and N. Elhadad, Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts, *Journal of Biomedical Information*, 46(6), 1088–1098, 2013.
[28] J.-H. Kim and P. C. Woodland, A rule-based named entity recognition system for speech input, in *ICSLP*, 2000.
[29] D. Hanisch, K. Fundel, H.-T. Mevissen, R. Zimmer, and J. Fluck, Prominer: Rule-based protein and gene entity recognition, *BMC Bioinformatics*, 6(1), 14, 2005.
[30] A. P. Quimbaya, A. S. Múnera, R. A. G. Rivera, J. C. D. Rodríguez, O. M. M. Velandia, A. A. G. Peña, and C. Labbé, Named entity recognition over electronic health records through a combined dictionary-based approach, *Procedia Computer Science*, 100, 55–61, 2016.
[31] M. Collins and Y. Singer, Unsupervised models for named entity classification, in *EMNLP*, 1999, 100–110.
[32] S. Sekine and E. Ranchhod, *Named Entities: Recognition, Classification and Use*. John Benjamins Publishing, 2009, vol. 19.
[33] W. Liao and S. Veeramachaneni, A simple semi-supervised algorithm for named entity recognition, in *NAACL-HLT*, 2009, 58–65.
[34] A. Toral and R. Munoz, A proposal to automatically build and maintain gazetteers for named entity recognition by using Wikipedia, in *Workshop on NEW TEXT Wikis and blogs and other dynamic text sources*, 2006.
[35] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum, Robust disambiguation of named entities in text, in *EMNLP*, 2011, 782–792.
[36] Z. Ji, A. Sun, G. Cong, and J. Han, Joint recognition and linking of fine-grained locations from tweets, in *WWW*, 2016, 1271–1281.
[37] V. Krishnan and C. D. Manning, An effective two-stage model for exploiting non-local dependencies in named entity recognition, in *ACL*, 2006, 1121–1128.
[38] URL: https://cwiki.apache.org/confluence/display/incubator/OpenNLPProposal
[39] URL: VISL. *beta.visl.sdu.dk*. Retrieved 2021-06-10.
[40] URL: *Bilingual dictionary – Apertium*. https://wiki.apertium.org. *Retrieved 2021-06-10*.
[41] URL: *Constraint-based lexical selection module – Apertium*. https://wiki.apertium.org. *Retrieved 2021-06-10*.
[42] URL: *Morphological dictionary – Apertium*. https://wiki.apertium.org. *Retrieved 2021-06-10*.
[43] URL: https://sourceforge.net/projects/chatscript/
[44] URL: https://github.com/ChatScript/ChatScript
[45] URL: https://gate.ac.uk/gate/plugins/
[46] URL: https://gate.ac.uk/wiki/twitie.html
[47] URL: https://github.com/rare-technologies/gensim
[48] URL: https://rare-technologies.com/incubator/
[49] R. Řehůřek and P. Sojka (2010). Software framework for topic modelling with large corpora. Proc. LREC Workshop on New Challenges for NLP Frameworks.
[50] URL: http://www.nltk.org/book/ch00.html
[51] S. Bird, E. Klein, E. Loper, and J. Baldridge (2008). Multidisciplinary instruction with the Natural Language Toolkit (PDF). Proceedings of the Third Workshop on Issues in Teaching Computational Linguistics, ACL. Archived from the original (PDF) on 2 September 2011.

[52] Choi et al. (2015). It Depends: Dependency Parser Comparison Using A Web-based Evaluation Tool.
[53] URL: *Google's new artificial intelligence can't understand these sentences. Can you?. Washington Post. Retrieved 2021-06-10*.
[54] URL: https://spacy.io/usage/models/
[55] S. A. Ellafi. *Comparing production-grade NLP libraries: Running Spark-NLP and spaCy pipelines. O'Reilly Media. Retrieved 2021-06-10*.
[56] S. A. Ellafi. *Comparing production-grade NLP libraries: Accuracy, performance, and scalability. O'Reilly Media. Retrieved 2021-06-10*.
[57] K. Ewbank. *Spark Gets NLP Library.* www.i-programmer.info. Retrieved 2021-06-10.
[58] A. Thomas, July 2020, *Natural Language Processing with Spark NLP: Learning to Understand Text at Scale*, First United States of America, O'Reilly Media, ISBN 978-1492047766.

Debabrata Swain, Paawan Sharma, Vinay Vakharia,
Tapash Kumar Tanty

Prediction of coronary artery disease using logistic regression

Abstract: At this time, machine learning has become a useful tool to find a complex pattern for classification problems from a collection of real-time datasets. Currently, healthcare field is suffering with the issue of timely and accurate diagnosis of different fatal diseases like cancer and cardiovascular disease. Hence, for saving the life of patient, machine learning can be applied for the timely diagnosis of the syndrome. Coronary artery ailment is a form of cardiac sickness due to which around 7 million people are dying every year. Here, a coronary artery disease prediction system is developed using logistic regression algorithm. For the experimentation purpose, heart disease datasets present in UCI repository are used. The proposed model has shown an accuracy of 94%.

Keywords: coronary artery disease, UCI repository, logistic regression

1 Introduction

Innovations in the modern technology made human life luxurious and lazy. For most of the jobs, people now depend on the gadgets. In addition to this, consumption of junk food, addiction with liquor, tobacco, and lack of sleep tend a person to have big bellies [1]. These unhealthy lifestyle and deficiency of physical exercise make human life more prone toward different chronic diseases like heart disease. One of the common forms of cardiac disease is coronary artery disease [2]. In this ailment, the width of the coronary artery gets blocked due to the presence of cholesterol and other substances. This blockage creates complication against the free flow of blood that mostly results in heart failure of a person [3]. Heart disease has mainly three variants like coronary artery disease, valvular cardiac, and cardiomyopathy. In coronary artery disease, the veins get affected. In valvular disease, the blood flow within the valves

Debabrata Swain, Department of Computer Science and Engineering, School of Technology, PDEU, Gandhinagar, India, e-mail: debabrata.swain7@yahoo.com
Paawan Sharma, Department of Information and Communication Technology, School of Technology, PDEU, Gandhinagar, India, e-mail: paawan.sharma@sot.pdpu.ac.in
Vinay Vakharia, Department of Mechanical Engineering, School of Technology, PDEU, Gandhinagar, India, e-mail: vinay.vakharia@sot.pdpu.ac.in
Tapash Kumar Tanty, Department of Information Technology, National Institute of Science and Technology, Berhampur, India, e-mail: tapas181920tanty@gmail.com

https://doi.org/10.1515/9783110734652-007

gets affected. In cardiomyopathy, the heart muscles functioning get affected [4]. Because of this infection, around 10% of world's mortality results each year [5]. For the finding of the ailment, physicians generally propose various tests like ECG, X-ray, blood test, and CT scan. All these methods apply different high power rays on the human body [6]. Repeated projection of these rays is bad for human health. In addition to this, for the identification of this disease, doctors generally refer to a number of test reports and use their past experience for prediction. This prediction is not always accurate because the disease is related with a number of bioindicators that are sometimes similar with other diseases. Due to this ambiguity in clinical decision making, sometimes it results in death of the patient. This variance in the clinical prediction can be solved to a large extent by applying computational intelligence using the large collection of clinical data. Intelligence within a computer can be developed by conducting a learning process by utilizing a collection of huge amount of healthcare data. This process of developing intelligence within a computer is otherwise known as machine learning (ML). ML is a branch of artificial intelligence that deals with creating intelligence within a machine to take decision as like a human being. Currently, ML is applied in different fields like crop prediction, climate prediction, market value estimation, and healthcare detection. With the motivation to create an efficient system for heart disease prediction, here an ML-based system is proposed using logistic regression (LR). For the model training and evaluation, the UCI heart disease dataset [7] is used.

2 Literature

Alberto et al. [8] proposed an ensemble-based kNN algorithm using different distance formulas. kNN algorithm uses a distance formula to find nearest N neighboring points of test point. Then it assigns a class to the test point based on majority of the neighboring points. The different distance formulas used here are Euclid, Manhattan, Chebyshev, Sorensen, Canberra, and Mahalanobis. Each distance algorithm finds a class for the test point, and at last voting algorithm is applied to select the label with majority.

Hamidreza et al. [9] implemented an ensemble classifier using decision tree, neural network, support vector machine (SVM), and naïve Bayes. Each classifier predicted the class for every test instance individually. The test point will be assigned with the class having majority.

Kasbe et al. [10] used fuzzy logic to predict the occurrence of cardiovascular disease. The entire system was divided into three subsystems namely fuzzification, rule base, and defuzzification. Initially, the fuzzy membership functions like triangular and trapezoidal functions are defined to determine the membership for each input and output. Then the fuzzy expert system is formed by considering the blend of the input and output. Finally, in fuzzy data rule base was used to form 86 rules with AND and OR operators.

Purushottam et al. [11] applied a data mining-based tool KEEL for classification of heart disease. Initially, a set of MV algorithms are used to impute for the missing values. After that the rule sets are created using a decision tree. Finally, the classification rules are generated through a phase of rules like original rule, pruned rule, rules without duplicate, classification rules, and polish rules.

Yeshvendra Singh et al. [12] proposed a heart illness classification model using random forest algorithm. The model is formed using a set of decision trees. The model has outperformed using parameter tuning.

Saba Bashir et al. [13] developed an ensemble-based model for the identification of cardiovascular disease. The different classifiers used in the framework are decision tree, naïve Bayes, and SVM. After obtaining the test result from each individual classifier, the voting algorithm is implemented to find the class with majority.

3 Dataset

In this work, the UCI heart disease dataset [7] is used for model training and evaluation. The dataset consists of four different clinical databases like Cleveland, Hungary, Switzerland, and Long beach containing 303, 294, 123, and 200 number of patent records.

Table 1: Feature table.

Sr No	Feature	Value Type
1	Age	Numeric
2	Sex	Categorical
3	Chest pain	Categorical
4	Resting BP	Numeric
5	Cholesterol	Numeric
6	Fasting Sugar	Categorical
7	Resting ECG	Categorical
8	Maximum rate	Numeric
9	Induced Angina	Categorical
10	ST Depression	Real
11	Slope	Categorical
12	Vessels colored	Categorical
13	Thalassemia	Categorical
14	Heart Disease	Categorical

The total number of features present in these databases is 14. Out of 14 features, 13 are dependent and 1 is target feature. The independent features are of categorical and continuous type. Categorical features are having discrete class values, whereas continuous are having series of values without any limited range. Table 1 shows the feature table.

4 Proposed system

The system is divided into two subsystems: data preprocessing and classification as shown in Figure 1. The data preprocessing subsystem is responsible for cleaning the data and classification subsystem is used for classifying the records into different target classes.

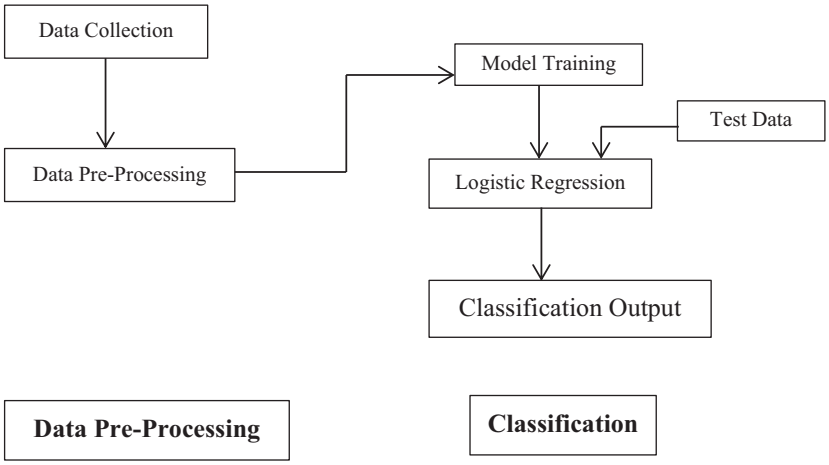


Figure 1: Proposed system architecture.

4.1 Data preprocessing

In this phase, the data cleaning and imputation operations are performed. The total count of the records present in the dataset is 920. Several features contain missing values in the dataset. The count of missing values in the slope of the peak exercise is 308, a major vessel colored by fluoroscopy is 606, and thalassemia is 477. The missing values are visualized using a heat map. Missing values in cholesterol are imputed with average. But for fasting sugar, vessels colored, and thalassemia mode is used. The number of records found in the final cleaned dataset is 854. The target column contains different values like 1, 2, 3, 4 for presence and 0 for absence of diseases. As the proposed system developed for binary prediction, hence, 1, 2, 3, 4

are replaced with 1 to indicate the presence of diseases. The data balancing is checked at the end for unbiased model development. The count of absence records is 389 (45%) and the presence of records are 465 (55%). The dataset before and after cleaning are shown in Figures 2 and 3.

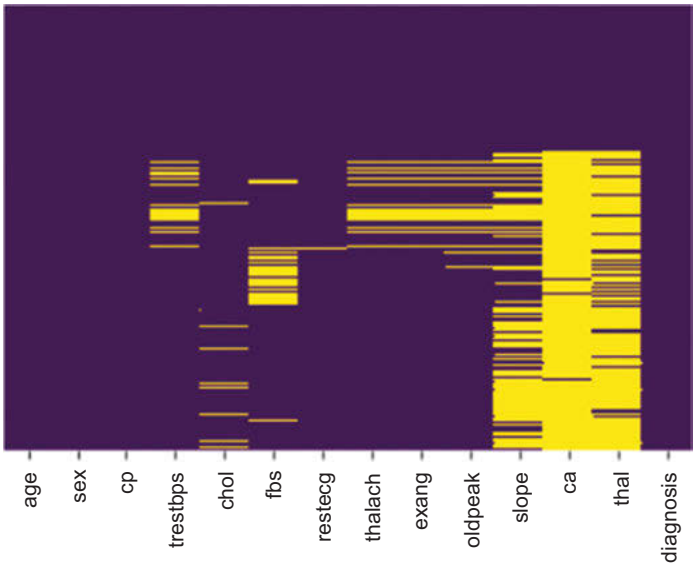


Figure 2: Dataset before cleaning.

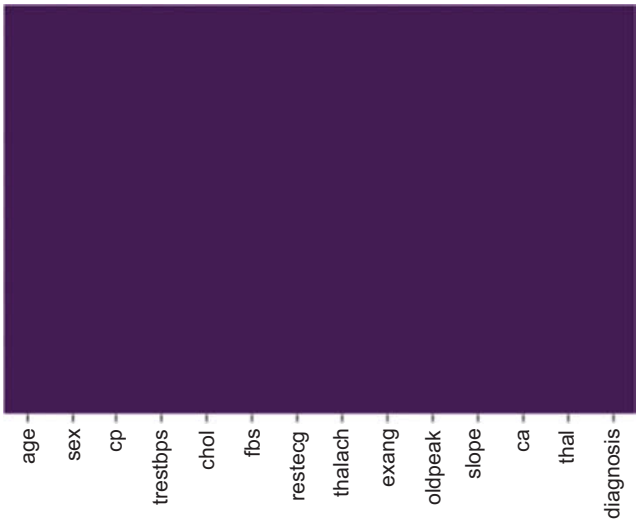


Figure 3: Dataset after cleaning.

4.2 Model details

LR is a supervised classification algorithm. It classifies a test point into two distinct classes [14]. Here the proposed system is having two classes for the target feature, that is, sick or healthy class. The algorithm creates threshold for making any classification decision. When any point lies above the threshold, it is classified into one class; otherwise, it is classified into another class [15]. LR works as per the following equation:

$$y = \sigma(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

where y is the target value, x is the input feature value, $\sigma(x)$ is the sigmoid function.

LR works on the basis of sigmoid function. The sigmoid function generally maps an input value having range ($-\infty$ to $+\infty$) in between 0 and 1. If the value of output feature is below the threshold value 0.5, then the point is classified as class 0, otherwise, as class 1. The characteristic graph is shown in Figure 4.

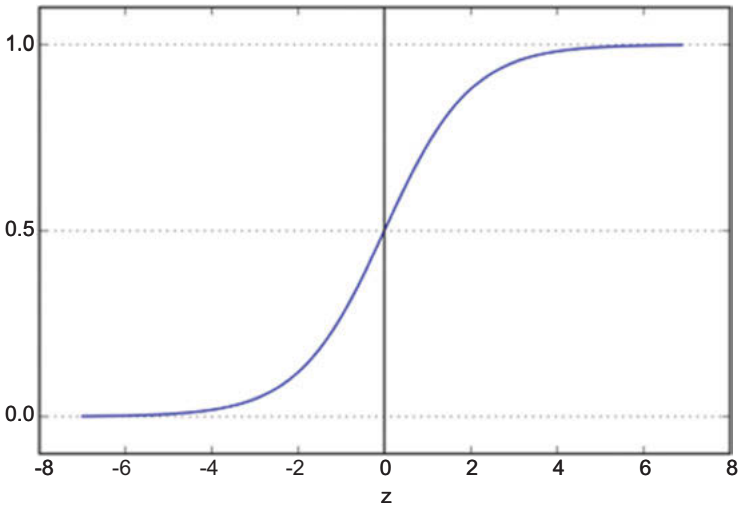


Figure 4: Sigmoid function.

5 Test result and performance analysis

The efficiency of the model is measured using the unseen test data. The entire data split into training and evaluation set. Here the model is validated using 20% of the data. The training dataset count is 683 and test data count is 171. The model is evaluated using different performance metrics like accuracy, precision, recall, F1 score,

and ROC curve. Accuracy is the proportion of correctly predicted data with total data [16]. Precision is the proportion of correctly positive predicted samples with the total positive predicted samples [17]. Recall measures the ability of the model to recognize the true positive cases correctly. F1 score measures the performance of a model in terms of precision and recall. The equation for the discussed performance parameters are shown as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4}$$

$$\text{F1 score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

The proposed model has obtained an accuracy of 94%, precision of 94%, recall of 94%, and F1 score of 94%. The classification report and confusion matrix are shown in Tables 2 and 3.

Table 2: Different Split Accuracy.

Split	Training Accuracy	Testing Accuracy
70–30	93.8	93.77
80–20	93.99	93.56
90–10	93.22	94.18

Table 3: Confusion matrix.

Parameter	Count
TP	67
FP	2
TN	93
FN	9

The model is also evaluated using some other split percentage of train and test like (70–30%) and (90–10%). It is observed in all split percentage that the accuracy during training testing is not fluctuating and remains constant. That shows the generic behavior of the model. This also shows that the model is efficiently

handling the overfitting and underfitting issues. The observations are shown in Table 4 and Figure 5.

Table 4: Different split accuracy.

Split	Training accuracy	Testing accuracy
70–30	93.8	93.77
80–20	93.99	93.56
90–10	93.22	94.18

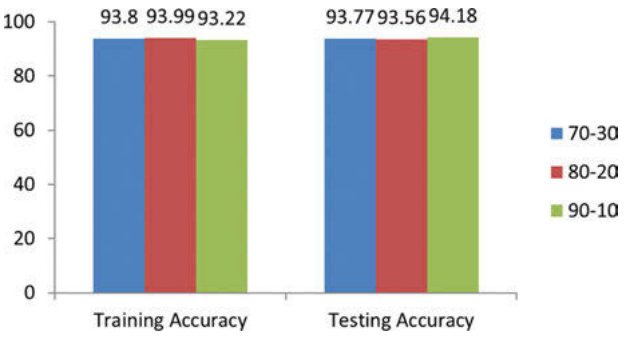


Figure 5: Accuracy plot at different splits.

5.1 ROC curve

ROC curve is a plot drawn between true positive rate and false positive rate at different threshold values [18]. The AUC is used to define the area under the ROC. The AUC value varies between 0 and 1. The proposed model has obtained an AUC score of 0.94. An efficient classification model always has AUC score close to 1. The ROC curve is shown in Figure 6.

5.2 Loss function

The loss function is used to find the difference between the predicted and actual output. The log loss function is used in LR demonstrated in eq. (6). It mainly calculates how much the predicted output is close to the actual [19]:

$$\text{Log loss} = \Sigma - p(\log(p^I) - (1-p)\log(1-p^I) \tag{6}$$

where p is the actual output and p^I is the predicted output.

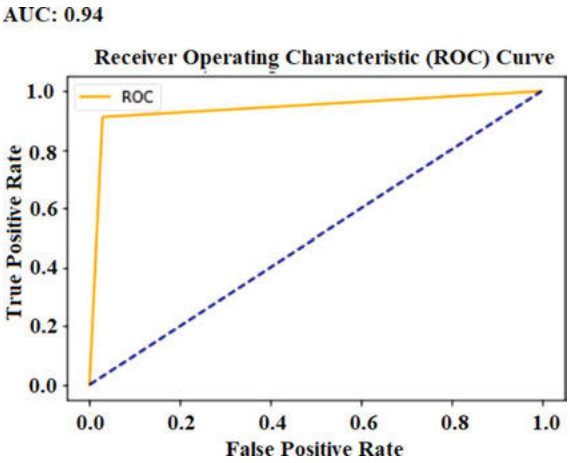


Figure 6: ROC curve.

6 Conclusion

In the above section, a detailed discussion is done on the performance of the proposed classification model using accuracy, precision, recall, F1 score, and AUC score. The result shows that how well the model recognizes the unseen test data correctly. The bigger dataset is used in the proposed work to develop a generic model. A generic model has less chances of suffering from overfitting and underfitting issues. A good AUC score shows its usefulness in real-time environment. The system can be trained and validated using cardiac ailment data of different geographical locations for timely and accurate prediction of the illness.

