Machine Learning and Deep Learning Techniques in Wireless and Mobile Networking Systems
Industry 4.0 or fourth industrial revolution refers to interconnectivity, automation and real time data exchange between machines and processes. There is a tremendous growth in big data from internet of things (IoT) and information services which drives the industry to develop new models and distributed tools to handle big data. Cutting-edge digital technologies are being harnessed to optimize and automate production including upstream supply-chain processes, warehouse management systems, automated guided vehicles, drones etc. The ultimate goal of industry 4.0 is to drive manufacturing or services in a progressive way to be faster, effective and efficient that can only be achieved by embedding modern day technology in machines, components, and parts that will transmit real-time data to networked IT systems. These, in turn, apply advanced soft computing paradigms such as machine learning algorithms to run the process automatically without any manual operations.

The new book series will provide readers with an overview of the state-of-the-art in the field of Industry 4.0 and related research advancements. The respective books will identify and discuss new dimensions of both risk factors and success factors, along with performance metrics that can be employed in future research work. The series will also discuss a number of real-time issues, problems and applications with corresponding solutions and suggestions. Sharing new theoretical findings, tools and techniques for Industry 4.0, and covering both theoretical and application-oriented approaches. The book series will offer a valuable asset for newcomers to the field and practicing professionals alike. The focus is to collate the recent advances in the field, so that undergraduate and postgraduate students, researchers, academicians, and Industry people can easily understand the implications and applications of the field.

**Industry 4.0 Interoperability, Analytics, Security, and Case Studies**  
*Edited by G. Rajesh, X. Mercilin Raajini, and Hien Dang*

**Big Data and Artificial Intelligence for Healthcare Applications**  
*Edited by Ankur Saxena, Nicolas Brault, and Shazia Rashid*

**Machine Learning and Deep Learning Techniques in Wireless and Mobile Networking Systems**  
*Edited by K. Suganthi, R. Karthik, G. Rajesh, and Peter Ho Chiung Ching*

For more information on this series, please visit: www.routledge.com/Big-Data-for-Industry-4.0-Challenges-and-Applications/book-series/CRCBDICA
Machine Learning and Deep Learning Techniques in Wireless and Mobile Networking Systems

Edited by
K. Suganthi, R. Karthik, G. Rajesh, and Peter Ho Chiung Ching
Contents

Preface ........................................................................................................................................ vii
Editors .......................................................................................................................................... ix

Chapter 1  Overview of Machine Learning and Deep Learning Approaches ................................................................. 1

Annie Johnson, Sundar Anand, R. Karthik, and Ganesh Subramanian

Chapter 2  ML and DL Approaches for Intelligent Wireless Sensor Networks ................................................................. 11

Bhanu Chander

Chapter 3  Machine Learning-Based Optimal Wi-Fi HaLow Standard for Dense IoT Networks ....................................... 41

M. Mahesh and V.P. Harigovindan

Chapter 4  Energy Efficiency Optimization in Clustered Wireless Sensor Networks via Machine Learning Algorithms ........ 59

T. Sudarson Rama Perumal, V. Muthumanikandan, and S. Mohanalakshmi

Chapter 5  Machine Learning Approaches in Big Data Analytics Optimization for Wireless Sensor Networks ................ 79

G. Sabarmathi and R. Chinnaiyan

Chapter 6  Improved Video Steganography for Secured Communication Using Clustering and Chaotic Mapping .......... 97

Lakshmi Harika Palivela, M.R. Sumalatha, A. Naveen, S. Maithreyan, and S.V. Vijay

Chapter 7  Target Prophecy in an Underwater Environment Using a KNN Algorithm ...................................................... 117

Menaka D, Sabitha Gauni, C.T. Manimegalai, K. Kalimuthu, and Bhuvan Unhelkar
Chapter 8  A Model for Evaluating Trustworthiness Using Behaviour and Recommendation in Cloud Computing Integrated with Wireless Sensor Networks.................................................. 131

Lokesh B. Bhajantri and Tabassum N. Mujawar

Chapter 9  Design of Wireless Sensor Networks Using Fog Computing for the Optimal Provisioning of Analytics as a Service.................................................................................... 153

Rajalakshmi Shenbaga Moorthy and P. Pabitha

Chapter 10  DLA-RL: Distributed Link Aware-Reinforcement Learning Algorithm for Delay-Sensitive Networks ................................................................. 175

O.S. Gnana Prakasi, P. Kanmani, K. Suganthi, G. Rajesh, and T. Samraj Lawrence

Chapter 11  Deep Learning-Based Modulation Detector for an MIMO System........................................................................................................... 189

Arumbu Vanmathi Neduncheran, Malarvizhi Subramani, Ezra Morris, Abraham Gnanamuthu, Siddhartha Dhar Choudhury, and Sangeetha Manoharan