References

[1] https://www.health.harvard.edu/heart-health/top-five-habits-that-harm-the-heart
[2] https://www.nhlbi.nih.gov/health-topics/coronary-heart-disease
[3] https://www.mayoclinic.org/diseases-conditions/coronary-artery-disease/symptoms-causes/syc-20350613
[4] https://www.medicinenet.com/heart_disease_coronary_artery_disease/article.htm
[5] https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death
[6] Y. Ukai, *et al*. A coronary calcification diagnosis system based on helical CT images, *IEEE Transactions on Nuclear Science*, 45(6), 3083–3088 Dec 1998. 10.1109/23.737668.
[7] https://archive.ics.uci.edu/ml/index.php

[8] A. P. Pawlovsky, "An ensemble based on distances for a kNN method for heart disease diagnosis," 2018 International Conference on Electronics, Information, and Communication (ICEIC), 2018, pp. 1–4, doi: 10.23919/ELINFOCOM.2018.8330570.
[9] H. A. Esfahani and M. Ghazanfari, "Cardiovascular disease detection using a new ensemble classifier," 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), 2017, pp. 1011–1014, doi: 10.1109/KBEI.2017.8324946.
[10] T. Kasbe and R. S. Pippal, "Design of heart disease diagnosis system using fuzzy logic," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), 2017, pp. 3183–3187, doi: 10.1109/ICECDS.2017.8390044.
[11] K. S. Purushottam and R. Sharma, Efficient Heart Disease Prediction System, *Procedia Computer Science*, Volume 85, 2016, Pages 962–969. ISSN 1877 0509 https://doi.org/10.1016/j.procs.2016.05.288. Last accessed on 5 July 2021.
[12] Y. K. Singh, N. Sinha, and S. K. Singh, Heart Disease Prediction System Using Random Forest, Advances in Computing and Data Sciences, 2017, Volume 721, ISBN: 978-981-10-5426-6
[13] S. Bashir, U. Qamar, and M. Younus Javed, "An ensemble based decision support framework for intelligent heart disease diagnosis," *International Conference on Information Society (i-Society 2014)*, 2014, pp. 259–264, doi: 10.1109/i-Society.2014.7009056.
[14] K. Y. Mon Thant and K. T. Nwet, "Comparison of Supervised Machine Learning Models for Categorizing E-Commerce Product Titles in Myanmar Text," 2020 International Conference on Advanced Information Technologies (ICAIT), 2020, pp. 194–199, doi: 10.1109/ICAIT51105.2020.9261779.
[15] K. Hara and K. Nakayamma, "Comparison of activation functions in multilayer neural network for pattern classification," Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94), 1994, pp. 2997–3002 vol.5, doi: 10.1109/ICNN.1994.374710.
[16] D. Swain, S. K. Pani, and D. Swain, "A Metaphoric Investigation on Prediction of Heart Disease using Machine Learning," *2018 International Conference on Advanced Computation and Telecommunication (ICACAT)*, 2018, pp. 1–6, doi: 10.1109/ICACAT.2018.8933603.
[17] D, Swain, S. K. Pani, and D. Swain. An efficient system for the prediction of coronary artery disease using dense neural network with hyper parameter tuning, *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 8, 6S, 2019.
[18] D. Swain, S. Pani, and D. Swain, Diagnosis of coronary artery disease using 1-D convolutional neural network, *International Journal Recent Technology Engineering (IJRTE)*, 8(2), 2959–2966, 2019.
[19] https://www.analyticsvidhya.com/blog/2019/08/detailed-guide-7-loss-functions-machine-learning-python-code/

Anuradha

Design of antenna with biocomputing approach

Abstract: Various biological computing techniques such as genetic algorithm (GA), bacterial foraging optimization (BFO), artificial neural network (ANN) or neural network (NN), and swarm intelligence (SI) techniques such as particle swarm optimization (PSO) and ant colony optimization (ACO) have successfully been implemented in many electromagnetic (EM) applications. These techniques are low cost and very powerful computational techniques to solve the nonlinear and multidimensional EM design problems. GA has become an outdated technique for designing antenna applications and it also takes a long time to arrive at global optimum solution. Like GA, BFO is also very time-consuming to arrive the best solution of the problem, and a high probability of getting stuck in the local optimum solution means weak convergence. ACO is a fast optimization technique but a quite complicated algorithm. On the other hand, PSO has been considered as an efficient optimization algorithm due to its easy calculation and durable solution for EM designs which make PSO comparable to most of the optimization techniques described above. ANN has also been applied to many antennas and EM engineering. In this chapter, we have discussed the methodology of our research work using the merits of ANN and PSO methods. Here, the NN is used for the regression analysis between the antenna's input design parameters and its performance parameters. In order to obtain the final antenna structure at customized frequencies in a short time, SI algorithm is applied to optimize the antenna because NN-evolved methodology responds rapidly during the entire optimization process. Therefore, the strategy adopted using NN and PSO is fast and user-friendly to design a custom-built antenna. The optimized dimensions of the antenna were verified by the neural computer-aided design model to validate the final design of the antenna. Laboratory prototype of customized antenna was fabricated and measured experimentally to cross-validate the developed methodology.

Keywords: antenna, biocomputing methods, ANN, PSO, fractal antenna, Sierpinski's gasket, monopole antenna

Anuradha, Department of Electronics and Communication Engineering, National Institute of Technology, Hamirpur, Himachal Pradesh, India, e-mail: sonanu8@gmail.com

https://doi.org/10.1515/9783110734652-008

1 Introduction

Optimization techniques are becoming popular in engineering design problems, where the main attention of optimization is to maximize or minimize the target [1]. In another field of engineering such as electromagnetic (EM) systems, integral solutions or differential solutions of equations with boundary conditions are considered endless efforts. For example, if we have to determine the current pattern on the antenna, numerical methods are applied for the solution of integral or differential equation. Finding such a solution seems complicated even with powerful computers. Therefore, several optimization algorithms have been applied to EM problems.

Biological computational techniques have been gaining popularity among researchers in every field of engineering over the years [1]. Bioinspired optimization algorithms are sections of optimization tools whose concepts are derived from the biological studies. Widespread use of these optimizers has been successfully implemented where classical optimizers pose problems. The library of biocomputing techniques includes a long list such as artificial neural network (ANN) or simply neural network (NN), fuzzy logic (FL), rough sets (RS), genetic optimization algorithm (GA), particle swarm optimization (PSO) or SO algorithm, or swarm intelligence (SI), ant colony optimization (ACO) algorithm, and bacterial foraging optimization (BFO) algorithm. Recently, other techniques are also being added to this list of biocomputing methods, which have their own merits and demerits.

To find an easy solution to the problem, researchers are using these algorithms such as SO, GA, NN, and BFO. The persistence of these algorithms has been assessed in problems encountered in every engineering domain. EM engineers have used these techniques several times over the past two decades [2]. A lot of research works on applications of biocomputing methods in antenna engineering are available in many books [3–5], journal special issues [6–8], review articles [9–11], and number of research papers have been covered under this topic.

In this chapter, we have discussed the two soft computing techniques, namely, NN and PSO, due to their inherent characteristics for the development of custom-made EM devices. Before discussing both these techniques in detail, its role has been explained which simplifies the traditional EM design process. The end section of this chapter focuses on the methodology formulated using the combination of NN and PSO for the design of user-friendly antennas.

2 Traditional electromagnetic design process

In the initial design process of the EM model, an approximation-based model is simulated on a simulator as shown in Figure 1. An approximate expression is used to calculate the initial values of these variables which gives the closed formula of

these variables. These design parameters are updated in a hit-and-trial way by the designer till the response matches with the user's requirement. In these situations, a designer may require lengthy simulations before targeting the final design. A central processing unit performance of a system on which software is installed depends on system computations and system memory. Furthermore, it is not possible for everyone to afford to buy a full-fledged commercial software.

Figure 1 indicates the positions of the simulator and optimizer where they will be used in a common design process. In this work, we use different biocomputing techniques for simulator and optimizer. A trained neural computer-aided design (CAD) is used in place of simulator, whereas a PSO technique is implemented for the role of the optimizer. The following sections explain these two techniques.

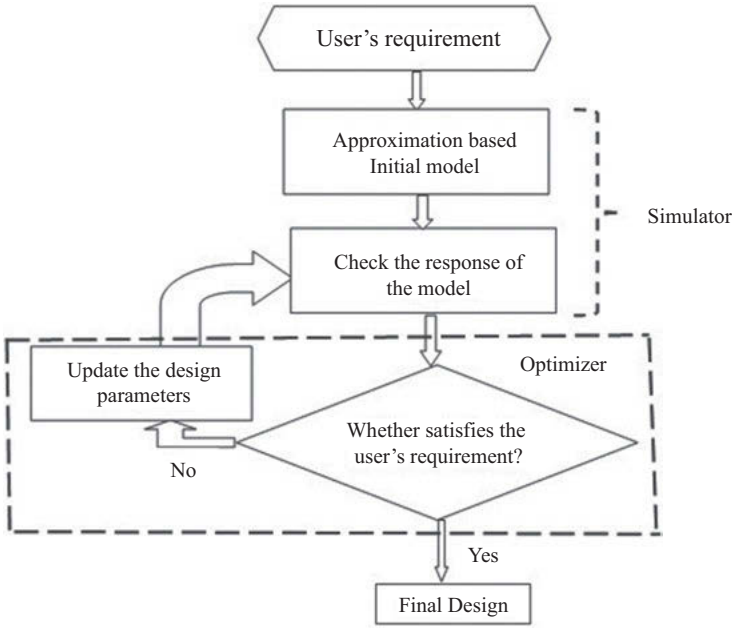


Figure 1: A typical design process.

3 Neural network

The neural network is a machine to process data in the same way as human brain neuron performs a specific function. Neural system is a massively parallelism computing that has a natural tendency to store experimental knowledge and prepare it for use [12].

It is similar to the brain in the following ways:

(a) A neural learning through which information is received by the network.
(b) Interconnected neural strengths are used to store the knowledge.

Since its inception, the neural computing is constantly being exploited for various different applications. Some of the typical examples of neural system that give an impression of its achievements are: (i) text-to-speech transformation, (ii) picture data compression, (iii) recognition of handwriting, (iv) industrial inspection, (v) steering of autonomous vehicles, (vi) explosive detector, (vii) adaptive robot steering, and (viii) signature examiner. These applications of neural computing are due to its three important features: (i) learning, (ii) generalization, and (iii) robustness against noise.

These attractive features also make ANNs a good candidate for solving some of the difficulties in microwave modeling and optimization problems [13].

One of the interesting properties of NNs is its potential to learn from its environment, and to enhance its response through learning. The improvement in performance takes place with time in accordance with some prescribed measures. Learning, in the context of NNs, is clarified as upgradation in weight connection values that results in the capture of information that can be recalled later. Figure 2 shows a pictorial representation of ANN learning using the backpropagation algorithm.

Data is an important component for ANN. The network has to adapt itself to input and corresponding output data pattern, generated by an environment. It may be noted that the dimensionality of the input pattern is not necessarily the same as the output pattern.

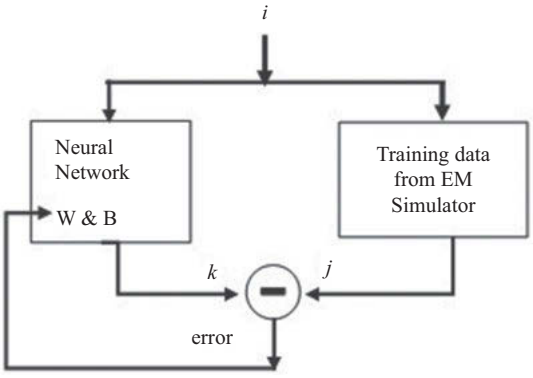


Figure 2: ANN learning in backpropagation mode [40].

The inputs must represent the features of the environment in a parametrical form. Sometimes the data are preprocessed in order to draw the features and a postprocessing is applied at the output. Generally, the generated data are classified into two sets. One

set of data that is used to train the network is called training dataset and another set is called test dataset to find out whether the network has been trained effectively or not. Obviously, the training dataset is different from the test dataset.

Mostly, multilayered feedforward NNs trained in the backpropagation mode are in use for antenna applications. These are best suited for solving mapping formulation type of problem. Although the explanations of this backpropagation learning algorithm can be studied elsewhere in the literature [14], the steps for training can be summarized as follows:

1. The weights have to be initialized
2. The following process is replicated for every pair in the training database
   2.1. Input vector (i) is given
   2.2. Output vector (j) is calculated
   2.3. Error is measured at the output/outmost layer (k)
   2.4. The weights are updated layer by layer to reduce the error so as to achieve satisfactory performance.

ANNs have some peculiar abilities [15]:

i. To solve complex problems, data processing is performed in parallel mode instead of serial mode.
ii. In unsupervised learning, when the NN system self-adapts and determines patterns according to the given training dataset to the network, or in supervised learning, when the NN system is provided with the known output pattern.
iii. Optimal choice of three-layer neural architecture can compute any nonlinearity between input and output data in comparison to other traditional statistical techniques or pattern algorithm.
iv. It takes less time to find out the geometrical parameters of antenna design due to rapid response of the neural system.
v. Hardware design of NN architecture is feasible using the very large-scale integration technology.

NNs have the potential for those problems that are related to the classification pattern method, optimization, self-organized, and associative memory. In this chapter, we have used ANN as a black-box model for the antenna input and its output characteristics. Once a neural CAD is developed, then there will be no need to resimulate the antenna model every time.

4 Particle swarm optimization

As the name highlights, PSO is basically an optimization technique and motivated by social behavior of a scrum of birds and insect swarms. This optimization technique was originally proposed by Kennedy and Eberhart [16]. This and other biocomputing optimization techniques are gaining popularity over the classical optimization techniques for engineering design challenges. There are few noted drawbacks of the classical-based optimizer as follows:

(i) These optimizers are mostly a time-taking process
(ii) They are flop optimizers for solving design problems that are complex, high-dimensional, and involving many variables.
(iii) These optimizers are called as local optimizers and include the derivative techniques.
(iv) These optimizers are not computationally stable in many cases.

These powerful bioinspired optimization techniques such as SO, GA, ACO, and BFO are also known as global search algorithm and probabilistic process. They are less prone to converge to a poor local optima than traditional approaches.

In this work, we have selected PSO, because it is uncomplicated to implement due to less mathematical operations. Although several recent versions of PSO have been searched out but they make PSO computation tough. Both advanced PSO and simple PSO are almost identical in terms of convergence and success rate [17]. In this work, the optimizer aims to update the various parameters of the EM structure in a systematic way during the simulation rather than the trial-and-error approach (Figure 1).

To solve a design problem with a PSO algorithm, initial group of random solutions is called a population. If the number of parameters in the design problem is N, then the PSO must find the optimal solution in the solution's search space of the N-dimension within an acceptable limit to optimize each design parameter. The optimum answer is measured by the value of a cost function. A cost function takes feasible solutions of every N parameter in a defined search region and returns a numeric digit that defines the purity of a solution.

After giving a cost function and search space of the solution, the values of SO's parameters are fixed and then the PSO script is executed. The particle positions and velocities for each particle in the population are changed as per the following expressions [16, 18]:

$$v_{iN} = wv_{iN} + c_1 \,\mathrm{rand}()\,(p_{iN} - x_{iN}) + c_2 \,\mathrm{rand}()\,(p_{gi} - x_{iN}) \tag{1}$$

$$x_{iN} = x_{iN} + v_{iN} \tag{2}$$

where c_1 and c_2 represent the acceleration terms. A parameter c_1 defines how motivated each agent is by storing their best position information while a parameter c_2

defines how much each agent is inspired by the remaining flock. A random number is denoted as rand() in the range of [0, 1], and the random numbers change the pull stochastically relative to the best locations of the particles [18].

The current position vector for the *i*th agent in the ***N***-dimensional search region is expressed as $\boldsymbol{x_i} = (x_{i1}, x_{i2}, \ldots, x_{iN})$. Subscript *i* highlights the number of particles in the population. $\boldsymbol{p_i} = (p_{i1}, p_{i2}, \ldots, p_{iN})$ represents the best previous position vector means the location that gives the best value of the fitness function and termed as a personalbest or pBest of *i*th agent. Subscript (*g*) in the above equation highlights the globalbest or gBest value among all the particles in the population. Velocity vector $v_i = (v_{i1}, v_{i2}, \ldots, v_{iN})$ indicates the rate of change of its position which is velocity for particle *i*. After discovering the two best values, the particle upgrades its velocity as well as positions using eqs. (1) and (2). A flowchart for the classical PSO algorithm is displayed in Figure 3.

Equation (1) has three components: the first one is the "momentum" component, the second one is the "cognitive" component which says that learn from your own personal flying experience, and the last one is the "social" component which signifies the cooperation among the agent's learning from the experience of flock. The inertial weight *w* has been introduced to set an equilibrium between globally and locally explored capabilities. A larger value of (*w*) allows the detection of a global solution while a smaller value of (*w*) allows a local solution.

Another parameter, population size, should be chosen carefully in PSO algorithm. The bigger the population, the deeper the search for solutions, the greater the fitness assessment and computation time. In the PSO, it is concluded that taking a relatively small size of the population can avoid excessive evaluation of the fitness function while saving computation time.

Six different types of boundary conditions were defined for the PSO, to enforce the agent to explore within the allowed solution space during the optimization process. These kinds are designated as absorbing, reflecting, damping, and invisible boundary conditions, and the remaining two are a combination of invisible and reflective boundary and invisible and damping boundary [19].

In order to make the SO code suitable for the available research work, the following SO variables of the algorithm have been analyzed such as (i) the positive constants c_1 and c_2 were tuned at 2 in a hit-and-trial manner; (ii) the inertia weight (*w*) is tuned from 0.9 to 0.4 during the entire optimization; and (iii) variable analysis of SO code found that the population size of up to about 30 is the optimum choice for most problems and small populations of about 10–20 particles have also been useful for engineering problems, which were also addressed in invisible and reflected boundary type of boundary conditions.

The detailed SO algorithm is discussed extensively elsewhere in the literature [11]. Although advance editions for SO techniques are available [20], in the literature, but here, standard form of PSO has been used, because our goal is to demonstrate

the feasibility of the technique for the problem at hand. Table 1 shows the relationship of keywords of PSO algorithm in antenna design problem.

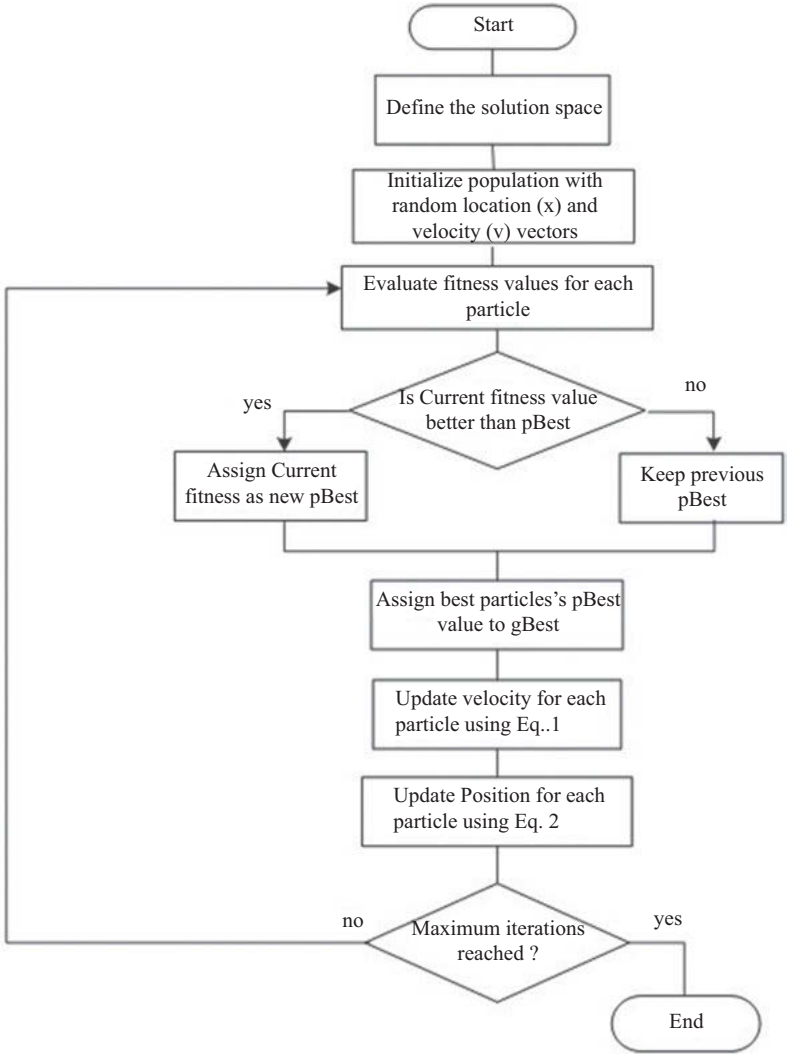


Figure 3: Flowchart depicting the classical PSO algorithm [40].

Table 1: Relationship of key features of SI techniques in antenna engineering [40].

Key features	SI's wordbook	Antenna's wordbook
Solution space of *N*-dimensions or search region	Number of variables to optimize	Number of antenna design parameters to be optimized
Population or swarm	Random initialization of solutions in the *N*-dimensional space	Primary choice of probable solutions, and each contains a class of parameters for the antenna design of the optimization problem
Particle or agent	One individual identity in the population	Individual design dimension of the antenna
Fitness	Reflects a numeric digit defining the purity of that solution	Based on our problem, goodness of fitness may be expressed in terms of minimum return loss, maximum gain, etc., for the antenna.

5 Methodology

In this chapter, we have integrated a neural CAD with PSO optimizer which is a fast and user-friendly design process and replaces the conventional design process of EM devices (especially antennas). An SI technique has been applied to handle the design challenges in the antenna field. The evaluation of fitness function for the SO is done with the previously developed trained NN. The developed methodology consists of ANN and PSO which is shown diagrammatically in Figure 4.

The whole optimization process takes only a few seconds because the response time of NNs is very fast. The custom-made antennas working at user-specified frequencies were designed using the developed methodology. A brief explanation of the neural CAD implementation and PSO implementation procedure is given in the following sections.

5.1 Neural CAD implementation

Several novel applications of NN models for antenna have been developed and reported in the literature. Here, NN is used to map the antenna's input and its output parameters. To eliminate the frequent use of software, NN models have been developed. Figure 5 represents the NN input vectors ($\boldsymbol{i}_{\mathbf{R}}$) and output vectors ($\boldsymbol{k}_{\mathbf{M}}$) for the analysis of antennas, where $\boldsymbol{R}$ denotes the number of elements in input vector ($\boldsymbol{i}$) and $\boldsymbol{M}$ indicates the number of elements in the output vector ($\boldsymbol{k}$). Multilayer perceptron NN is trained in the backpropagation mode for the development of the neural CAD models.

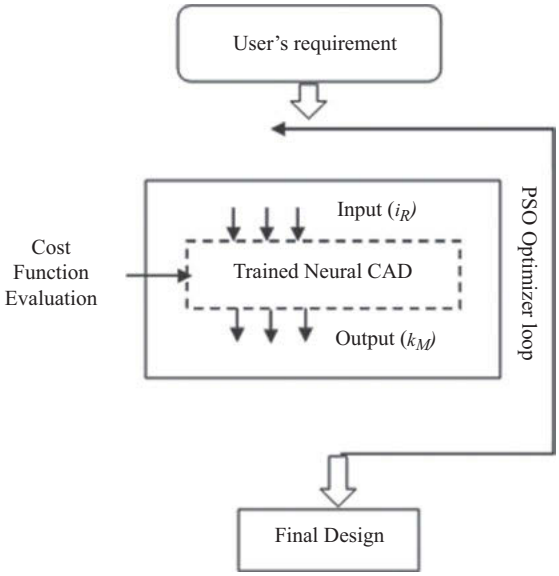


Figure 4: Design strategy for antennas using PSO and NN method [40].

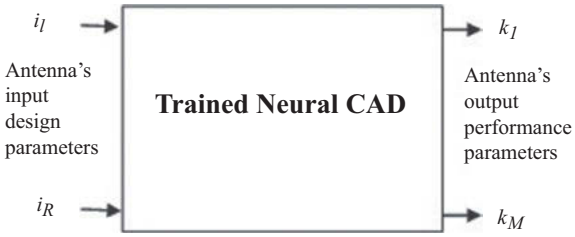


Figure 5: Input and output parameters of the neural CAD analysis of the antenna [40].

By observing the training curve and by testing it with the test dataset, the updated weights and biases values were stored after getting a properly trained NN. These weights and biases can be used to develop neural CAD module, in place of a simulator, to produce the response of the antenna's structure. A trained neural CAD efficiently determines the solution, as well as a long and time-consuming simulation work can also be avoided by using it.

5.2 PSO implementation

The main aim of using PSO in the present problem is to systematize the trial-and-error method of updating each parameter of the EM structure to function at the desired frequencies as fixed by the user. These parameters are the antenna design parameters

that form the population for the PSO. As desired by the PSO algorithm, these parameters have to be specified within their lower and upper bound values. Designer inputs the frequencies on which the antenna has to function. In order to evaluate the cost function, for each set of these antenna parameter, during each iteration of the PSO, the corresponding frequencies have to be found out. At this point, we took the advantage of the developed trained neural CAD. After sufficient number of iterations or when the stopping criterion of the PSO was met, it settles down to a solution giving optimized design of the antenna.

In order to cross-check the validity of the developed methodology, the performance of the optimized antenna was compared with simulation as well as experimental results.

In this work, the developed methodology is applied on fractal antennas that are known as multiband antennas. The Sierpinski gasket fractal was taken as the test fractal antenna to verify the validity of the developed strategy, and the same method can be applied to other antennas.

6 Custom-made fractal antennas

The term "fractal" was initially invented by Benoit Mandelbrot and was derived from the Latin word "fractus" meaning broken or irregular. A lot of research papers have been published in the area of fractal EMs and several fractal geometries have been studied for antenna applications [21–23]. These fractal antennas have a self-similar characteristic, due to which they resonate over multiple bands and are also called multiband antennas. Besides this, fractals are space-filling contours, meaning electrically large features can be efficiently packed into small areas.

A variety of fractal structures and geometries such as Koch-based models, Sierpinski's carpets, the Hilbert fractals, and fractal trees have been used in antenna technology for many years. However, a complete design methodology of these antennas according to the user's wishes is still missing from the literature [24–27].

In this chapter, our goal is to develop a user-friendly design procedure for fractal antennas in a suitable manner. Therefore, custom-made fractal antennas are designed for specific frequencies as per the user's preference.

6.1 Sierpinski's gasket antenna

The Sierpinski gasket antenna in a monopole configuration has been considered to be the most suitable antenna for multi-band applications [28, 29]. This Sierpinski gasket is also known as antenna array [31]. The Sierpinski monopole antenna presents a logarithmic periodic nature with $\boldsymbol{n}$ bands or iterations number with its scale factor (τ)

[29, 30]. The multiband behavior of the Sierpinski gasket to shift in the fractal frequencies depends on the height (H), scaling (τ), and iteration number (n) of gasket-perturbed monopole antenna as shown in Figure 6. The initial height or first height (H) of the gasket monopole is calculated by eq. (3), where c represents the speed of light in vacuum. A second iterated perturbed Sierpinski's gasket antenna has three fractal frequencies (f_{r1}, f_{r2}, f_{r3}) according to its three iterated heights, respectively, and calculated by eq. (4):

$$H = \frac{c}{f_r \times 4} \tag{3}$$

$$\tau = \frac{H_{n+1}}{H_n} = \frac{f_{rn}}{f_{rn+1}} \tag{4}$$

This deviation in frequencies has been found in both classical and perturbed Sierpinski's gasket-based monopole antenna. Some approximate and modified formulas were published for these gasket monopole antennas, to find out the operational band of frequencies [32–36], but all these formulas were not true for low-band frequencies.

Therefore, we have created an NN module that can provide the antenna's resonance bands for the final design of specific applications.

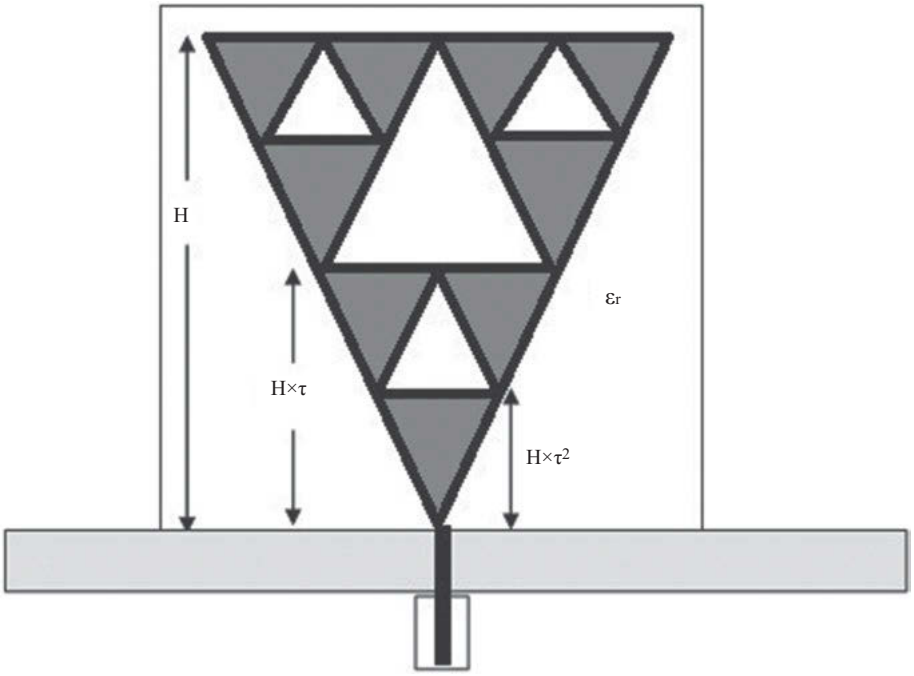


Figure 6: A second iterated perturbed Sierpinski's monopole antenna [39].

6.1.1 Sierpinski's gasket neural modeling

A monopole Sierpinski's gasket antenna has three heights that cause corresponding resonance bands, and the gasket with the highest height belongs to the lowest band of the antenna while the gasket with the lowest height belongs to the highest band of the antenna. Therefore, these frequencies are controlled by the overall height (H) and scale factor (τ) of the monopole gasket antenna.

Neuron models have been developed to map the inputs (geometric design parameters) and outputs (resonance bands and other performing parameters) of perturbed Sierpinski's monopole antenna as well as standard Sierpinski's monopole antennae. Therefore, a generalized neuron model is suitable for both standard and perturbed Sierpinski-based monopole antennas and operates in dual-band and triple-band modes. A second iterated perturbed Sierpinski's gasket antenna has three fractal frequencies.

The overall height (H) of the Sierpinski monopole antenna is extended from 10.0 to 90.0 mm, and the scaling (τ) of the antenna is varied from 0.1 to 0.9 to generate the training database for the NN model. Simulations were performed to find out the operating bands for the second iterated perturbed Sierpinski-based monopole antenna on CST (Computer Simulation Tool) software [37].

Here, this work clarifies that the relative permittivity (ε_r) and thickness (d) of the dielectric were not included in the input set of the training database because the aim of the substrate is to provide a firm support for the antenna. To prepare a database for neural training, a Sierpinski's monopole antenna is printed on a substrate sheet of 1.60 mm thickness (d), and its relative permittivity is defined as 2.5. Most importantly, all the structures of Sierpinski's monopole antenna were excited through a ground plane size of 150 mm × 150 mm perpendicular to the radiating patch [28].

A total database of 153 samples with 2 inputs (H and τ) and their respective 9 outputs (3 resonant bands, 3 respective bandwidths, and 3 respective impedances) was used to generate the training data. For smooth training, this database was preprocessed, and frequencies were transformed into f_{r1}, f_{r2}/f_{r1}, and f_{r3}/f_{r1} as shown in Figure 7.

A three-layer perceptron neural model was trained under the backpropagation algorithm [38] with 1,023 epochs to obtain the least mean square error. After proper training, the network weight values and bias values were stored for further use to design the final antenna structure. Table 2 represents the details of the trained neural network for the perturbed gasket antenna.

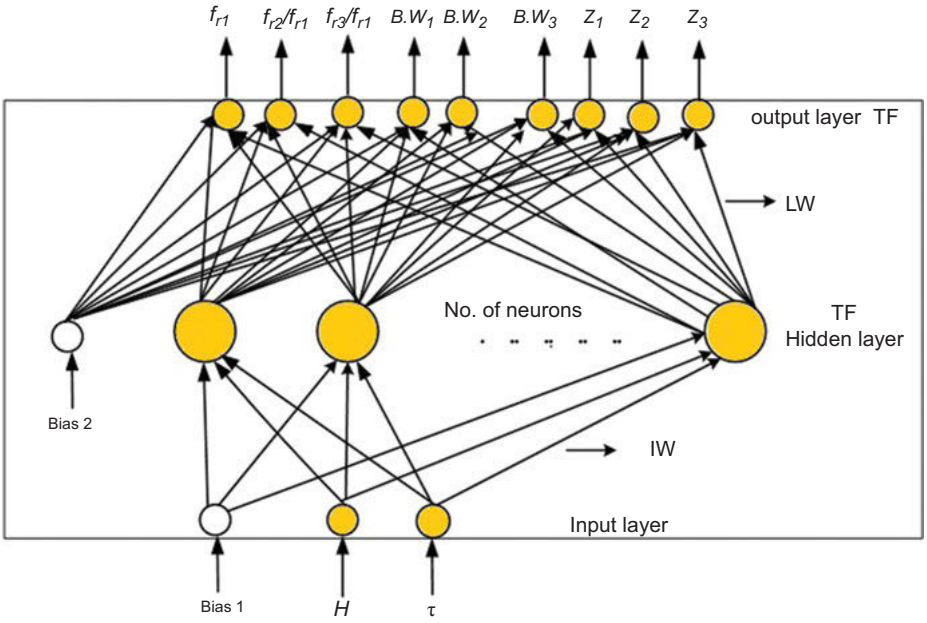


Figure 7: A trained neural architecture of perturbed gasket monopole antenna [40].

6.1.2 Sierpinski's gasket PSO modeling

After NN implementation of the Sierpinski monopole antenna, PSO modeling of this antenna was done in which optimized parameters of the Sierpinski antenna for user-specific frequencies are to be explored. These geometrical parameters of the antenna are as follows:

i. Overall height (H) of the perturbed Sierpinski-based monopole antenna
ii. Scale factor (τ) of the perturbed Sierpinski-based monopole antenna

Each parameter of the antenna was defined in a 2D solution space with their upper and lower bounds, at which the swarm technique has to search for the optimal values of these two parameters.

Each coordinate location of two parameters of antenna in the solution space indicates a possible Sierpinski's monopole antenna design while each coordinate relates to its corresponding parametric value.

Having defined the solution space, we formulate the fitness function that collects values of every two coordinates and reflects a single number indicating the goodness of the Sierpinski-based monopole antenna. A dual-band Sierpinski-based monopole antenna was designed using this method for the industrial, scientific, and medical radio bands (ISM) and wireless local area network applications.

Antenna (ANT): to resonate at 2.40 and 5.20 GHz.

The following fitness function is prepared to determine the geometric parameters of a Sierpinski-based antenna that resonates at frequencies that were defined by the users. The following instantaneous frequencies in the cost function such as f_{r1} and f_{r2} were computed from the trained neural CAD as shown in Figure 7:

$$\text{Cost function} = (2.40 - f_{r1})^2 + (5.20 - f_{r2})^2 \tag{5}$$

Table 2: Description of trained neural architecture of perturbed gasket monopole antenna [40].

NN input $\{i_R\}$	H = [10–90 mm] τ = [0.1–0.9]
NN output $\{k_M\}$	$f_r = [f_{r1}, f_{r2}/f_{r1}, f_{r3}/f_{r1}]$ $BW = [BW_1, BW_2, BW_3]$ $Z = [Z_1, Z_2, Z_3]$
No. of hidden layer neurons	20
Network size	2 × 20 × 9
No. of training dataset	153
Transfer function for hidden layer neuron	$f(x) = \frac{2}{1+e^{-2x}}$
Transfer function for output layer neuron	$f(x) = x$
Learning rate	0.073
Training error	10^{-3}
Epochs	1023

6.2 Results and discussion

The optimized dimensions of the Sierpinski monopole antenna are obtained as shown in Table 3. This table also shows the time of computation in design determination. This design was then verified by using the trained neural CAD and CST tool, as result is shown in Table 4. These values were then used to fabricate the custom-made Sierpinski gasket antennas.

It can be seen from Figure 8, an antenna gasket with optimized dimensions (H = 65 mm, τ = 0.5) is fabricated in the laboratory and then tested with classical feed of 50 ohm. The frequency response curve for the measured and simulated antenna is shown in Figure 9. It is clear from the figure that the antenna efficiently radiates with –10 dB reflections at user's defined frequencies. The radiation pattern always maintains the monopole pattern shape [39] for each defined frequency of a gasket-based monopole antenna and shown in Figure 10. Simulated and measured results of the

Table 3: Optimized Sierpinski's gasket monopole design parameters [40].

User-specific frequencies (GHz)	2.4, 5.2
H (mm)	65
T	0.5
Computation time (s)	2.98

Table 4: Validated results using neural CAD and CST tools for optimized dimensions of gasket monopole antenna.

Optimized dimensions	**Height (*H*): 65 (mm) Scaling (*τ*): 0.5**					
Tool(s)	**CST**			**Neural CAD**		
Frequencies (GHz)	f_{r1}	f_{r2}	f_{r3}	f_{r1}	f_{r2}	f_{r3}
	0.7	2.44	5.2	0.7	2.45	5.18
Input impedances	Z_1	Z_2	Z_3	Z_1	Z_2	Z_3
	34	38	55	38	41	50
Bandwidth(s) (GHz)	BW_1	BW_2	BW_3	BW_1	BW_2	BW_3
	0.070	0.290	1.27	0.100	0.283	1.25

custom-made gasket monopole antenna have been compared for the validation of the developed methodology [39, 40].

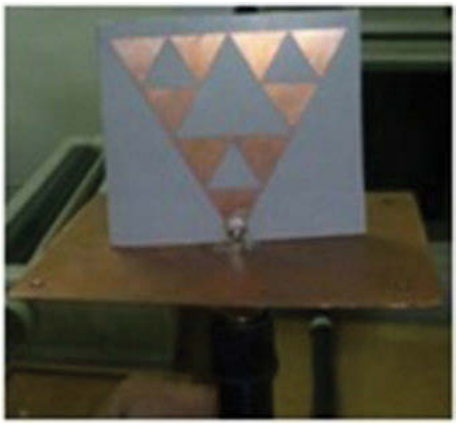


Figure 8: Fabricated Sierpinski's gasket antenna at 2.40 and 5.20 GHz.

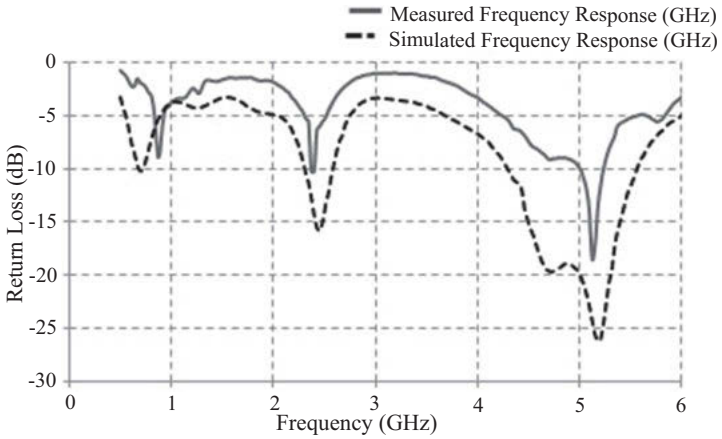


Figure 9: Frequency response of the Sierpinski gasket antenna at user-defined frequencies 2.40 and 5.20 GHz [40].

7 Conclusion

A property of self-similarity in most fractal geometries in the antenna makes it a multiband fractal antenna. But such fractal antennas must resonate at the frequencies specified by the user. In this chapter, a flexible and fast method for making the customized Sierpinski-based monopole antenna was elaborated. The utilization of NN and PSO techniques together permitted the process to eliminate the time-taking simulation procedure of EM devices.

In the first phase of the work, a generalized neural CAD was prepared for the regression problem of the Sierpinski gasket-based monopole antenna, in order to find out the operational band of frequencies of classical as well as perturbed fractal antennas. This NN was then used to evaluate the fitness function of an SI algorithm to get the optimized parameters of the Sierpinski monopole antenna.

With the emerging interest in using fractal theory for multispectral antennas for global system for mobile communication, wireless LAN, and ISM radio band handset applications, the produced strategy may be effectively implemented to design multispectral fractal-based antennas at target frequencies. Although developed methodology has been used for design of Sierpinski's gasket monopole antennas, the same can suitably be expanded for the designing of other fractal antennas, multilayer patch antennas, and frequency-selective surfaces [40–44]. With the introduction of this methodology, we expect a dramatic change in the area of antenna design.

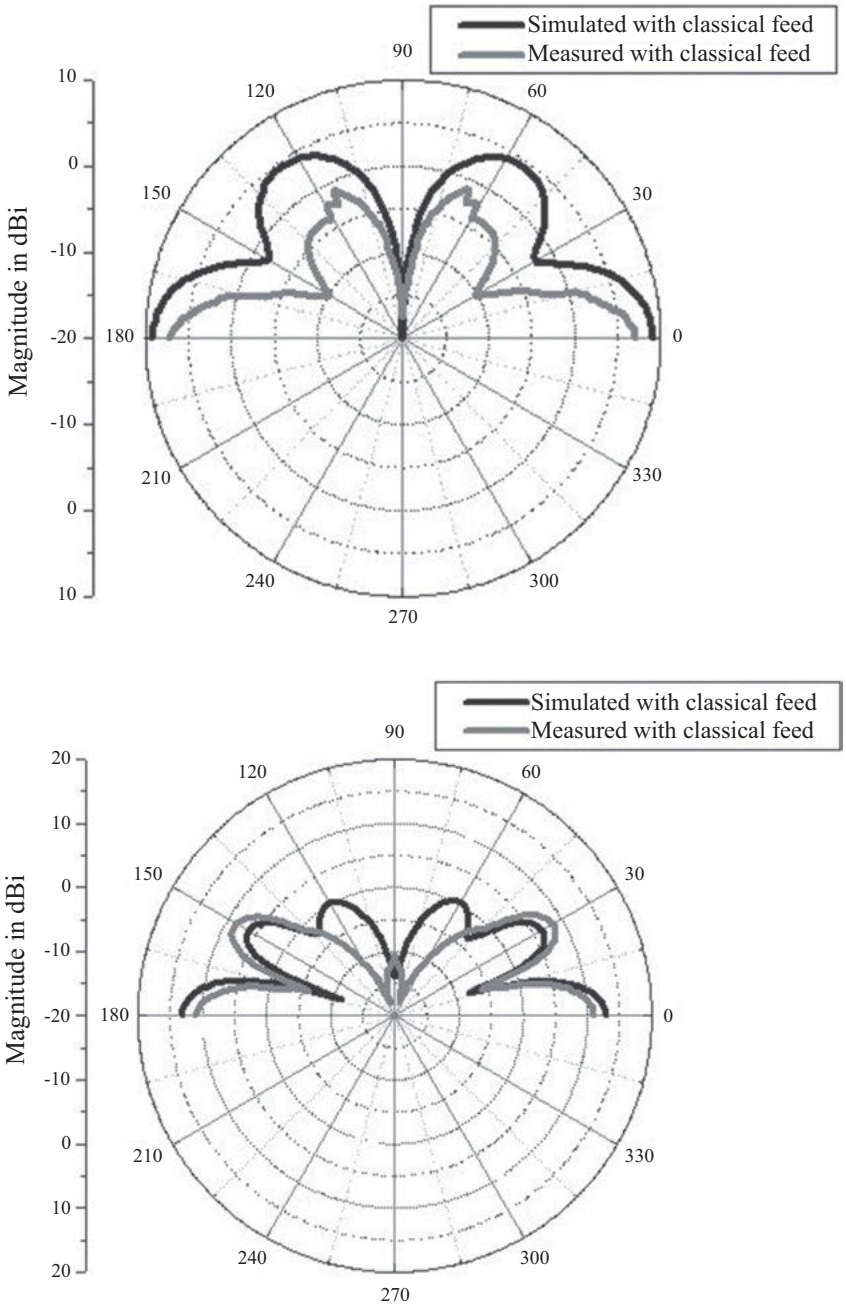


Figure 10: Simulated and measured radiation patterns at the two user-defined frequencies. (a) 2.40 GHz and (b) 5.20 GHz for custom gasket antenna [40].

References

[1] K. Deb, *Optimization for Engineering Design Algorithms and Examples*, Prentice- hall India, 2003.

[2] N. C. Chauhan, M. V. Kartikeyan, and A. Mittal, *Soft Computing Methods for Microwave and Millimeter-Wave Design Problems*, Springer, 2012.

[3] Q. J. Zhang and K. C. Gupta, *Neural Networks for RF and Microwave Design*, Artech House, 2000.

[4] C. Christodoulou and M. Georgiopoulos, *Applications of Neural Networks in Electromagnetics*, Artech House, 2001.

[5] R. L. Haupt and D. H. Werner, *Genetic Algorithms in Electromagnetics*, John Wiley and Sons, 2007.