Chapter 12  Deep Learning with an LSTM-Based Defence Mechanism for DDoS Attacks in WSNs........................................................................ 201

P. Manju Bala, S. Usharani, T. Ananth Kumar, R. Rajmohan, and P. Praveen Kumar

Chapter 13  A Knowledge Investigation Framework for Crowdsourcing Analysis for e-Commerce Networks.................................................. 225

Harsh Jigneshkumar Patel, Dipanshi Digga, and Jai Prakash Verma

Chapter 14  Intelligent Stackelberg Game Theory with Threshold-Based VM Allocation Strategy for Detecting Malicious Co-Resident Virtual Nodes in Cloud Computing Networks......... 249

Shawakat Akbar Almamun, E. Balamurugan, Shahidul Hasan, N.M. Saravana Kumar, and K. Sangeetha

Index ...................................................................................................................... 269
Preface

The design and development of automated approaches to improve the performance of wireless networks are considered among the challenging research issues in the field of wireless and mobile networking. The application of artificial intelligence (AI), machine learning (ML), and deep learning (DL) is relatively limited in the field of wireless networking systems and needs new models and methods to be developed to improve performance. Wireless network technologies such as the Internet of Things (IoT), Industry 4.0, Industrial Internet of Things (IIoT), VANET, and FANET-based applications demand data-driven approaches which involve complex mathematical models. These models can be automated and optimized using ML and DL techniques. AI-, ML-, and DL-based schemes are more adaptable to the wireless environment. These models provide an optimized way to reduce the complexity and overhead of the traditional tractable system models.

The large amount of data produced by wireless networks need to be stored and processed quickly to support real-time applications. This necessitates the attraction of data-driven approaches such as AI-, ML-, and DL-based schemes toward wireless communication and networking. Compared to traditional technologies, new technologies such as cyber-physical systems, cloud computing, virtualization, FANET, and VANET will have diverse service requirements and complicated system models that are harder to manage with conventional approaches properly. To cater to these needs, ML- and DL-based techniques can be employed in this domain to achieve automation. At present, automated learning algorithms in mobile wireless systems are in a growing phase, and the performance of these models needs to be optimized. This book aims to cover the state-of-the-art approaches in AI, ML, and DL for building intelligence in wireless and mobile networking systems.

It provides the latest advancements in the field of machine learning for wireless communications, encourages fruitful development on the challenges and prospects of this new research field. It provides a broad spectrum to understand the improvements in ML/DL that are motivated by the specific constraints posed by wireless communications.
Editors

K. Suganthi received her BE in computer science and engineering from Madras University, her master’s in systems engineering and operations research, and her PhD in wireless sensor networks from the Anna University. She has been an assistant professor senior in the School of Electronics Engineering (SENSE) at Vellore Institute of Technology, Chennai campus, India, since 2016. She is the author of about 20 scientific publications in journals and international conferences. Her research interests include wireless sensor networks, the Internet of Things, data analytics, and artificial intelligence.

R. Karthik obtained his doctoral degree from Vellore Institute of Technology, India and his master’s degree from Anna University, India. Currently, he serves as a senior assistant professor in the Research Center for Cyber Physical Systems, Vellore Institute of Technology, Chennai. His research interest includes deep learning, computer vision, digital image processing, and medical image analysis. He has published about 32 papers in peer-reviewed journals and conferences. He is an active reviewer for journals published by Elsevier, IEEE, Springer, and Nature.

G. Rajesh is an assistant professor in the Department of Information Technology of Anna University, Chennai, India. He completed his PhD from Anna University in wireless sensor networks and has approximately 12 years of teaching and research experience. His area of research interest includes wireless sensor networks and its Internet of Things applications, software engineering, and computational optimization. He has published more than 20 research papers in journals and conferences.

Peter Ho Chiung Ching received his PhD in information technology from the Faculty of Computing and Informatics, Multimedia University. His doctoral research work explored the performance evaluation of multimodal biometric systems using fusion techniques. Dr. Ho is a senior member of the Institute of Electrical and Electronics Engineers. Dr. Ho has published a number of peer-reviewed papers related to location intelligence, multimodal biometrics, action recognition, and text mining. He is currently an adjunct senior research fellow in the Department of Computing and Information Systems, School of Science and Technology, Sunway University.
Overview of Machine Learning and Deep Learning Approaches