[6] Q. J. Zhang and G. L. Creech Guest Editors, Special issue on application of artificial neural networks to RF and microwave design, *International Journal of RF and Microwave Computer-Aided Engineering*, 9(3), 1999.

[7] Q. J. Zhang and M. Mongiardo Guest Editors, Special issue on application of artificial neural networks to RF and microwave design, *International Journal of RF and Microwave Computer-Aided Engineering*, 12(1), 1–140, 2002.

[8] Y. R. Samii and C. G. Christodoulou Guest Editors, Special issue on synthesis and optimization techniques in electromagnetics and antenna design, *IEEE Transactions on Antennas and Propagation*, 55(3), 518–522, 2007.

[9] A. Patnaik, D. E. Anagnostou, R. K. Mishra, C. G. Christodoulou, and J. C. Lyke, Application of neural networks in wireless communications, *IEEE Antennas and Propagation Magazine*, 46(3), 130–137, 2004.

[10] S. W. Daniel and E. Michielssen, Genetic algorithm optimization applied to electromagnetics: A review, *IEEE Transactions on Antennas and Propagation*, 45(3), 343–353, 1997.

[11] N. Jin and Y. R. Samii, Particle Swarm Optimization (PSO) for antenna designs in engineering electromagnetics, *Journal of Artificial Evolution and Applications*, 2008, 1–10, 2008.

[12] S. Haykins, *Neural Networks: A Comprehensive Foundation*, IEEE Press/IEEE Computer Society Press, 1994.

[13] A. Patnaik and R. K. Mishra, Artificial neural network techniques in microwave engineering, *IEEE Microwave Magazine*, 1(1), 55–60, 2000.

[14] J. M. Zurada, *Introduction to Artificial Neural Systems*, Jaico Publishing House, 1999.

[15] J. C. Principe, N. R. Euliano, and W. C. Lefebvre, *Neural and Adaptive Systems: Fundamentals through Simulations*, John Wiley & Sons, 2000.

[16] J. Kennedy and R. C. Eberhart, Particle swarm optimization," in *Proc. Int. Conf. On Neural Networks*, Perth, Australia, 1942–1948, 1995.

[17] N. Jin and Y. Samii, Advances in particle swarm optimization for antenna designs: Real-number, binary, single-objective and multiobjective implementations, *IEEE Transactions on Antennas and Propagation*, 55(3), 556–567, 2007.

[18] J. Robinson and Y. R. Samii, Particle swarm optimization in electromagnetic, *IEEE Transactions on Antennas and Propagation*, 52(2), 397–407, 2004.

[19] S. Xu and Y. Samii, Boundary condition in particle swarm optimization revisited, *IEEE Transactions on Antennas and Propagation*, 55(3), 760–765, 2007.

[20] K. E. Parsopoulos and M. N. Vrahatis, Recent approaches to global optimization problems through particle swarm optimization, *International Journal on Natural Computing*, 1(2), 2002.

[21] J. P. Gianvittorio and Y. Rahmat-Samii, Fractals antennas: A novel antenna miniaturization technique and applications, *IEEE Antennas and Propagation Magazine*, 44(1), 20–36, 2002.

[22] D. H. Werner and S. Ganguly, An overview of fractal antenna engineering research, *IEEE Antennas and Propagation Magazine*, 45(5), 38–5, 2003.

[23] D. H. Werner, R. L. Haupt, and P. L. Werner, Fractal antenna engineering: The theory and design of fractal antenna arrays, *IEEE Antennas and Propagation Magazine*, 41(5), 37–59, 1999.
[24] R. Azaro, L. Debiasi, E. Zeni, M. Benedetti, P. Rocca, and A. Massa, A hybrid prefractal three-band antenna for multistandard mobile wireless applications, *IEEE Antennas and Wireless Propagation Letters*, 8, 905–908, 2009.
[25] R. Azaro, G. Boato, M. Donelli, A. Massa, and E. Zeni, Design of a prefractal monopolar antenna for 3.4–3.6 GHz Wi-Max band portable devices, *IEEE Antennas and Wireless Propagation Letters*, 5, 116–119, 2006.
[26] J. Vemagiri, M. Balachandran, M. Agarwal, and K. Varahramyan, Development of compact half-Sierpinski fractal antenna for RFID applications, *IEEE Electronics Letters*, 43(22), 1–2, 2007.
[27] W. J. Krzysztofil, Modified Sierpinski Fractal Monopole for ISM-bands Handset Applications, *IEEE Transactions on Antennas and Propagation*, 57(3), 606–615, 2009.
[28] C. Puente, J. Romeu, R. Pous, X. Garcia, and F. Benitez, Fractal multiband antenna based on the Sierpinski gasket, *Electronics Letters*, 32(1), 1–2, 1996.
[29] C. Puente and J. Romeu, On the behavior of the Sierpinski multiband fractal antenna, *IEEE Transactions on Antennas and Propagation*, 46(4), 517–524, 1998.
[30] C. T. P. Song, P. S. Hall, and H. G. Shiraz, Perturbed Sierpinski multiband fractal antenna with improved feeding technique, *IEEE Transactions on Antennas and Propagation*, 51(5), 1011–1017, 2003.
[31] D. H. Werner and R. Mittra, *Frontiers in Electromagnetic*, IEEE Press, chapters 1–3, 1999.
[32] C. Puente, J. Romeu, R. Bartoleme, and R. Pous, Perturbation of the Sierpinski antenna to allocate operating bands, *Electronics Letters*, 32(24), 2186–2188, 1996.
[33] C. T. P. Song, P. S. Hall, H. Ghafouri-Shiraz, and D. Wake, Sierpinski monopole antenna with controlled band spacing and input impedance, *Electronics Letters*, 35(13), 1036–1038, 1999.
[34] S. R. Best, On the significance of self-similar fractal geometry in determining the multiband behavior of the Sierpinski gasket antenna, *IEEE Antennas and Wireless Propagation Letters*, 1, 22–25, 2002.
[35] J. Romeu and J. Soler, Generalized Sierpinski fractal multiband antenna, *IEEE Transaction on Antennas and Propagation*, 49(8), 1237–1239, 2001.
[36] R. K. Mishra, R. Ghatak, and D. R. Poddar, Design formula for Sierpinski gasket pre-fractal planar monopole antennas, *IEEE Antennas and Propagation Magazine*, 50(3), 104–107, 2008.
[37] CST Microwave Studio®
[38] *http://www.mathworks.com*
[39] Anuradha A. Patnaik, and S. N. Sinha, Design of custom-made fractal multiband antennas using ANN-PSO, *IEEE Antennas & Propagation Magazine*, 53, 94–101, 2011.
[40] Anuradha, Design of fractal antennas and frequency selective surfaces using biologically inspired computational techniques, PhD thesis, Indian Institute of Technology, Roorkee, India, 2013.
[41] Anuradha, A. Patnaik, and S. N. Sinha, Design of Koch fractal based custom-made antennas with ANN-PSO, in *Proc. Int. Symposium on Antenna and Propagation*, Toronto, Ontario, Canada, 2010, 1–4.
[42] Anuradha, A. Patnaik, S. N. Sinha, and J. R. Mosig, Design of customized fractal FSS, in *Proc. Int. Symposium on Antennas and Propagation*, Chicago, IL, USA, 2012, 1–2.
[43] Deepanshu and Anuradha, Fractals for custom monopole antennas solutions: A review, *IETE Technical Review*, in Press, doi: 10.1080/02564602.2020.1837683.
[44] S. K. Jain, A. Patnaik, and S. N. Sinha, Design of custom-made stacked patch antenna: A machine learning approach, *International Journal of Machine Learning and Cybernetics*, 4, 189–194, 2013.

Manoj Tolani, Arun Balodi, Ambar Bajpai, Sunny,
Rajat Kumar Singh

Energy-efficient methods for railway monitoring using WSN

Abstract: Demand for railway transportation is persistently expanding step by step. Due to the high load on the railway track, manual monitoring of the track is not preferable. In this chapter, we will discuss the ongoing research of wireless sensor network (WSN)-based real-time inspection of railway tracks. The energy-efficient WSN is one of the important challenges for the railway track condition monitoring system. The growth of machine learning and artificial intelligence is the center of attraction for researchers for monitoring applications. This chapter deals with all the possible ways of energy-efficient railway track condition monitoring systems. The greater part of the energy utilization of the network happens because of transreceiver radio. Therefore, we have discussed various contention and schedule-based medium access control (MAC) protocols for railway track condition monitoring systems. Inefficient data transmission is also a major challenge of railway monitoring. To address the issue, spatiotemporal aggregation protocol is discussed. The analysis of various MAC protocols shows that energy-efficient hybrid MAC protocols are more reasonable for high data traffic applications. Whereas contention-based MAC protocols and modified IEEE 802.15.4 standard MAC protocols are more suitable for moderate data traffic applications. Artificial intelligence–based techniques strengthen the protocol performance. Early prediction reduces the idle duration and energy consumption. From the analysis, it can be concluded that the efficient spatiotemporal aggregation scheme can reduce the energy consumption to up to 50%.

Keywords: artificial intelligence, machine learning, railway monitoring, WSN, spatiotemporal aggregation

Manoj Tolani, Atria Institute of Technology, Bangalore, e-mail: manoj9721@gmail.com
Arun Balodi, Atria Institute of Technology, Bangalore, e-mail: drbalodi@gmail.com
Ambar Bajpai, Atria Institute of Technology, Bangalore, e-mail: ambarbajpai@gmail.com
Sunny, Indian Institute of Information Technology, Allahabad, e-mail: sunnymeharwal@gmail.com
Rajat Kumar Singh, Indian Institute of Information Technology, Allahabad,
e-mail: rajatsingh@iiita.ac.in

https://doi.org/10.1515/9783110734652-009

1 Introduction

Railway monitoring is the interest of researchers for the last few decades [19, 20]. Due to the increase in the population, the load on the railway tracks is also increasing. Therefore, manual monitoring is not suitable for current scenarios. The researchers are working on a railway track condition monitoring system that can be utilized for efficient monitoring. Due to the heavy load on the railway track, sometimes it is very difficult to monitor the track manually. Therefore, the researchers are working on a wireless sensor network (WSN) based on real-time monitoring. The WSN-based monitoring is more efficient in terms of reliability, efficiency, performance, latency, and various other quality of service parameters. The major challenge in the design of a real-time railway monitoring system is that to reduce the energy consumption of the network. The sensor nodes (SN) will be distributed on the field and will operate using a battery or some energy harvesting system. Therefore, the energy consumption of the network should be minimum. To reduce the energy consumption, it is mandatory to reduce the overall radio trans-receiver power consumption as most of the power dissipation of the SN occurs in radio transmission and reception of packets. The researchers are working in the field of efficient design of medium access control (MAC) [2–18, 70–106] protocol and aggregation protocols for filtration of the redundant data. The work is divided into various categories as shown in Figure 1. In railway monitoring applications, the data generated by the sensors are classified into continuous monitoring (CM) and event-monitoring (EM) data traffic. CM generates high data traffic and EM generates low data traffic. Therefore, to handle such data traffic, researchers have proposed different hybrid protocols. The traditional TDMA [10] based protocol is suitable for CM applications; however, when the nodes have no data to transmit, it wastes its bandwidth. Due to wastage of bandwidth and energy consumption, the protocol is not suitable for event-driven data traffic applications. To optimize the performance bit-mapping [14] based approach is proposed by the researchers. In the bit-mapping approach, the cluster head (CH) node first broadcasts a message to all the SN. Only source nodes that have data packets reserve the data slots. All the other SN (non-source nodes) remain in sleep mode during the contention period. However, the contention period duration causes overhead and wastes the energy of the network. Moreover, the energy-saving for the event-driven application is much higher than the energy wastage due to contention period overhead. Therefore, the bit-map-assisted (BMA) MAC [14] protocol is only suitable for the event-driven application. There are various other hybrid protocols proposed by the researchers to handle the data traffic of both CM and EM. In this chapter, we discuss all the different types of MAC protocols for energy-efficient railway monitoring. There are few other techniques that are useful for the reduction of energy consumption. To reduce energy consumption, the researchers have proposed various protocols for energy-efficient data transmission. Few of the researchers have proposed the artificial intelligence–based machine learning algorithm for early prediction of the aggregated data [107]. The data aggregation and

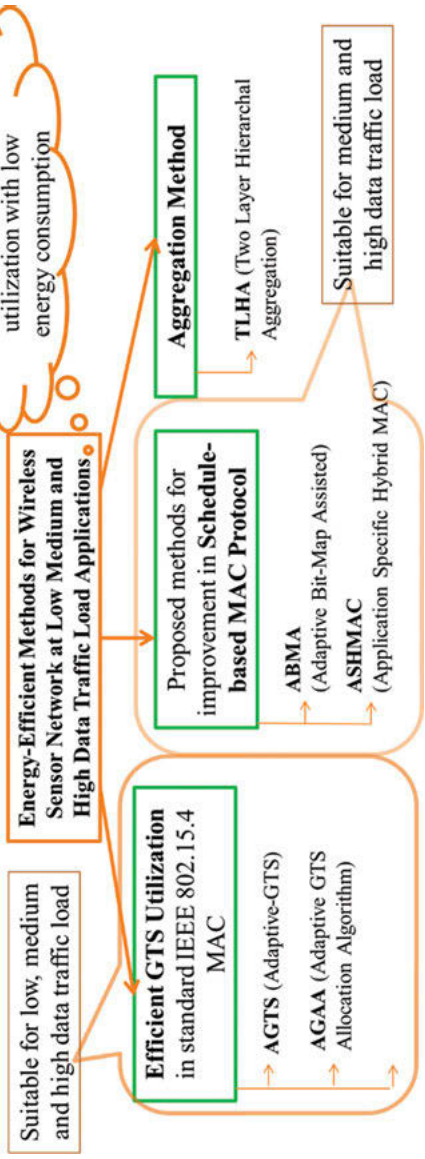


Figure 1: Classification of energy-efficient protocols for railway monitoring.

filtration technique are one of them, which efficiently reduces the unnecessary energy consumption that occurs due to repeated data transmission. The spatiotemporal algorithm is one of the techniques to filter redundant data. The aggregation operation can be performed at SN and CH nodes. The architecture is shown in Figure 2. The aggregation protocol can utilize BMA and TDMA protocols to reduce the energy consumption of the network. A two-layered architecture proposed by Tolani et al. can reduce the energy consumption of the railway network [68]. In this chapter, we have analyzed both the techniques, their methodology, operations, and the effectiveness of the results. Few researchers also proposed the method of WPAN (using ZigBee) and WLAN (using Wi-Fi) hybridization method for the reduction of energy consumption [1]. In the lower layer, they utilized the WPAN for the low-range, low energy consumption of the network. Similarly, for the upper layer, they utilized the WLAN for high-bandwidth and long-range communication. The architecture of the distribution of the SN and CH nodes is shown in Figures 3 and 4, respectively. Few researchers proposed the three levels (alert, early warning, and fire) of artificial intelligence–based early prediction monitoring techniques. The technique can be utilized for early prediction-based railway monitoring applications [108]. The rest of the chapter is described as given. Section 2 describes the literature study. Section 3 describes the various MAC protocols' operation and methodology. Section 4 describes the GTS utilization-based MAC protocol. Section 5 describes the operation of the aggregation protocol operation and methodology. Finally, Section 6 concludes the final findings.

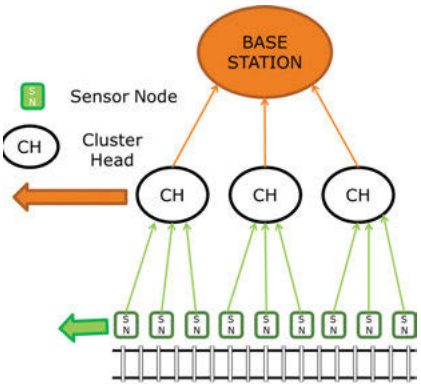


Figure 2: Architecture of two-layer aggregation method for railway monitoring.

2 Literature study

There are various works reported by the researchers related to the MAC protocol and aggregation protocols. Saifullah et al. proposed an energy-efficient MAC protocol for CM data traffic applications [11]. In this protocol, SN check their buffer and turn off the transreceiver if they have no data for transmission. This saves the energy of

the SN and reduces the overall energy consumption of the network. Further, to modify the BMA MAC protocol, the energy-efficient bit map assisted (EBMA) MAC protocol is defined for railway monitoring cases [14]. The EBMA MAC protocol reduces the energy consumption of the network by efficient utilization of the slots. In this protocol, each node uses the piggybacking method for the reservation of the consecutive next slot. This reduces the energy consumption of the contention period. The session and cycle operation of the TDMA [10], EATDMA [11], BMA [14], and E-BMA [15] MAC protocols is shown in Figure 5. Tolani et al. proposed the application-specific hybrid MAC (ASHMAC) protocol [105] to reduce the energy consumption of the network. The architecture of ASHMAC protocol is shown in Figure 6. ASHMAC protocol includes the properties of both TDMA and BMA MAC protocols that reduce the overall energy consumption. For this, each session is divided into two subslots. Subslot-1 is reserved for CM nodes and subslot-2 is reserved for event-based data generation. Therefore, subslot-1 saves the unnecessary energy consumption of CM nodes that occurs in the contention period of the BMA MAC protocol. Similarly, subslot-2 saves the energy consumption that occurs in an unnecessary allotment of the slots to the nonsource nodes. To reduce energy consumption, an energy-efficient hybrid MAC (EE-HMAC) protocol [102] is proposed. The EE-HMAC protocol reduces the energy consumption of the subslot-1 by efficient utilization of the slots. It uses the EATDMA MAC protocol in subslot-1. Few other protocols are also reported, which utilizes the standard IEEE 802.15.4 MAC protocol. Tolani et al. proposed the utilization of the guaranteed time slot for CM nodes and the contention access period for event-driven nodes. However, many times, for time-constraint applications, GTS slots are utilized. Most of the time, the time-constraint SN did not utilize their transmission slots. Therefore, due to slot underutilization, the SN do not optimally utilize the slot. However, the duty cycle can be reduced to save bandwidth. Nevertheless, it reduces the duration of CAP also that has its consequences. Therefore, to improve the protocol performance, independent control of CAP and CFP duty cycle approach is proposed. This method saves energy consumption as well as the bandwidth of the network [103]. Khan et al. [107] proposed the artificial intelligence–based machine learning algorithm for early prediction of the aggregated. Wahyono et al. [108] proposed the three levels (alert, early warning, and fire) of artificial intelligence–based early prediction monitoring technique. The technique can be utilized for early prediction-based railway monitoring applications. Few researchers also proposed an aggregation-based method for the efficient utilization of the slots. It reduces the energy consumption by filtration of the redundant data. Tolani et al. [68] proposed a two-layer filtration method for the reduction of data consumption. In the first layer, SN performs temporal aggregation operation, and in the second layer, the CH node performs a spatiotemporal aggregation operation. With the help of both the aggregation methods and the reclassification approach, the author reduces the energy consumption with optimal utilization of the bandwidth.

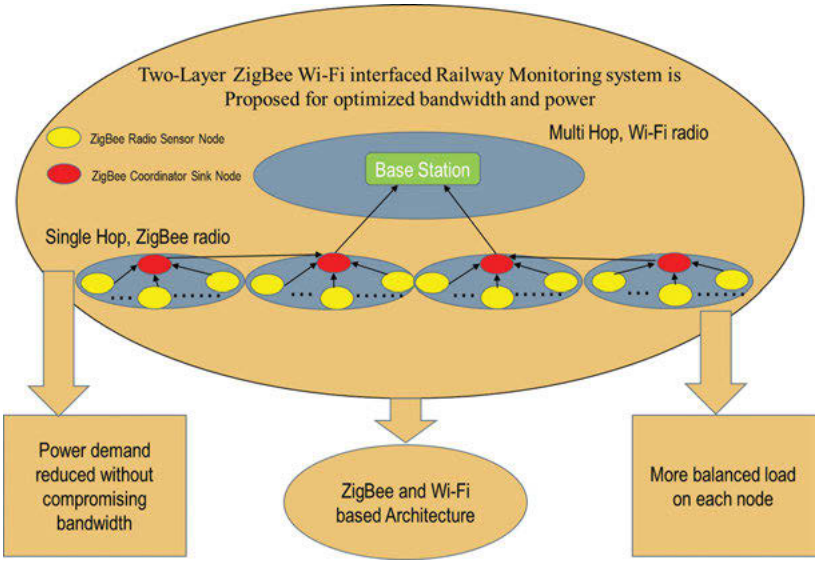


Figure 3: Architecture of two-layer ZigBee Wi-Fi-based railway monitoring system.

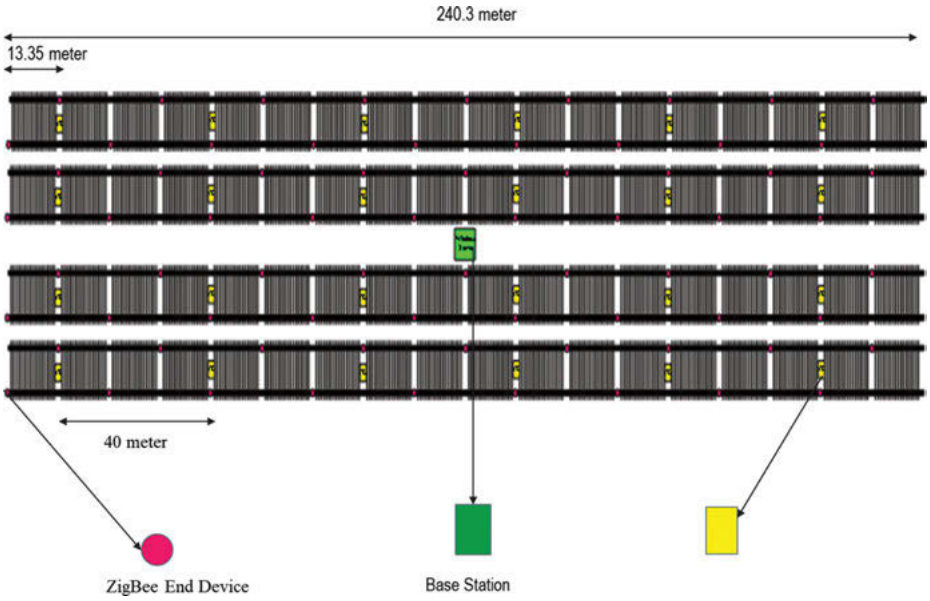


Figure 4: Distribution of sensor nodes and cluster head nodes on railway track.

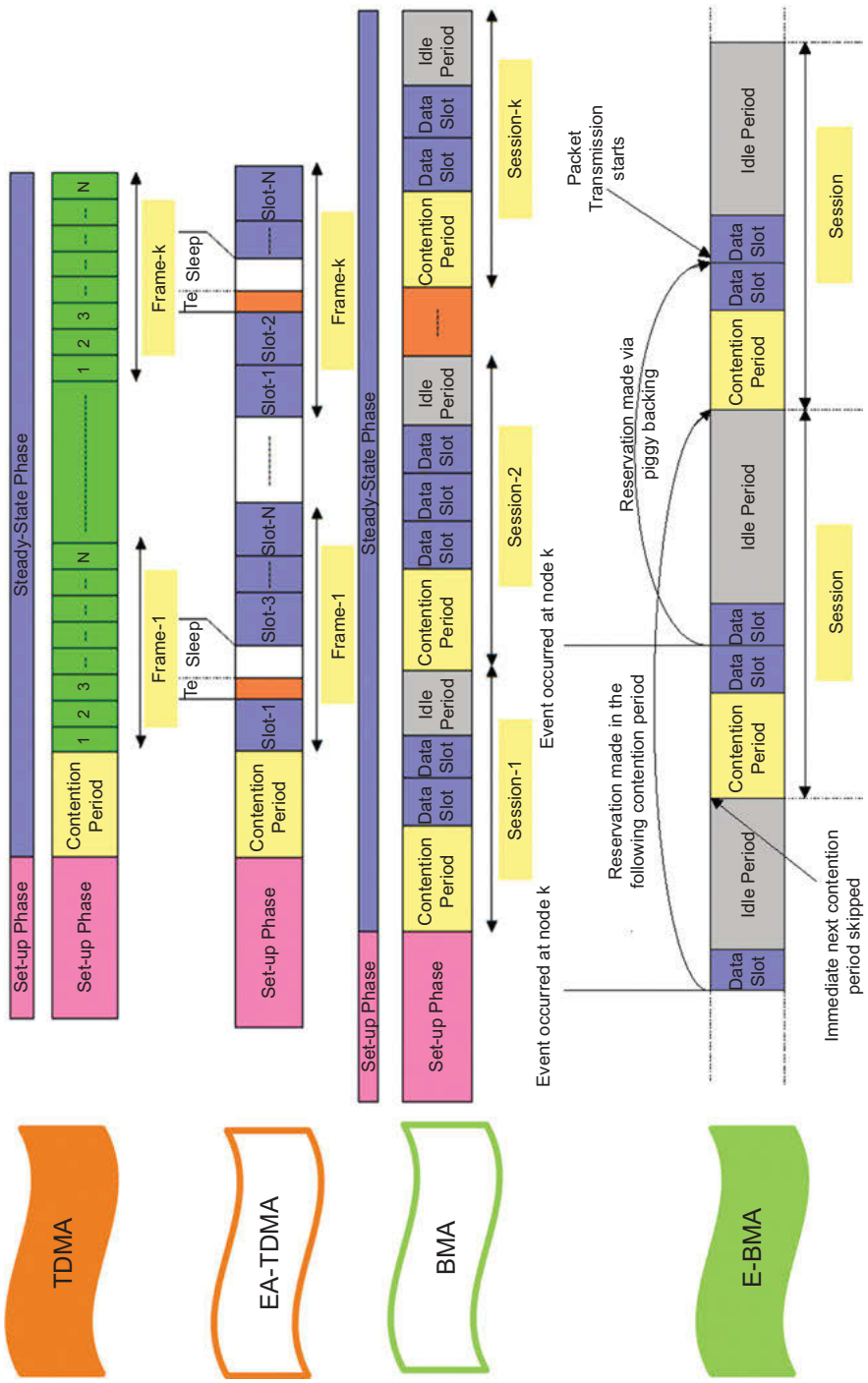


Figure 5: The operation of TDMA, EATDMA, BMA, and E-BMA MAC protocols.

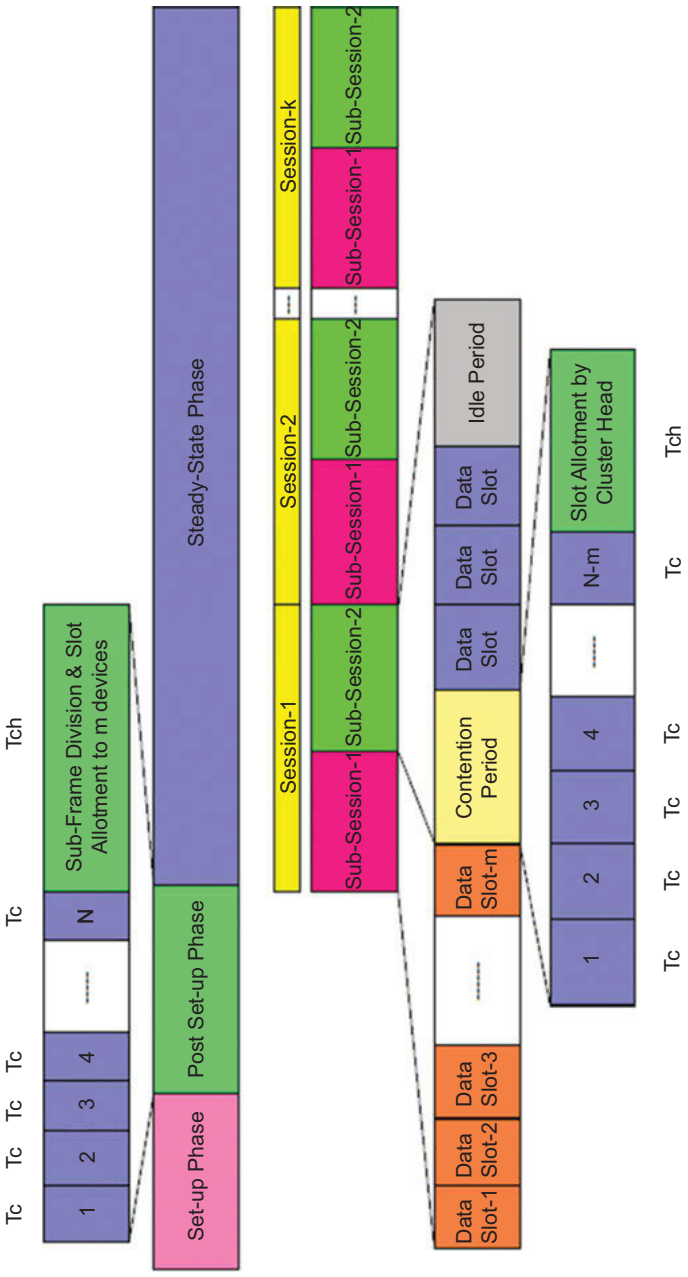


Figure 6: Operation of ASHMAC protocol.

3 Schedule-based MAC protocol

There are many works reported for the energy-efficient MAC protocols. In this section, we will discuss a few of the important works.

3.1 ASHMAC protocol

ASHMAC protocol includes the properties of both TDMA and EATDMA MAC protocols. In this protocol, each session is partitioned into two subslots. The subslot-1 is assigned to the CM nodes and subslot-2 is apportioned to the event-driven nodes. To sort the nodes into continuous and event-driven nodes, another stage is presented in this protocol. The nodes turn on their radio as per their category and based on the reservation of the slot. The excellence of the protocol is that it saves the unnecessary energy consumption that happens to the event-driven nodes. This likewise saves the energy wastage of CM due to unnecessary contention periods. The architecture of the ASHMAC protocol is shown in Figure 6.

3.2 ABMA MAC protocol

ABMA MAC protocol is the modified form of the EBMA MAC protocol [69]. The ABMA MAC protocol consumes lower energy as compared to the ABMA protocol. However, it compromises with the maximum transmission latency. The average transmission latency of the ABMA MAC protocol is less than the EBMA MAC protocol. The architecture of the ABMA MAC protocol is shown in Figure 7. The flowchart of operation is also shown in Figure 8. In this protocol, each node transmits a two-bit status to the piggybacking bit. The CH node can have a lot of the variable number of slots to the SN as per the demand of the SN. The SN demands the data slot based on the available data packets in the buffer.

3.3 EE-HMAC protocol

EE-HMAC protocol is a modified form of ASHMAC protocol [102]. In EE-HMAC protocol, the SNs follow the EATDMA MAC protocol in sub-slot-1. The architecture of the protocol is similar to the ASHMAC protocol. Due to the use of the EATDMA MAC protocol, the energy consumption of the SN (time-constraint nodes) is less in the subslot-1. Therefore, the overall energy consumption of the network reduces.

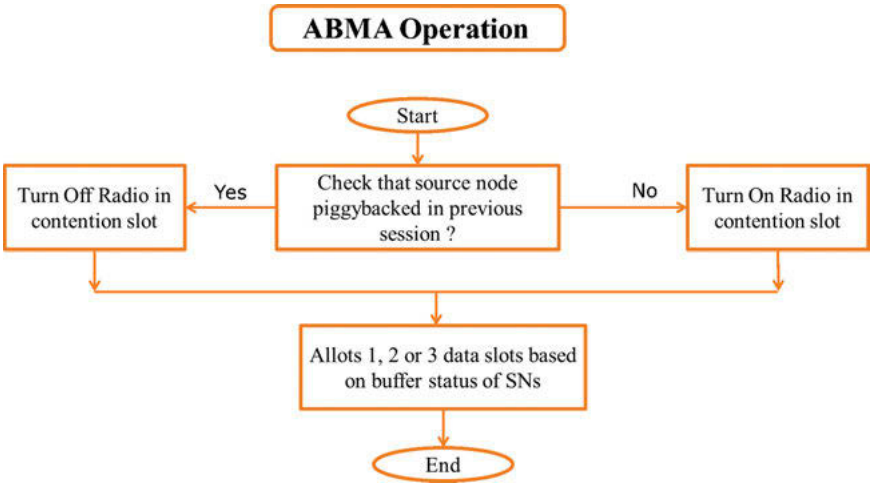


Figure 7: Flowchart of ABMA MAC protocol.

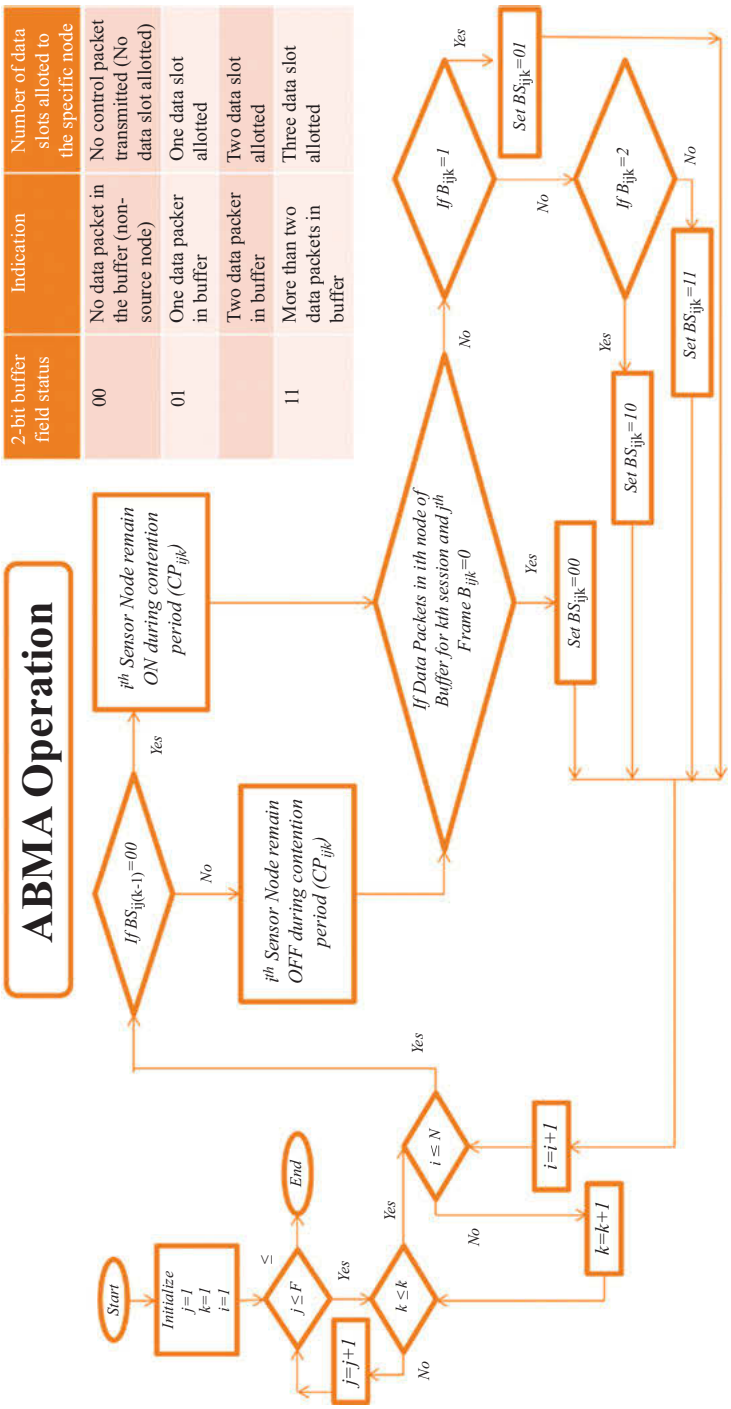
3.4 Result summary

The summary of the result is shown in Table 1. The standard protocol has poor bandwidth utilization. The latency is also very high. During the congestion period, the device fails to operate correctly. TDMA consumes lower energy with respect to the standard MAC protocol. However, it wastes energy by unnecessary allotment of the slots to the nonsource nodes. The EATDMA consumes lower energy than the TDMA. The bandwidth utilization of BMA is better than TDMA and EATDMA. It additionally burns through lower energy for low data traffic conditions. E-BMA absorbs lower energy; however, the most extreme transmission latency of the E-BMA MAC protocol is higher than the BMA. ABMA also consumes lower energy but its transmission latency increases. The EE-HMAC protocol consumes lower energy without compromising with the latency.

4 GTS utilization-based energy-efficient MAC protocol

4.1 AGAA MAC protocol

The adaptive GTS allocation algorithm is a modified form of standard IEEE 802.15.4 MAC protocol [21–49]. The AGAA MAC protocol efficiently handles the time-constraint nodes. It can independently control the duty cycle of the CAP and CFP. Therefore, it not compromises the slot duration. The architecture of the AGAA MAC protocol is shown in Figure 9 [103].



2-bit buffer field status	Indication	Number of data slots alloted to the specific node
00	No data packet in the buffer (non-source node)	No control packet transmitted (No data slot allotted)
01	One data packer in buffer	One data slot allotted
	Two data packer in buffer	Two data slot allotted
11	More than two data packets in buffer	Three data slot allotted

Figure 8: Operation of ABMA MAC protocol.

Table 1: Comparative analysis of various schedule-based MAC protocols.

Protocols and parameters	Energy consumption	Latency	Bandwidth utilization	Suitable application
Standard IEEE 802.15.4	Medium	High	Poor	Event-driven and time-constraint applications
TDMA [10]	Low–medium	Low	Good	CM application
EATDMA [11]	Low–medium (less than TDMA)	Low	Fair	CM application
BMA [14]	Low–medium	Low	Good	Event-driven application
E-BMA [15]	Low	Medium	Good	Event-driven application
ASHMAC [105]	Low–medium	Low	Excellent	Both continuous and event-driven applications
ABMA [99]	Low	Medium	Good	Event-driven application
EEHMAC [102]	Low	Low	Excellent	Both continuous and event-driven applications

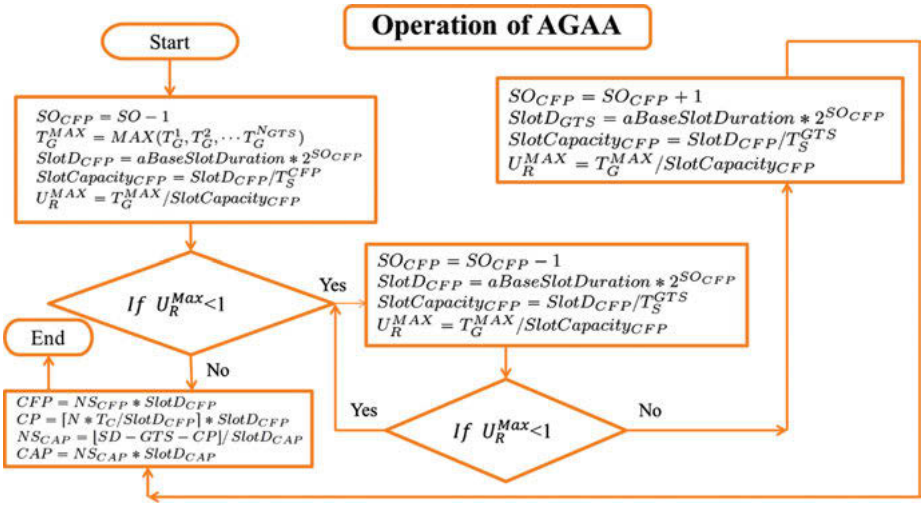


Figure 9: Operation of AGAA MAC protocol.

4.2 Result summary

The summary of the results is shown in Table 2. As shown in Table 2, AGAA MAC protocol is suitable for time-constraint data traffic applications. It consumes lower energy without compromising the latency. The AGAA protocol can handle both CM, event-driven, and time-constraint data traffic application.

Table 2: Performance analysis of AGAA MAC protocol.

Protocols and parameters	Standard IEEE 802.15.4 [102]	AGAA [103]	Schedule-based MAC protocols
Suitability for time-constraint nodes	Yes	Yes	Sometimes
Suitability for non-time-constraint nodes	Yes	Yes	Yes
Energy consumption	High	Medium	Low
Latency	High	Medium	Low
Bandwidth utilization	Poor	Good	Good
Suitable application	Event-driven and time-constraint application	Continuous monitoring, event-driven, and time-constraint application	Continuous monitoring application

5 Aggregation-based method

5.1 Two-layer hierarchal aggregation (TLHA)

A two-layer hierarchal aggregation (TLHA) protocol architecture is shown in Figure 10. The flowchart is shown in Figure 11. The aggregation protocol performs temporal aggregation operations in the lower layer. The SN performs temporal aggregation operation for filtration of the time-correlated data. The CH performs spatiotemporal aggregation operation. The CH classified the data into three classes. If the deviation is lower than the lower threshold, it does not transmit any data to the base station. If it belongs to class 2, the CH transmits only the mean value. However, if the deviation is very large, it belongs to class 3 and transmits all the data to the BS [50–68].

5.2 Experimental setup

The experimental setup of the aggregation protocol is shown in Figure 12. The Arduino board function as an SN and the Raspberry Pi board performs as a CH node. The humidity sensor, temperature sensor, and tilt sensor are used to record the data to perform aggregation operations. NRF2401 is used as a radio transreceiver.

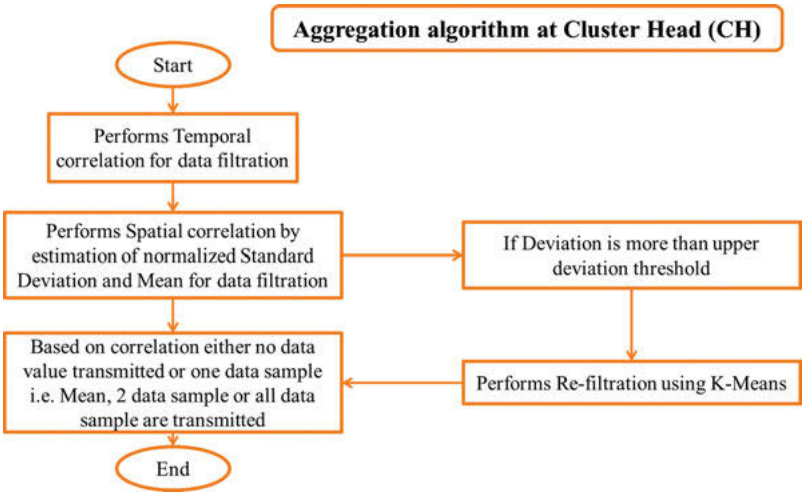


Figure 10: Flowchart of TLHA aggregation protocol.

5.3 Result analysis

As shown in the result section in Table 3 that the spatiotemporal correlation algorithm consumes higher energy but accuracy is better than the DAWF [62]. DAWF consumes lower energy than STCA [48] but compromises the accuracy of the received data. The TLHA [68] protocol performs optimally for all the conditions.

6 Role of Artificial Intelligence for Railway Application

Many of the researchers have proposed artificial intelligence–based monitoring using WSN. Artificial intelligence–based techniques are much useful due to their fault tolerance and adaptive nature. In a railway monitoring application, the technique can be used for the estimation of the early buffer status of the nodes. It will help the CH node to the allotment of the data slots to the nodes. Researchers have proposed various

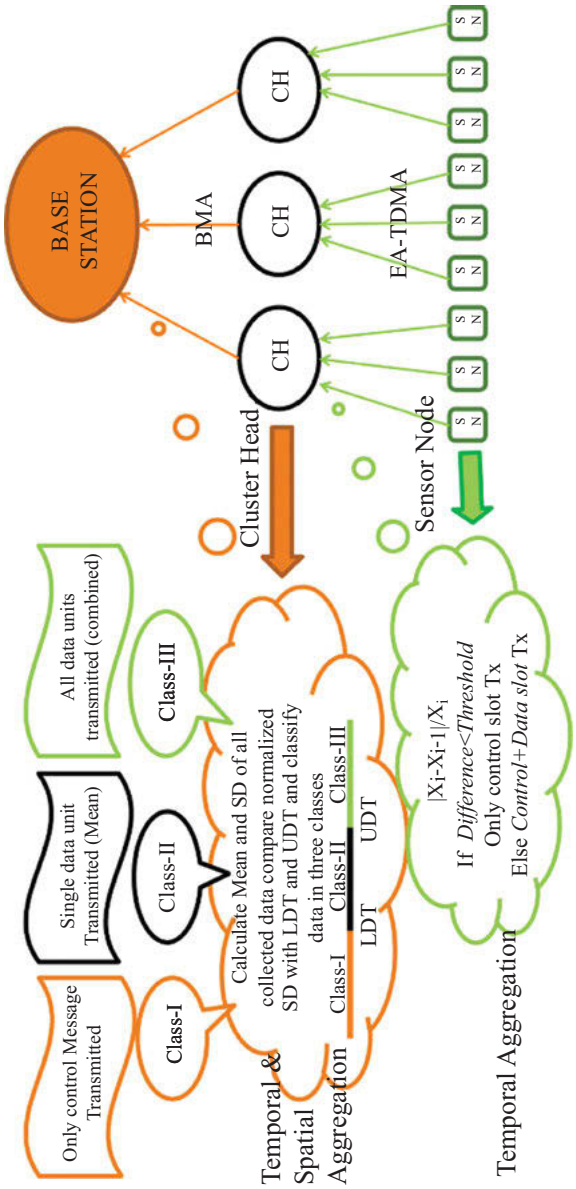


Figure 11: Operation of TLHA aggregation protocol.

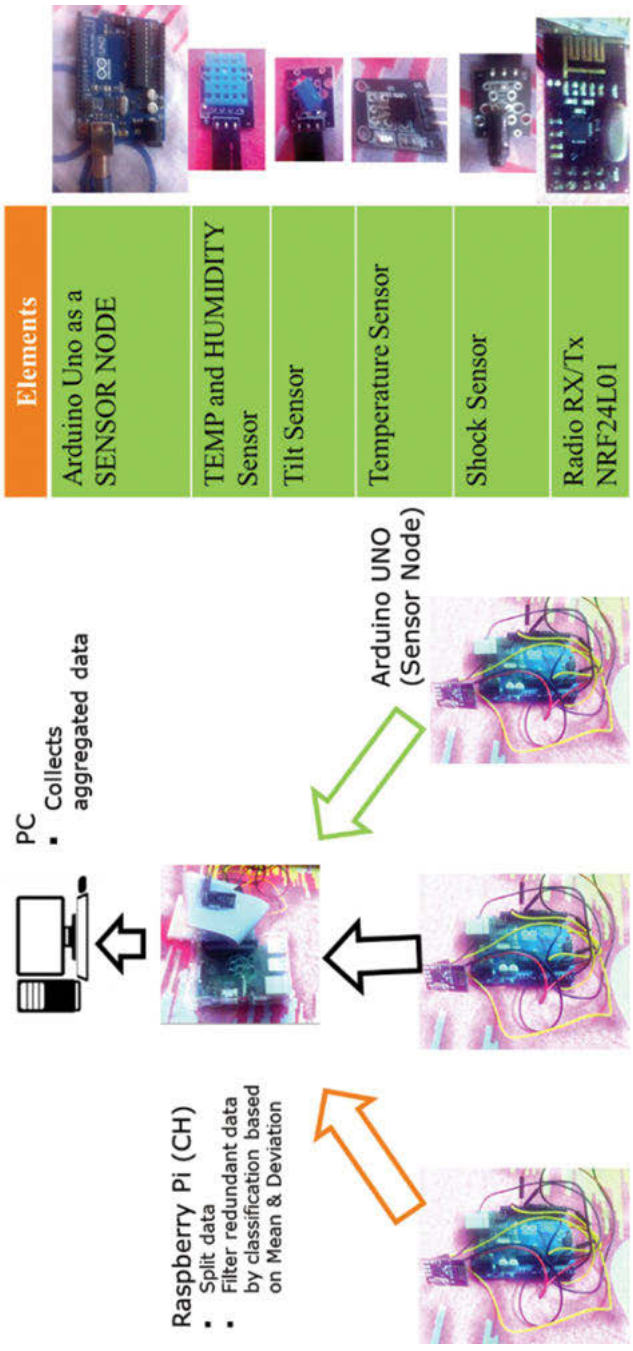


Figure 12: Experimental setup of TLHA aggregation protocol.

Table 3: Performance comparison of aggregation protocol.

Protocols and parameters	STCA [48]	DAWF [62]	TLHA [68]	No aggregation
Energy consumption	High	Medium	Low	Very high
Latency	Medium	Medium	Medium	Low
Filtering efficiency	Low	Moderate	Highest	–
Mean square error	Lowest	Medium	Low	–

techniques for the artificial neural network (ANN)-trained SN and CH nodes [109]. Also, Sun et al. [110] reported backpropagation network data aggregation (BPNDA) technique for efficient data aggregation [110]. The technique can be used for the filtration of redundant data. We can merge the BPNDA approach with the TLHA algorithm to improve the efficiency of the protocol. Artificial intelligence can play a significant role in the improvement of aggregation and MAC protocol for energy-efficient operations.

7 Conclusion

In this chapter, we have analyzed various possible ways to design an efficient railway monitoring system using a WSN. For the efficient design of the railway monitoring system, the energy consumption of the network should be low. In this chapter, the possible ways of reducing energy consumption are discussed. The analysis is classified into two categories. In the first category, we have analyzed various MAC protocols, and in the second category, we have analyzed aggregation protocols. The MAC protocols are further categorized into schedule-based MAC protocol and contention-based MAC protocol. It is found in the analysis that EEHMAC protocol performs better than the existing MAC protocols in terms of energy consumption and latency. In contention-based MAC protocol, AGAA performs ideally. In the case of aggregation protocol, it is discovered clearly that TLHA decreases energy consumption without compromising with latency. The consequences of artificial intelligence–based WSN show the better performance of the protocol as far as energy consumption.

References

[1] M. Tolani, R. K. S. Sunny, K. Shubham, and R. Kumar, Two-layer optimized railway monitoring system using Wi-Fi and ZigBee interfaced WSN, *IEEE Sensors Journal*, 17(7), 2241–2248, April 1, 1 2017.

[2] H. Rasouli, Y. S. Kavian, and H. F. Rashvand, ADCA: Adaptive duty cycle algorithm for energy efficient IEEE 802.15.4 beacon-enabled WSN, *IEEE Sensors Journal*, 14(11), 3893–3902, Nov. 2014.

[3] J. Misic, V. B. Misic, and S. Shafi, Performance of IEEE 802.15.4 beacon enabled PAN with uplink transmissions in non-saturation mode – access delay for finite buffers, First International Conference on Broadband Networks, San Jose, CA, USA, 2004, 416–425.
[4] C. Y. Jung, H. Y. Hwang, D. K. Sung, and G. U. Hwang, Enhanced Markov chain model and throughput analysis of the slotted CSMA/CA for IEEE 802.15.4 under unsaturated traffic conditions, *IEEE Transactions on Vehicular Technology*, 58(1), 473–478, Jan. 2009.
[5] H. Zhang, S. Xin, R. Yu, Z. Lin, and Y. Guo, An adaptive GTS allocation mechanism in IEEE 802.15.4 for various rate applications, 2009 Fourth International Conference on Communications and Networking in China.
[6] C. Ho, C. Lin, and W. Hwang, Dynamic GTS allocation scheme in IEEE 802.15.4 by multi-factor, 2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing.
[7] L. Yang and S. Zeng, A new GTS allocation schemes for IEEE 802.15.4, 2012 5th International Conference on BioMedical Engineering and Informatics (BMEI 2012)
[8] J. Hurtado-López and E. Casilari, An adaptive algorithm to optimize the dynamics of IEEE 802.15.4 network, Mobile Networks and Management, 2013, 136–148.
[9] Standard for part 15.4: Wireless medium access control (MAC) and physical layer specifications for low rate wireless personal area networks (LR-W PAN), IEEE Standard 802.15.4, Jun. 2006.
[10] G. Pei and C. Chien, Low power TDMA in large WSNs, 2001 MILCOM Proceedings Communications for Network-Centric Operations: Creating the Information Force (Cat. No.01CH37277), 2001, 347–351 vol.1.
[11] G. M. Shafiullah, A. Thompson, P. Wolf, and S. Ali, Energy-efficient TDMA MAC protocol for WSNs applications, Proc. 5th ICECE, Dhaka, Bangladesh, Dec. 24–27, 2008, 85–90.
[12] Hoesel and Havinga, A Lightweight Medium Access Protocol (LMAC) for WSNs: Reducing preamble transmissions and transceiver state switches, 1st International Workshop on Networked Sensing Systems, 2004, 205–208.
[13] A. N. Alvi, S. H. Bouk, S. H. Ahmed, M. A. Yaqub, M. Sarkar, and H. Song, BEST-MAC: Bitmap-assisted efficient and scalable TDMA-based WSN MAC protocol for smart cities, *IEEE Access*, 4, 312–322, 2016.
[14] L. Jing and G. Y. Lazarou, A bit-map-assisted energy-efficient MAC scheme for WSNs, Third International Symposium on Information Processing in Sensor Networks, 2004. IPSN 2004, 55–60.
[15] G. Shafiullah, S. A. Azad, and A. B. M. S. Ali, Energy-efficient wireless MAC protocols for railway monitoring applications, *IEEE Transactions on Intelligent Transportation Systems*, 14(2), 649–659, Jun. 2013.
[16] R. K. Patro, M. Raina, V. Ganapathy, M. Shamaiah, and C. Thejaswi, Analysis and improvement of contention access protocol in IEEE 802.15.4 star network, 2007 IEEE International Conference on Mobile Ad hoc and Sensor Systems, Pisa, 2007, 1–8.
[17] S. Pollin et al., Performance analysis of slotted carrier sense IEEE 802.15.4 medium access layer, in *IEEE Transactions on Wireless Communications*, 7(9), 3359–3371, September 2008.
[18] P. Park, P. Di Marco, P. Soldati, C. Fischione, and K. H. Johansson, A generalized Markov chain model for effective analysis of slotted IEEE 802.15.4, IEEE 6th International Conference on Mobile Ad hoc and Sensor Systems Macau, 2009, 130–139.
[19] E. Aboelela, W. Edberg, C. Papakonstantinou, and V. Vokkarane, WSN based model for secure railway operations, in Proc. 25th IEEE Int. Perform., Computer Communication Conf., Phoenix, AZ, USA, 2006, 1–6.

[20] G. Shafiullah, A. Gyasi-Agyei, and P. Wolfs, Survey of wireless communications applications in the railway industry, in Proc. 2nd Int. Conf. Wireless Broadband Ultra Wideband Communication, Sydney, NSW, Australia, 2007, p. 65.
[21] B. Shrestha, E. Hossain, and S. Camorlinga, A Markov model for IEEE 802.15.4 MAC with GTS transmissions and heterogeneous traffic in non-saturation mode, IEEE International Conference on Communication Systems, Singapore, 2010, 56–61
[22] P. Park, P. Di Marco, C. Fischione, and K. H. Johansson, Modeling and optimization of the IEEE 802.15.4 protocol for reliable and timely communications, *IEEE Transactions on Parallel and Distributed Systems*, 24(3), 550–564, March 2013.
[23] A. Farhad, Y. Zia, S. Farid, and F. B. Hussain, A traffic aware dynamic super-frame adaptation algorithm for the IEEE 802.15.4 based networks, IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob), Bandung, 2015, 261–266.
[24] S. Moulik, S. Misra, and D. Das, AT-MAC: Adaptive MAC-frame payload tuning for reliable communication in wireless body area network, *IEEE Transactions on Mobile Computing*, 16(6), 1516–1529, 1 June 2017.
[25] N. Choudhury and R. Matam, Distributed beacon scheduling for IEEE 802.15.4 cluster-tree topology, IEEE Annual India Conference (INDICON), Bangalore, 2016, 1–6.
[26] N. Choudhury, R. Matam, M. Mukherjee, and L. Shu, Adaptive Duty Cycling in IEEE 802.15.4 Cluster Tree Networks Using MAC Parameters, Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Mobihoc '17, 2017, Chennai, India 37: 1–37:2
[27] S. Moulik, S. Misra, and C. Chakraborty, Performance evaluation and delay-power trade-off analysis of zigbee protocol, *IEEE Transactions on Mobile Computing*, 18(2), 404–416, 1 Feb. 2019.
[28] A. Barbieri, F. Chiti, and R. Fantacci, Proposal of an adaptive MAC protocol for efficient IEEE 802.15.4 low power communications, in Proc. IEEE 49th Global Telecommunication Conference, Dec. 2006, 1–5.
[29] B.-H. Lee and H.-K. Wu, Study on a dynamic superframe adjustment algorithm for IEEE 802.15.4 LR-WPAN, in Proc. IEEE Veh. Technol. Conf. (VTC), May 2010, 1–5.
[30] J. Jeon, J. W. Lee, J. Y. Ha, and W. H. Kwon, DCA: Duty-cycle adaptation algorithm for IEEE 802.15.4 beacon-enabled networks, in Proc. 65th IEEE Veh. Technol. Conf., Apr. 2007, 110–113.
[31] R. Goyal, R. B. Patel, H. S. Bhadauria, and D. Prasad, Dynamic slot allocation scheme for efficient bandwidth utilization in Wireless Body Area Network, 9th International Conference on Industrial and Information Systems (ICIIS), Gwalior, 2014, 1–7.
[32] C. Na, Y. Yang, and A. Mishra, An optimal GTS scheduling algorithm for time-sensitive transactions in IEEE 802.15.4 networks, *Computer Networks*, 52(13), 2543–2557, Sept. 2008.
[33] M. S. Akbar, H. Yu, and S. Cang, TMP: Tele-medicine protocol for slotted 802.15.4 with duty-cycle optimization in wireless body area sensor networks, *IEEE Sensors Journal*, 17(6), 1925–1936, March15, 15 2017.
[34] A. Koubaa, M. Alves, and E. Tovar, GTS allocation analysis in IEEE 802.15.4 for real-time WSNs, Proceedings 20th IEEE International Parallel and Distributed Processing Symposium, Rhodes Island, 2006, 8.
[35] P. Park, C. Fischione, and K. H. Johansson, Modeling and stability analysis of hybrid multiple access in the IEEE 802.15.4 protocol, *ACM Transactions on Sensor Networks*, 9(2), 13, 2013.
[36] A. Alvi, R. Mehmood, M. Ahmed, M. Abdullah, and S. H. Bouk, Optimized GTS utilization for IEEE 802.15.4 standard, International Workshop on Architectures for Future Mobile Computing and Internet of Things, 2018.

[37] J. Song, J. Ryool, S. Kim, J. Kim, H. Kim, and P. Mah, A dynamic GTS allocation algorithm in IEEE 802.15.4 for QoS guaranteed real-time applications, IEEE International Symposium on Consumer Electronics, 2007. ISCE 2007.
[38] H. Lee, K. Lee, and Y. Shin, A GTS allocation scheme for emergency data transmission in cluster-tree WSNs, ICACT2012, Feb (2012), 19–22.
[39] X. Lei, Y. Choi, S. Park, and S. Hyong Rhee, GTS allocation for emergency data in low-rate WPAN, 18th Asia-Pacific Conference on Communications (APCC), October 2012.
[40] L. Yang and S. Zeng, A new GTS allocation schemes for IEEE 802.15.4, 2012 5th International Conference on BioMedical Engineering and Informatics (BMEI 2012)
[41] L. Cheng, A. G. Bourgeois, and X. Zhang, A new GTS allocation scheme for IEEE 802.15.4 networks with improved bandwidth utilization, International Symposium on Communications and Information Technologies, 2007.
[42] M. U. H. Al Rasyid, B. Lee, and A. Sudarsono, PEGAS: Partitioned GTS Allocation Scheme for IEEE 802.15.4 Networks, International Conference on Computer, Control, Informatics and Its Applications, 2013.
[43] S. Roy, I. Mallik, A. Poddar, and S. Moulik, PAG-MAC: Prioritized allocation of GTSs in IEEE 802.15.4 MAC protocol – A dynamic approach based on analytic hierarchy process, 14th IEEE India Council International Conference (INDICON), December 2017.
[44] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, *IEEE Wireless Communication Transaction*, 1(4), 660–670, Oct. 2002.
[45] A. Philipose and A. Rajesh, Performance analysis of an improved energy aware MAC protocol for railway systems, 2nd International Conference on Electronics and Communication Systems (ICECS), Coimbatore, 2015, 233–236.
[46] D. Kumar and M. P. Singh, Bit-map-assisted energy-efficient MAC protocol for WSNs, *International Journal of Advanced Science and Technology*, 119(2018), 111–122.
[47] E. J. Duarte-Melo and M. Liu, Analysis of energy-consumption and lifetime of heterogeneous WSNs, Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE, 2002, 21–25 vol. 1.
[48] V. C. Shabna, K. Jamshid, and S. Manoj Kumar, Energy minimization by removing data redundancy in WSNs, 2014 International Conference on Communication and Signal Processing, Melmaruvathur, 2014, 1658–1663.
[49] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. Hanzo, Network-lifetime maximization of WSNs, *IEEE Access*, 3, 2191–2226, 2015.
[50] R. Rajagopalan and P. K. Varshney, Data-aggregation techniques in sensor networks: A survey, *IEEE Communications Surveys & Tutorials*, 8(4), 48–63, Fourth Quarter 2006.
[51] P. Jesus, C. Baquero, and P. S. Almeida, A survey of distributed data aggregation algorithms, *IEEE Communications Surveys Tutorials*, 17(1), 381–404, First quarter 2015.
[52] F. Zhou, Z. Chen, S. Guo, and J. Li, Maximizing lifetime of data-gathering trees with different aggregation modes in WSNs, *IEEE Sensors Journal*, 16(22), 8167–8177, Nov.15, 2016.
[53] N. Sofra, T. He, P. Zerfos, B. J. Ko, K. W. Lee, and K. K. Leung, Accuracy analysis of data aggregation for network monitoring, MILCOM 2008-2008 IEEE Military Communications Conference, San Diego, CA, 2008, 1–7.
[54] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, Energy-efficient communication protocols for wireless microsensor networks, Proceedings of the 33rd Hawaiian International Conference on Systems Science (HICSS), January 2000.
[55] J. Liang, J. Wang, J. Cao, J. Chen, and M. Lu, An efficient algorithm for constructing maximum lifetime tree for data gathering without aggregation in WSNs, 2010 Proceedings IEEE INFOCOM, San Diego, CA, 2010, 1–5.

[56] Y. Wu, Z. Mao, S. Fahmy, and N. B. Shroff, Constructing maximum-lifetime data-gathering forests in sensor networks, *IEEE/ACM Transactions on Networking*, 18(5), 1571–1584, Oct. 2010.
[57] D. Luo, X. Zhu, X. Wu, and G. Chen, Maximizing lifetime for the shortest path aggregation tree in WSNs, 2011 Proceedings IEEE INFOCOM, Shanghai, 2011, 1566–1574.
[58] C. Hua and T. S. P. Yum, Optimal routing and data aggregation for maximizing lifetime of WSNs, *IEEE/ACM Transactions on Networking*, 16(4), 892–903, Aug. 2008.
[59] K. Choi and K. Chae, Data aggregation using temporal and spatial correlations in Advanced Metering Infrastructure, The International Conference on Information Networking 2014 (ICOIN2014), Phuket, 2014, 541–544
[60] L. A. Villas, A. Boukerche, D. L. Guidoni, A. B. F. Horacio, R. B. de Araujo, and A. A. F. Loureiro, An energy-aware spatio-temporal correlation mechanism to perform efficient data collection in WSNs, *Computer Communications*, 36(9), 1054–1066, 2013.
[61] C. Liu, K. Wu, and J. Pei, An energy-efficient data collection framework for WSNs by exploiting spatiotemporal correlation, *IEEE Transactions on Parallel and Distributed Systems*, 18(7), 1010–1023, July 2007.
[62] S. Kandukuri, J. Lebreton, R. Lorion, N. Murad, and J. D. Lan-Sun-Luk, Energy-efficient data aggregation techniques for exploiting spatio-temporal correlations in WSNs, 2016 Wireless Telecommunications Symposium (WTS), London, 2016, 1–6.
[63] D. Mantri, N. R. Prasad, and R. Prasad, *Wireless Personal Communications*, 5, 2589, 2014, doi: https://doi.org/10.1007/s11277-013-1489-x.
[64] D. Mantri, N. R. Prasad, R. Prasad, and S. Ohmori, Two Tier Cluster based Data Aggregation (TTCDA) in WSN, 2012 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS),
[65] N. D. Pham, T. D. Le, K. Park, and H. Choo, SCCS: Spatiotemporal clustering and compressing schemes for efficient data collection applications in WSNs, *International Journal of Communication Systems*, 23, 1311–1333.
[66] L. A. Villas, A. Boukerche, H. A. B. F. de Oliveira, R. B. de Araujo, and A. A. F. Loureiro, A spatial correlation aware algorithm to perform efficient data collection in WSNs, *Ad Hoc Networks*, 12, 69–85, 2014, ISSN 1570-8705.
[67] B. Krishnamachari, D. Estrin, and S. B. Wicker, The impact of data aggregation in WSNs, in: ICDCSW '02: Proceedings of the 22nd International Conference on Distributed Computing Systems, IEEE Computer Society, Washington, DC, USA, 2002, 575–578.
[68] M. Tolani, Sunny, and R. K. Singh, Lifetime improvement of WSN by information sensitive aggregation method for railway condition monitoring, *Ad Hoc Networks*, 87, 128–145, 2019, ISSN 1570-8705, doi: https://doi.org/10.1016/j.adhoc.2018.11.009
[69] M. Tolani, Sunny, and R. K. Singh, Energy efficient adaptive bit-map-assisted medium access control protocol, *Wireless Personal Communication*, 108, 1595–1610, 2019, doi: https://doi.org/10.1007/s11277-019-06486-9.
[70] J. B. MacQueen, Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1:281–297.
[71] J. Mišić, S. Shafi, and V. B. Mišić., The impact of MAC parameters on the performance of 802.15.4 PAN, *Ad Hoc Network*, 3(5), 509–528, September 2005, doi: http://dx.doi.org/10.1016/j.adhoc.2004.08.002.
[72] An IEEE 802.15.4 complaint and ZigBee-ready 2.4 GHz RF transceiver, *Journal of Microwave*, 47(6), 130–135, Jun. 2004.
[73] W. Dargie and C. Poellabauer, *Fundamentals of WSNs: Theory and Practice*, Wiley Publishing, 2010.

[74] P. Park, C. Fischione, and K. H. Johansson, Modeling and stability analysis of hybrid multiple access in the IEEE 802.15.4 protocol, *ACM Transactions on Sensor Networks*, 9(2), 55, Article 13 April 2013.
[75] Y. Zhan, Y. Xia, and M. Anwar, GTS size adaptation algorithm for IEEE 802.15.4 wireless networks, *Ad Hoc Networks*, 37, Part 2, 486–498, 2016, ISSN 1570-8705, doi: https://doi.org/10.1016/j.adhoc.2015.09.012.
[76] I. Iala, I. Dbibih, and O. Zytoune, Adaptive duty-cycle scheme based on a new prediction mechanism for energy optimization over IEEE 802.15.4 wireless network, *International Journal of Intelligent Engineering and Systems*, 11(5), 2018, doi: 10.22266/ijies2018.1031.10.
[77] A. Boulis, Castalia: A simulator for WSNs and Body Area Networks, user's manual version 3.2, 2011, NICTA.
[78] P. Kolakowskil, J. Szelazek, K. Sekula, A. Swiercz, K. Mizerski, and P. Gutkiewicz, Structural health monitoring of a railway truss bridge using vibration-based and ultrasonic methods, *Smart Materials and Structures*, 20, (3), 035016, Mar, 2011.
[79] T. A. Al-Janabi and H. S. Al-Raweshidy, An energy efficient hybrid MAC protocol with dynamic sleep-based scheduling for high density IoT networks, *IEEE Internet of Things Journal*, 6(2), 2273–2287, April 2019.
[80] M. T. Penella-López and M. Gasulla-Forner, *Powering Autonomous Sensors: An Integral Approach with Focus on Solar and RF Energy Harvesting*, Springer Link, 2011, doi: https://doi.org/10.1007/978-94-007-1573-8.
[81] H. Farag, M. Gidlund, P. Österberg, and A. Delay-Bounded, MAC protocol for mission- and time-critical applications in industrial WSNs, vol. 18, no. 6, *IEEE Sensors Journal*, 18(6), 2607–2616, 2018.
[82] C. H. Lin, K. C. J. Lin, and W. T. Chen, Channel-aware polling-based MAC protocol for body area networks: Design and analysis, *IEEE Sensors Journal*, 17(9), 2936–2948, 2017.
[83] V. J. Hodge, S. O'Keefe, M. Weeks, and A. Moulds, WSNs for condition monitoring in the railway industry: A survey, *IEEE Transactions on Intelligent Transportation Systems*, 16(3), 1088–1106, 2015.
[84] W. Ye, J. Heidemann, and D. Estrin, An energy-efficient MAC protocol for WSNs, vol.3, Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, 1567–1576 (2002)
[85] S. Siddiqui, S. Ghani, and A. A. Khan, ADP-MAC: An adaptive and dynamic polling-based MAC protocol for WSNs, *IEEE Sensors Journal*, 18(2), 860–874, 2018.
[86] M. Stem and R. H. Katz, Measuring and reducing energy-consumption of network interfaces in hand held devices, *IEICE Transactions on Communications*, E80-B(8), 1125–1131, 1997.
[87] A. H. Lee, M. H. Jing, and C. Y. Kao, LMAC: An energy-latency trade-off MAC protocol for WSNs, International Symposium on Computer Science and its Applications, Hobart, ACT, 2008, 233–238.
[88] H. Karl and A. Willig, *Protocols and Architectures for WSNs*, John Wiley & Sons Ltd, 2005.
[89] C. Balakrishnan, E. Vijayalakshmi, and B. Vinayagasundaram, An enhanced iterative filtering technique for data aggregation in WSN, 2016 International Conference on Information Communication and Embedded Systems (ICICES), Chennai, 2016, 1–6.
[90] P. Nayak and A. Devulapalli, Fuzzy logic-based clustering algorithm for WSN to extend the network lifetime, *IEEE Sensors Journal*, 16(1), 137–144, Jan.1, 2016.
[91] M. Tolani, A. Bajpai, Sunny, R. K. Singh, L. Wuttisittikulkij, and P. Kovintavewat, Energy efficient hybrid medium access control protocol for WSN, The 36th International Technical Conference on Circuits/Systems, Computers and Communications, June 28th(Mon) – 30th (Wed) / Grand Hyatt Jeju, Republic of Korea, 2021.

[92] M. G. C. Torres, energy-consumption in WSNs Using GSP, University of Pittsburgh, M.Sc. Thesis, April.
[93] K. Chebrolu, B. Raman, N. Mishra, P. Valiveti, and R. Kumar, Brimon: a sensor network system for railway bridge monitoring, in Proc. 6th Int. Conf. Mobile Syst., Appl. Serv., Breckenridge, CO, USA, 2008, 2–14.
[94] A. Pascale, N. Varanese, G. Maier, and U. Spagnolini, A WSN architecture for railway signalling, in Proc. 9th Italian Netw. Workshop, Courmayeur, Italy, 2012, 1–4.
[95] M. Grudén, A. Westman, J. Platbardis, P. Hallbjorner, and A. Rydberg, Reliability experiments for WSNs in train environment, in Proc. Eur. Wireless Technol. Conf., 2009, 37–40.
[96] J. Rabatel, S. Bringay, and P. Poncelet, SO-MAD: Sensor mining for anomaly detection in railway data, *Advances in Data Mining: Applications and Theoretical Aspects, LNCS*, 5633, 191–205, 2009.
[97] J. Rabatel, S. Bringay, and P. Poncelet, Anomaly detection in monitoring sensor data for preventive maintenance, *Expert Systems With Applications*, 38(6), 7003–7015, Jun. 2011.
[98] J. Reason, H. Chen, R. Crepaldi, and S. Duri, Intelligent telemetry for freight trains, In: *Mobile Computing, Applications, Services*, vol. 35, Springer-Verlag, Berlin, Germany, 72–91, 2010.
[99] J. Reason and R. Crepaldi, Ambient intelligence for freight railroads, *IBM Journal of Research and Development*, 53(3), 1–14, May 2009.
[100] K. Tuck, Using the 32 samples First In First Out (FIFO) in the MMA8450Q, energy scale solutions by free scale, FreeScale Solutions, 2010 [Online]. Available: http://www.nxp.com/docs/en/application-note/AN3920.pdf
[101] S. Pagano, S. Peirani, and M. Valle, Indoor ranging and localisation algorithm based on received signal strength indicator using statistic parameters for WSNs, *IET Wireless Sensor Systems*, 5(5), 243–249, 2015.
[102] M. Tolani, A. Bajpai, S. Sharma, R. K. Singh, L. Wuttisittikulkij, and Kovintavewat, Energy efficient hybrid medium access control protocol for WSN in 36th International Technical Conference on Circuits/Systems, Computers and Communications, (ITC-CSCC 21), at Jeju, South Korea, 28–30 June 2021.
[103] M. Tolani, Sunny, and R. K. Singh, Energy-efficient adaptive GTS allocation algorithm for IEEE 802.15.4 MAC protocol, In: *Telecommunication Systems*, Springer, 2020, doi: https://doi.org/10.1007/s11235-020-00719-0.
[104] M. Tolani, Sunny, and R. K. Singh, Adaptive duty cycle enabled energy-efficient bit-map-assisted MAC protocol, *Springer, SN Computer Science*, doi: 10.1007/s42979-020-00162-7.
[105] M. Tolani, Sunny, and R. K. Singh, Energy-efficient hybrid MAC protocol for railway monitoring sensor network, *Springer, SN Applied Sciences*, 2, 1404, 2020, doi: https://doi.org/10.1007/s42452-020-3194-1.
[106] M. Tolani, Sunny, and R. K. Singh, Energy-efficient aggregation-aware IEEE 802.15.4 MAC protocol for railway, tele-medicine & industrial applications, 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Gorakhpur, 2018, 1–5.
[107] A. A. Khan, M. S. Jamal, and S. Siddiqui, Dynamic duty-cycle control for WSNs using Artificial Neural Network (ANN), 2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2017, 420–424, doi: 10.1109/CyberC.2017.93.
[108] I. D. Wahyono, K. Asfani, M. M. Mohamad, H. Rosyid, A. Afandi, and Aripriharta, The new intelligent WSN using artificial intelligence for building fire disasters, 2020 Third International Conference on Vocational Education and Electrical Engineering (ICVEE), 2020, 1–6, doi: 10.1109/ICVEE50212.2020.9243210.

[109] F. Aliyu, S. Umar, and H. Al-Duwaish, A survey of applications of artificial neural networks in WSNs, 2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO), 2019, 1–5, doi: 10.1109/ICMSAO.2019.8880364.

[110] L. Sun, W. Cai, and X. Huang, Data aggregation scheme using neural networks in WSNs, 2010 2nd International Conference on Future Computer and Communication, vol. 1, V1–725-V1-729, May 2010.

Paawan Sharma, Vinay Vakharia, Debabrata Swain

Analysis of acoustic emission for milling operation using artificial neural networks

Abstract: Every natural or man-made signal is generated by some specific sources. Such sources may be deterministic or stochastic in nature. Depiction of deterministic signals is very easy as it requires only the exact mathematical system representation. However, stochastic or random process generated 1D or 2D signals that require extensive mathematical investigation for the modeling purpose. Hence, random signals can be represented with specific signature provided relevant signal transformations, and appropriate tests are performed. Techniques are widely used for system modeling in many applications. This chapter reports the usage of artificial neural networks for standard signal classification representing mechanical operation in milling. The developed model is used to analyze and establish correlation between acoustic emission and other aspects of the milling setup such as current flow and vibration data.

Keywords: signal analysis, artificial neural network, signal classification, milling

1 Introduction

The nature of real life signals from various domains is predominantly stochastic or random in nature. This is due to the natural phenomenon and its governance by various physical, chemical, or system boundations. Modeling of such systems is very challenging in comparison to systems that generate signals in a fashion. This is quite obvious, owing to the knowledge of mathematical relationship between input and output of the system. However, even a random signal generator can be modeled, provided it is a stationary system. Nonstationary signals are those that changes with respect to time. This makes modeling of random signals almost impossible. Such signals can be modeled only with proper mathematical transformation, which eventually can enable transition from nonstationary to stationary signal domain. Then, stationary

Paawan Sharma, Department of ICT, Pandit Deendayal Energy University, Gandhinagar, e-mail: paawan.sharma@sot.pdpu.ac.in
Vinay Vakharia, Department of Mechanical Engineering, Pandit Deendayal Energy University, Gandhinagar, e-mail: vinay.vakharia@sot.pdpu.ac.in
Debabrata Swain, Department of CSE, Pandit Deendayal Energy University, Gandhinagar, e-mail: Debabrata.Swain@sot.pdpu.ac.in

https://doi.org/10.1515/9783110734652-010

signals can be modeled with known probability density functions or their combination such as uniform, Poisson, and normal density functions.

Stationary random signals and processes are the ones in which the joint probability density function does not vary with time [3]. The properties of deterministic signal does not vary with respect to one or more of the independent variables [7], or more specifically, its variants are the well-known techniques for deterministic signal analysis [14]. On the other hand, analysis of nonstationary or stationary random signals may require one or more signal transformations as preprocessing steps [17]. Several works have reported the use of highly specific techniques for nonstationary signal analysis. Zheng and Pan [18] have reported an improved form of empirical mode decomposition for nonstationary signal processing. Boashash et al. [4] reported the use of time–frequency image feature sets for the analysis of nonstationary signals. Yue Hu et al. [10] discussed the use of method for signal decomposition by using an adaptive filter bank. Omer et al. [12] proposed the use of time warping of signals for estimation algorithms of nonstationary sound signal analysis. Work done by Jose et al. [9] presents an intuitive application of nonstationary rotating machinery surveillance detecting rolling element bearing defects. A detailed comparison of techniques for separating random and deterministic signal is reported in the work done by Randall et al. [13].

There are many research works done in recent times, highlighting the use of artificial neural networks (ANN) for the purpose of signal classifications. Thamba et al. [15] performed the application of, and for, self-aligning bearing fault diagnosis. Diker et al. [5] reported the classification of electrocardiogram signal by using machine learning methods.

2 Signal dataset selection

Datasets from the Center of Excellence, NASA, are very popular in research community with regard to their suitability for signal analysis in development of prognostic algorithms. Data generated from experiments on a milling machine for different speeds, feeds, and depth of cut are considered for this study [2]. The data report the wear of the milling insert, VB. This dataset reflects observations for different runs on a milling machine under various operational settings. Data pertaining to three different types of sensors, namely, acoustic emission, vibration, and current measurement, were reported to be acquired at several positions. Research studies such as [6, 8] report the use of this dataset.

The data are arranged in a 1×167 MATLAB struct array [2] with fields as case number, run counter, flank wear, feed, experiment time duration, depth of cut, material type, AC/DC spindle motor current (smcAC/smcDC), table and spindle vibration data (vib_table/vib_spindle), and table and spindle emission data (ae_table/ae_spindle).

There are two values for depth of cut, that is, 0.75 and 1.5 mm. Similarly, feed has two values: 0.25 and 0.50. For material types, cast iron and steel are the two classes under consideration. For each of the 167 test combinations, time-series data of 9,000 data points for smcAC, vib_table, smcDC, ae_table, vib_spindle, and ae_spindle are available. A strong emphasis is laid on tool wear control during this dataset experiment in the form of flank wear.

3 Proposed modeling

This study aims to derive relationship between time-series data for smcAC, vib_table, smcDC, ae_table, vib_spindle, and ae_spindle for different values of constant parameter values, using ANN. Figures 1–6 show plots of time-series data with specific parameter values for run number 6. For the analysis, run numbers 6, 37, 112, and 158 have been considered. The parameter values for specific run numbers are shown in Table 1. The proposed model for the analysis is shown in Figure 7. As a technically intuitive understanding, it is aimed to study the acoustic emission data (dependent variable) behavior with respect to independent variables in the form of smcAC/smcDC and spindle/table vibration data. Rapidminer studio [11] is used to perform deep learning modeling.

Figure 8 shows a step-by-step process execution with the help of a flowchart. The data is first imported in the computed platform and then passed through a series of steps including preprocessing and set creation till process deployment. Figure 9 shows a typical schematic of an ANN with input, hidden, and output layers. It is on the basis of number of layers and neurons that a deep neural network is defined with large number of hidden layers and neurons in action.

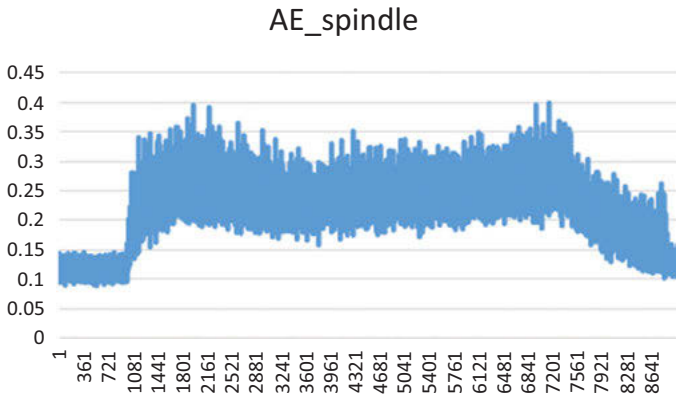


Figure 1: Spindle motor acoustic emission variation (run 6).

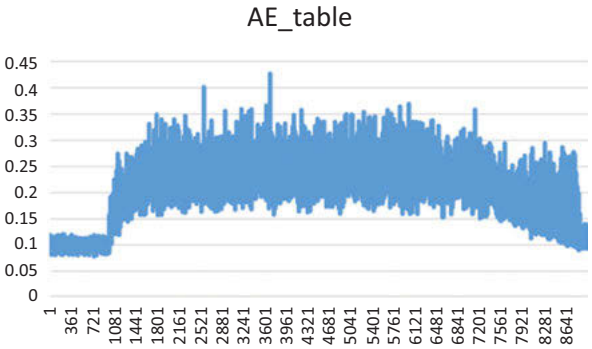


Figure 2: Table acoustic emission variation (run 6).

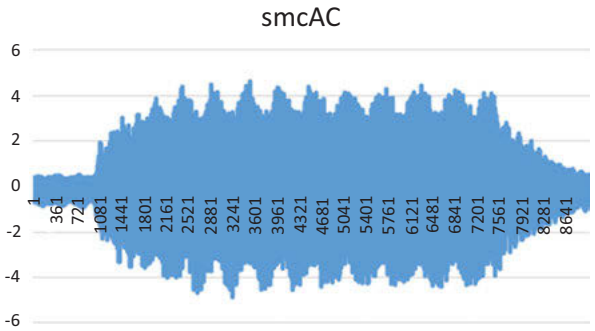


Figure 3: Spindle motor AC current (run 6).

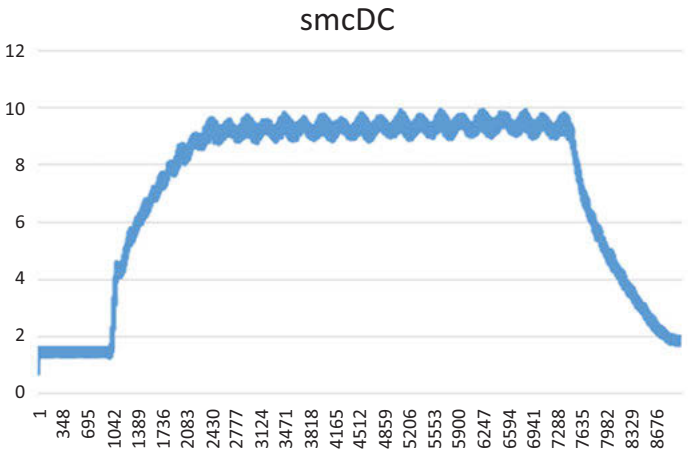


Figure 4: Spindle motor DC current (run 6).

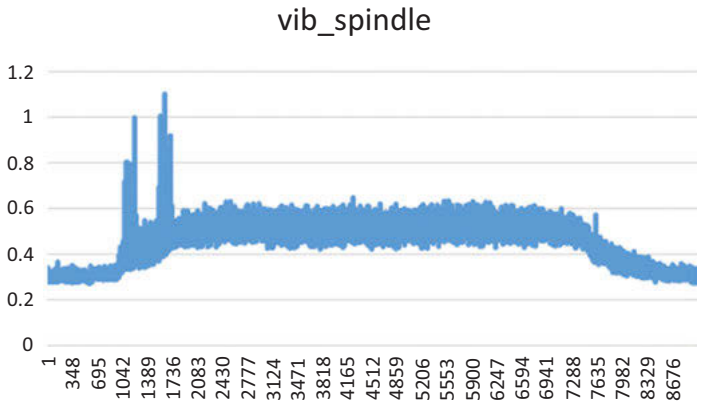


Figure 5: Spindle vibration (run 6).

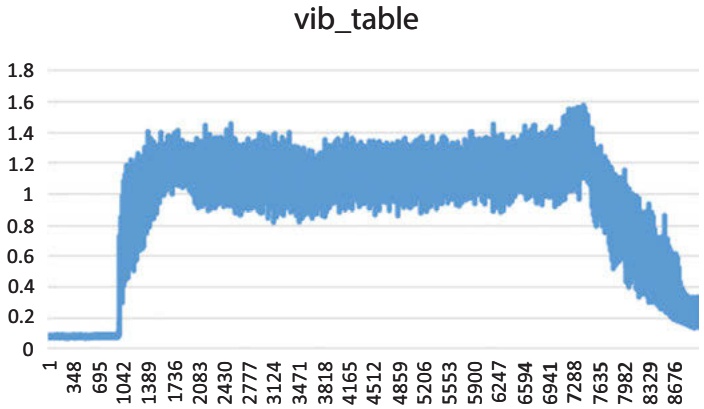


Figure 6: Table vibration (run 6).

Table 1: Run-wise parameter values.

Run	DOC	VB	Case	Feed	Material
6	1.5	0.2	1	0.5	1
37	0.75	0.2	3	0.25	1
112	1.5	0.29	5	0.5	2
158	1.5	0.37	15	0.25	2

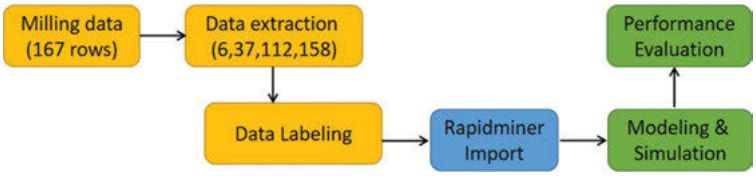


Figure 7: Proposed model.

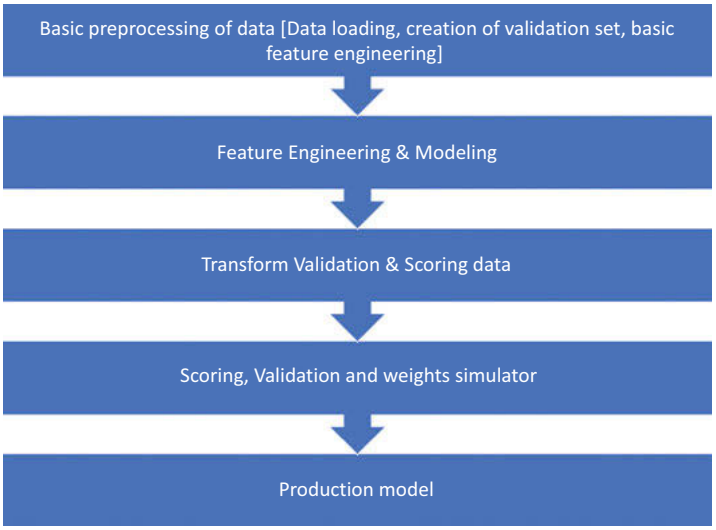


Figure 8: Process flowchart.

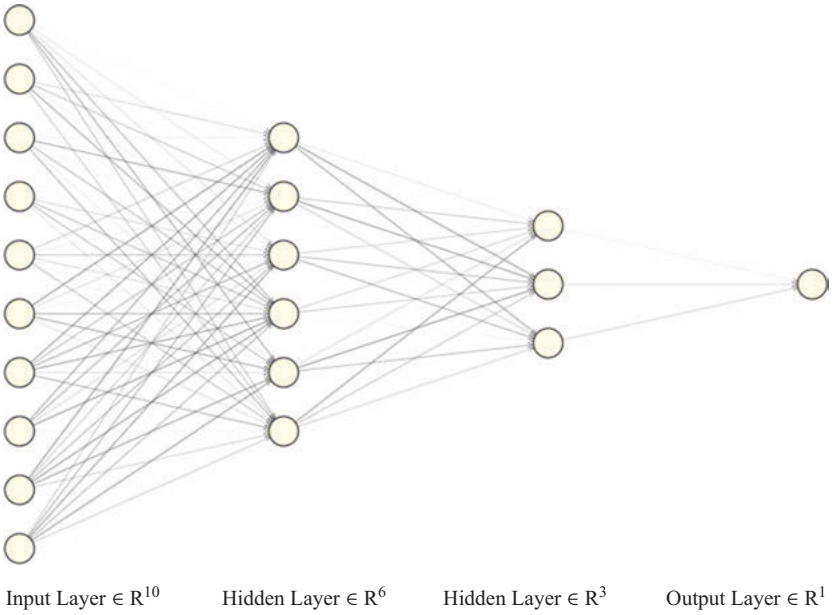


Figure 9: A typical artificial neural network schematic.

4 Results and discussion

Deep learning model with specifications as shown in Table 2 is simulated in Rapidminer studio for run number 6. For the same set of specifications, Table 3 lists performance metrics, while the prediction chart is shown in Figure 10.

Table 2: Model specifications (run 6).

Model Metrics Type: Regression												
Status of Neuron Layers (predicting AE_spindle, regression, gaussian distribution, Quadratic loss, 2,901 weights/biases, 38.9 KB, 54,000 training samples, mini-batch size 1):												
Layer	Units	Type	Dropout	L1	L2	Mean Rate	Rate RMS	Momentum	Mean Weight	Weight RMS	Mean Bias	Bias RMS
1	5	Input	0.00 %									
2	50	Rectifier	0	0.000010	0.000000	0.009114	0.006291	0.000000	0.001219	0.203138	0.266989	0.129315
3	50	Rectifier	0	0.000010	0.000000	0.110051	0.190999	0.000000	-0.036522	0.139264	0.890201	0.075469
4	1	Linear		0.000010	0.000000	0.002121	0.003113	0.000000	0.011203	0.136102	0.005060	0.000000

Table 3: Performance metrics (run 6).

MSE: 5.533738E-4
RMSE: 0.023523899
R^2: 0.8269624
mean residual deviance: 5.533738E-4
mean absolute error: 0.017758703
root mean squared log error: 0.018828485

Similar observation can be made in Tables 4–9 and Figures 11–14 for run numbers 37, 112, and 158, respectively.

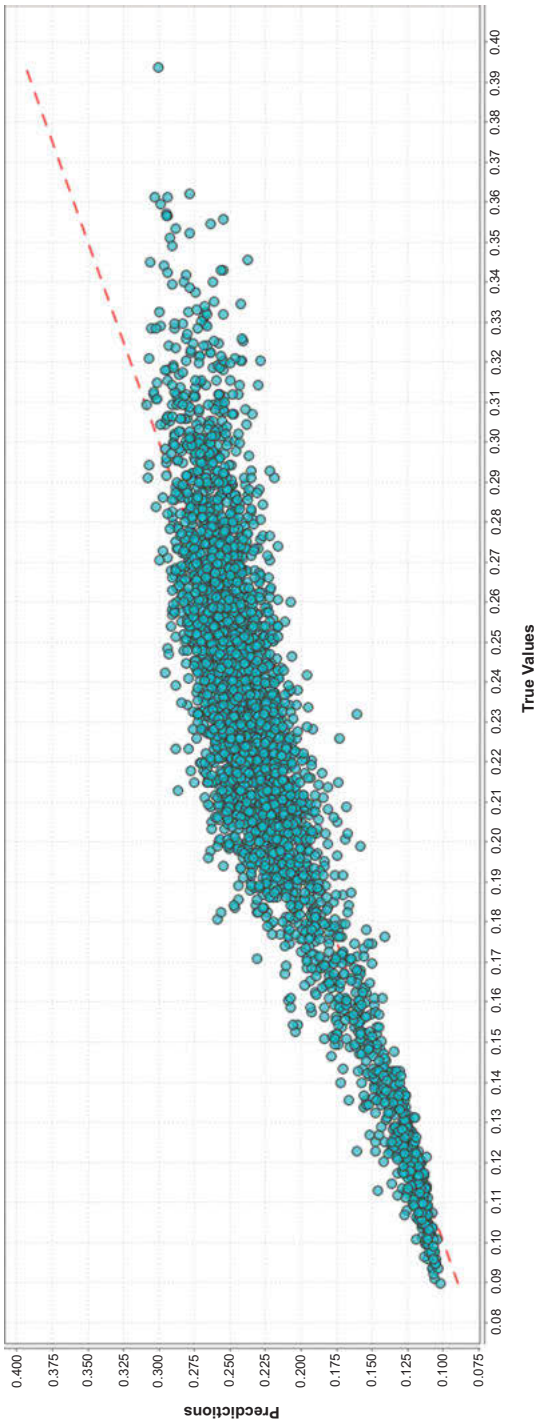


Figure 10: Prediction chart for the run 6 model.

Table 4: Model specifications (run 37).

Model Metrics Type: Regression
Status of Neuron Layers (predicting AE_spindle, regression, gaussian distribution, Quadratic loss, 2,901 weights/biases, 38.9 KB, 54,000 training samples, mini-batch size 1):

Layer	Units	Type	Dropout	L1	L2	Mean Rate	Rate RMS	Momentum	Mean Weight	Weight RMS	Mean Bias	Bias RMS
1	5	Input	0.00 %									
2	50	Rectifier	0	0.000010	0.000000	0.006369	0.004323	0.000000	0.005426	0.192274	0.330101	0.111667
3	50	Rectifier	0	0.000010	0.000000	0.089348	0.167428	0.000000	-0.029170	0.137201	0.925028	0.060124
4	1	Linear		0.000010	0.000000	0.001532	0.001501	0.000000	0.020191	0.145077	0.003962	0.000000

Table 5: Performance metrics (run 37).

MSE: 2.3402275E-4
RMSE: 0.015297802
R^2: 0.8092498
mean residual deviance: 2.3402275E-4
mean absolute error: 0.01168192
root mean squared log error: 0.012534039

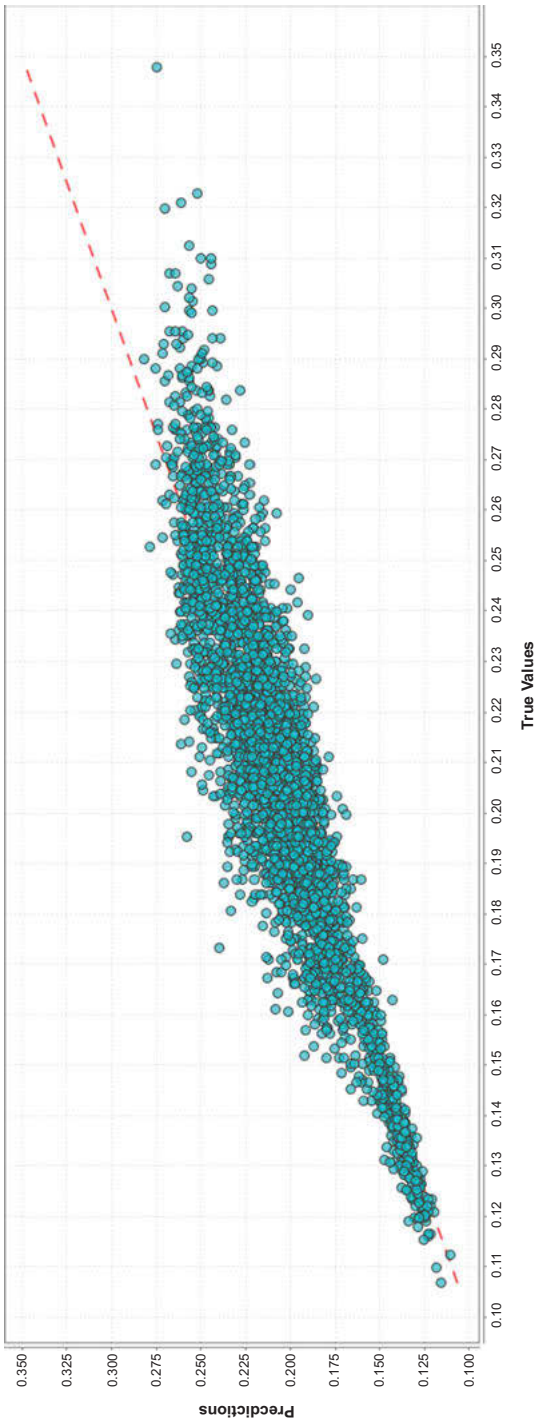


Figure 11: Prediction chart for the run 37 model.

Table 6: Model specifications (run 112).

Model Metrics Type: Regression
Status of Neuron Layers (predicting AE_spindle, regression, gaussian distribution, Quadratic loss, 2,901 weights/biases, 38.9 KB, 54,000 training samples, mini-batch size 1):

Layer	Units	Type	Dropout	L1	L2	Mean Rate	Rate RMS	Momentum	Mean Weight	Weight RMS	Mean Bias	Bias RMS
1	5	Input	0.00 %									
2	50	Rectifier	0	0.000010	0.000000	0.002286	0.002050	0.000000	0.009600	0.186523	0.436506	0.070012
3	50	Rectifier	0	0.000010	0.000000	0.061026	0.131527	0.000000	-0.024580	0.160644	0.957605	0.054035
4	1	Linear		0.000010	0.000000	0.001349	0.001819	0.000000	0.019308	0.186799	-0.017025	0.000000

Table 7: Performance metrics (run 112).

MSE: 2.1727191E-4
RMSE: 0.014740147
R^2: 0.84452564
mean residual deviance: 2.1727191E-4
mean absolute error: 0.01059435
root mean squared log error: 0.012115717

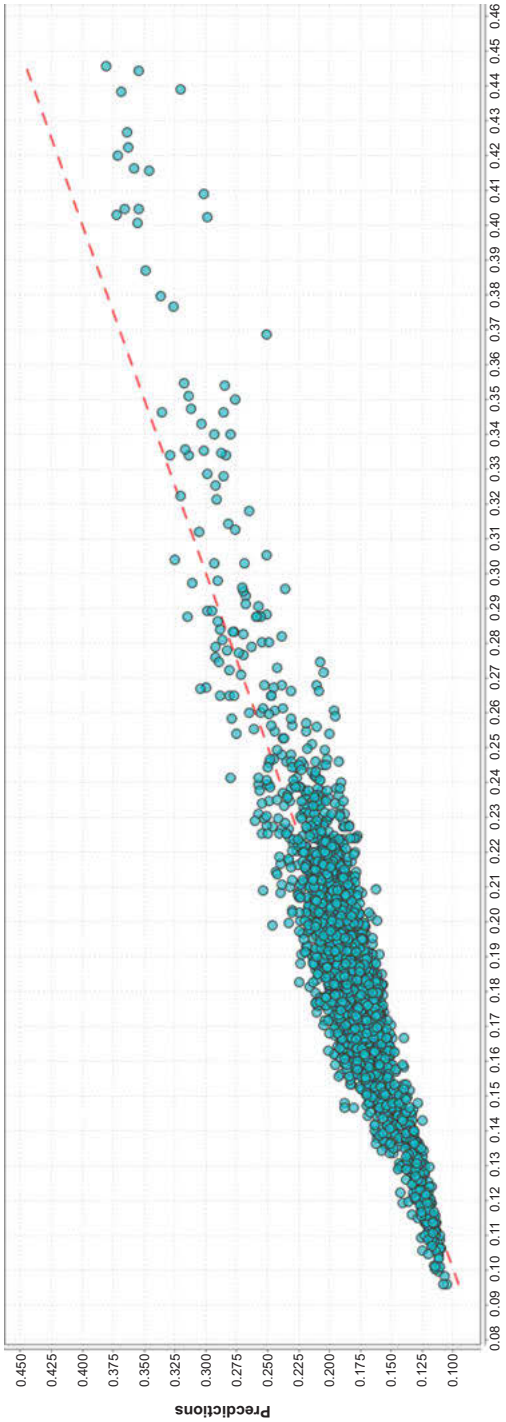


Figure 12: Prediction chart for the run 112 model.

Table 8: Model specifications (run 158).

Model Metrics Type: Regression

Status of Neuron Layers (predicting AE_spindle, regression, gaussian distribution, Quadratic loss, 2,901 weights/biases, 38.9 KB, 54,000 training samples, mini-batch size 1):

Layer	Units	Type	Dropout	L1	L2	Mean Rate	Rate RMS	Momentum	Mean Weight	Weight RMS	Mean Bias	Bias RMS
1	5	Input	0.00 %									
2	50	Rectifier	0	0.000010	0.000000	0.003617	0.003193	0.000000	0.004926	0.199697	0.436855	0.063379
3	50	Rectifier	0	0.000010	0.000000	0.056401	0.122792	0.000000	-0.028188	0.155225	0.958505	0.046880
4	1	Linear		0.000010	0.000000	0.001246	0.001146	0.000000	0.030186	0.174334	0.012852	0.000000

Table 9: Performance metrics (run 158).

MSE: 7.27797E-4
RMSE: 0.026977714
R^2: 0.91674465
mean residual deviance: 7.27797E-4
mean absolute error: 0.016874423
root mean squared log error: 0.019999502

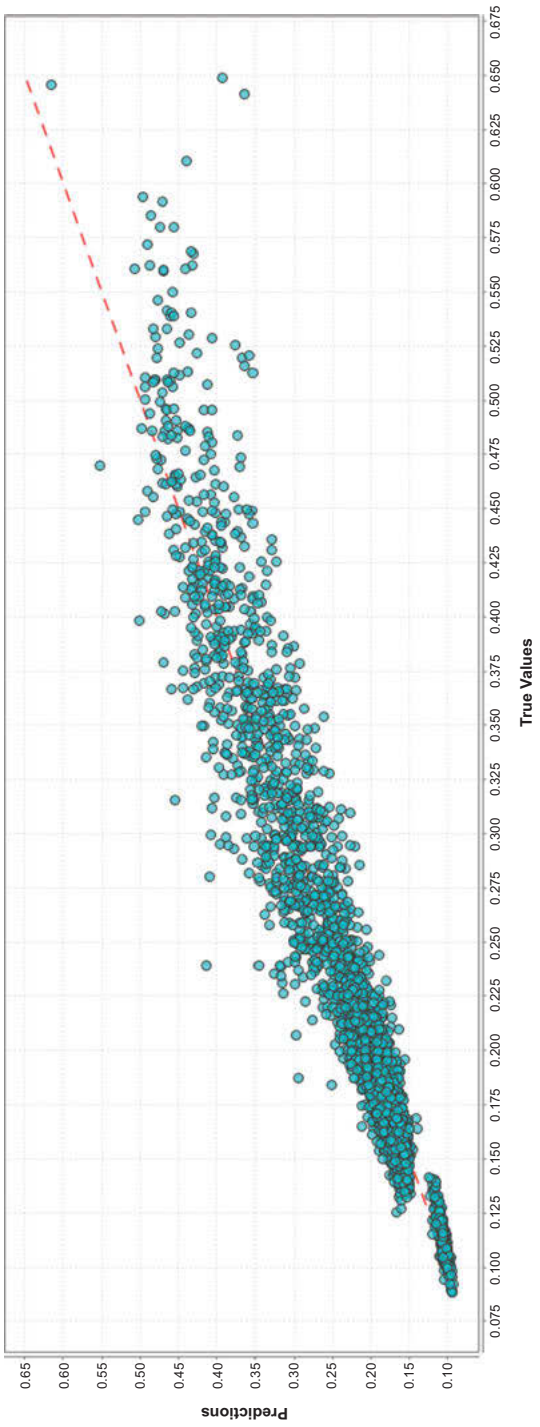


Figure 13: Prediction chart for the run 158 model.

Attributes	AE_spindle	AE_table	smaAC	smaDC	vib_spi...	vib_table
AE_spindle	1	0.922	–0.020	–0.017	0.123	0.694
AE_table	0.922	1	–0.066	0.084	0.154	0.752
smcAC	–0.020	–0.066	1	0.006	–0.002	0.000
smcDC	–0.0017	0.084	0.006	1	–0.244	0.299
vib_spindle	0.123	0.154	–0.002	–0.244	1	0.188
vib_table	0.694	0.752	0.000	0.299	0.188	1

Figure 14: Correlation of variables.

5 Conclusion

The simulation results show a very good performance by ANN. The R^2 value varies between 0.80 and 0.91. These results are fairly comparable with the similar analysis in milling dataset by Traini et al. [16]. Correlation matrix (run 158) as shown in Figure 9 reflects the fact that acoustic emission for spindle motor has a strong relationship with acoustic emission of table, while at the same time it has got least correlation with AC/DC current. Hence, with the help of such analysis, important information can be extracted using neural network techniques. Further, models can be optimized with best hyperparameter settings.

References

[1] Prognostics center of excellence – data repository. URL https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/.

[2] Agogino and K. Goebel. Milling data set, 2007. URL https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/.

[3] S. M. Alessio. *Digital Signal Processing and Spectral Analysis for Scientists: Concepts and Applications*, Springer International Publishing, Cham, Switzerland, 2015.

[4] B. Boashash, H. Barki, and S. Ouelha. Performance evaluation of time-frequency image feature sets for improved classification and analysis of non-stationary signals: Application to newborn EEG seizure detection, *Knowledge-Based Systems*, 132, 188–203, 2017.

[5] A. Dker, E. Avci, Z. Cömert, D. Avci, E. Kaçar, and H. Serhatliolu. Classification of ECG signal by using machine learning methods. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE, 2018.

[6] O. F. Eker, F. Camci, and I. K. Jennions. Major challenges in prognostics: Study on benchmarking prognostics datasets. In *PHM Society European Conference*, volume 1, 2012.

[7] A. Esmaili *Probability and Random Processes*, Technometrics, 47:3, 375, DOI: 10.1198/tech.2005.s294, 2005.
[8] K. F. Goebel. *Management of Uncertainty in Sensor Validation, Sensor Fusion, and Diagnosis of Mechanical Systems Using Soft Computing Techniques*, University of California, Berkeley, 1996.
[9] J. L. Gomez, I. Khelf, A. Bourdon, H. André, and R. Didier. Angular modeling of a rotating machine in non-stationary conditions: Application to monitoring bearing defects of wind turbines with instantaneous angular speed, *Mechanism and Machine Theory*, 136, 27–51, 2019.
[10] H. Yue, L. Fucai, L. Hongguang, and C. Liu. An enhanced empirical wavelet transform for noisy and non-stationary signal processing, *Digital Signal Processing*, 60, 220–229, 2017.
[11] I. Mierswa and R. Klinkenberg. Rapidminer studio (9.2) [data science, machine learning, predictive analytics], 2018.
[12] H. Omer and T. Bruno. Time-frequency and time-scale analysis of deformed stationary processes, with application to non-stationary sound modeling, *Applied and Computational Harmonic Analysis*, 43(1), 1–22, 2017.
[13] R. B. Randall, N. Sawalhi, and M. Coats A comparison of methods for separation of deterministic and random signals, *International Journal of Condition Monitoring*, 1(1), 11–19, 2011.
[14] S. Rapuano and F. J. Harris An introduction to FFT and time domain windows, *IEEE Instrumentation & Measurement Magazine*, 10(6), 32–44, 2007.
[15] N. B. Thamba, A. Aravind, A. Rakesh, M. Jahzan, et al. Application of EMD, ANN and DNN for self-aligning bearing fault diagnosis, *Archives of Acoustics*, 43(2), 163–175, 2018.
[16] E. Traini, G. Bruno, G. D'antonio, and F. Lombardi Machine learning framework for predictive maintenance in milling, *IFAC-PapersOnLine*, 52(13), 177–182, 2019.
[17] P. Walden. *Nonlinear and Nonstationary Signal Processing*, Cambridge University Press, Cambridge, 2000.
[18] J. Zheng and H. Pan. Mean-optimized mode decomposition: An improved EMD approach for non-stationary signal processing, *ISA Transactions*, 106, 392–401, 2020.

Index

https://doi.org/10.1515/9783110734652-011

De Gruyter Series on the Applications of Mathematics in Engineering and Information Sciences

Already published in the series

Volume 10: Meta-heuristic Optimization Techniques. Applications in Engineering
Mangey Ram, Anuj Kumar, Sangeeta Pant, Om Yadav (Eds.)
ISBN 978-3-11-071617-7, e-ISBN (PDF) 978-3-11-071621-4
e-ISBN (EPUB) 978-3-11-071625-2

Volume 9: Linear Integer Programming. Theory, Applications, Recent Developments
Elias Munapo, Santosh Kumar
ISBN 978-3-11-070292-7, e-ISBN (PDF) 978-3-11-070302-3
e-ISBN (EPUB) 978-3-11-070311-5

Volume 8: Mathematics for Reliability Engineering. Modern Concepts and Applications
Mangey Ram, Liudong Xing (Eds.)
ISBN 978-3-11-072556-8, e-ISBN (PDF) 978-3-11-072563-6
e-ISBN (EPUB) 978-3-11-072559-9

Volume 7: Mathematical Fluid Mechanics. Advances on Convection Instabilities and Incompressible Fluid Flow
B. Mahanthesh (Ed.)
ISBN 978-3-11-069603-5, e-ISBN (PDF) 978-3-11-069608-0
e-ISBN (EPUB) 978-3-11-069612-7

Volume 6: Distributed Denial of Service Attacks. Concepts, Mathematical and Cryptographic Solutions
Rajeev Singh, Mangey Ram (Eds.)
ISBN 978-3-11-061675-0, e-ISBN (PDF) 978-3-11-061975-1
e-ISBN (EPUB) 978-3-11-061985-0

Volume 5: Systems Reliability Engineering. Modeling and Performance Improvement
Amit Kumar, Mangey Ram (Eds.)
ISBN 978-3-11-060454-2, e-ISBN (PDF) 978-3-11-061737-5
e-ISBN (EPUB) 978-3-11-061754-2

Volume 4: Systems Performance Modeling
Adarsh Anand, Mangey Ram (Eds.)
ISBN 978-3-11-060450-4, e-ISBN (PDF) 978-3-11-061905-8
e-ISBN (EPUB) 978-3-11-060763-5

www.degruyter.com

Volume 3: Computational Intelligence. Theoretical Advances and Advanced Applications
Dinesh C. S. Bisht, Mangey Ram (Eds.)
ISBN 978-3-11-065524-7, e-ISBN (PDF) 978-3-11-067135-3
e-ISBN (EPUB) 978-3-11-066833-9

Volume 2: Supply Chain Sustainability. Modeling and Innovative Research Frameworks
Sachin Kumar Mangla, Mangey Ram (Eds.)
ISBN 978-3-11-062556-1, e-ISBN (PDF) 978-3-11-062859-3,
e-ISBN (EPUB) 978-3-11-062568-4

Volume 1: Soft Computing. Techniques in Engineering Sciences
Mangey Ram, Suraj B. Singh (Eds.)
ISBN 978-3-11-062560-8, e-ISBN (PDF) 978-3-11-062861-6,
e-ISBN (EPUB) 978-3-11-062571-4