Annie Johnson, Sundar Anand, R. Karthik, and Ganesh Subramanian

CONTENTS

1.1 Introduction .................................................................................................................. 1
1.2 ML ................................................................................................................................ 2
  1.2.1 Supervised Learning .............................................................................................. 2
     1.2.1.1 Linear Regression ......................................................................................... 3
     1.2.1.2 Logistic Regression ....................................................................................... 3
     1.2.1.3 KNNs ........................................................................................................... 4
     1.2.1.4 Naïve Bayes ................................................................................................. 4
     1.2.1.5 SVMs ............................................................................................................ 5
     1.2.1.6 Decision Trees and Random Forests ............................................................ 5
  1.2.2 Unsupervised Learning ........................................................................................... 5
     1.2.2.1 k-Means ....................................................................................................... 6
     1.2.2.2 HCA ............................................................................................................. 6
     1.2.2.3 PCA ............................................................................................................. 6
     1.2.2.4 Kernel PCA .................................................................................................. 7
     1.2.2.5 Apriori ......................................................................................................... 7
     1.2.2.6 Equivalence Class Transformation Algorithm (ECLAT) ....................... 7
  1.2.3 Semi-Supervised Learning ....................................................................................... 7
  1.2.4 Analysis of ML ....................................................................................................... 8
1.3 DL .................................................................................................................................. 8
  1.3.1 CNNs .................................................................................................................... 9
  1.3.2 RNNs .................................................................................................................. 9
  1.3.3 GANs ................................................................................................................ 9
  1.3.4 Analysis of DL ................................................................................................... 9
1.4 Conclusion ..................................................................................................................... 9
References .......................................................................................................................... 10

1.1 INTRODUCTION

In this age in which most platforms and services are automated, the network domain stands as no exception. Automation is applied to the various processes in the network life cycle that previously involved manual, time-consuming and unreliable

DOI: 10.1201/9781003107477-1
Machine Learning and Deep Learning

procedures. The network lifecycle consists of repeatable processes such as preparing, planning and designing, implementing, deploying, operating and optimizing the network. Traditional networks are slow and unresponsive as they are manually managed and hardware-centric. Therefore, to structure the networking systems better and intelligently control the cycle, software-defined networking (SDN) was introduced (1). The SDN approach enables a programmed and centrally controlled network that drastically improves the performance of the network.

It is beneficial to automate wireless networking systems as this would improve operational efficiency by reducing the number of network issues. As a result, the time involved in delivering solutions to those issues would also be minimal. Automation simplifies operations and makes the network cost-effective. These networks handle repetitive tasks with ease and are not susceptible to human errors. This establishes better control of the network and enables more innovations through the insights offered by network analytics. Automated networks are more resilient and experience lesser downtime. Hence, there has been a rise in the use of machine learning (ML) and deep learning (DL) techniques in network automation.

1.2 ML

ML is a subsection of artificial intelligence (AI) that equips computers to learn from data without having to explicitly program the learning algorithm. Developing an ML model capable of making accurate decisions consists of many stages beginning with the data collection phase. The data collected are usually split into two parts, namely, a training set that trains the ML model and a testing set used to determine the performance of the fully trained model. The data collected are then preprocessed in the data preparation stage. Then an appropriate algorithm to solve the problem at hand is determined. This is followed by the training phase during which the model identifies patterns and learns how to distinguish between the various input values provided. Once the model has been trained, it can be evaluated on a new set of data. These evaluation results are used to carry out parameter tuning and improve the performance of the model. Finally, the best network is used to make predictions. ML algorithms are useful as they can discover new patterns from massive amounts of data. These are several categories of ML algorithms that are classified based on multiple criteria.

Depending on whether or not these networks are trained with human supervision, ML algorithms are broadly classified into supervised learning, unsupervised learning and semi-supervised learning. The most widely used ML algorithms under each of these classes are discussed in the following sections.

1.2.1 Supervised Learning

ML models that utilize labeled data sets for training perform supervised learning. The algorithm relies on the output labels to form a relation between the input variable or the independent variable \((X)\) and the output variable or the dependent variable \((Y)\). The mapping function that denotes the relation between \(X\) and \(Y\) is represented
Overview of ML and DL Approaches

Supervised learning can be further classified into regression and classification problems based on the task performed by the algorithm.

Problems that involve the prediction of a numerical or continuous-valued output are known as regression problems. For example, if the price of a house is to be determined by leveraging features such as house plot area, number of floors, number of rooms and number of bathrooms, we would need input training data and the corresponding price labels. Using these data, a supervised learning model that predicts a numerical price value can be developed to solve this regression problem. Algorithms such as linear regression and logistic regression are popular regression-based ML algorithms that are used in supervised learning.

The second class of supervised learning problems is known as classification problems. Classification tasks involve mapping the test data to two or more categories. In these problems, the ML model is expected to provide only discrete output values that can be translated into one of the output classes. The most common type of problem that falls under this category is the image classification task. For instance, if images of cats and dogs had to be classified, then a supervised learning model for classification must be employed. Some well-known algorithms that fall under this category consist of k-nearest neighbors (KNNs), the Naïve Bayes model, Support Vector Machines (SVMs), decision trees and random forests.

1.2.1.1 Linear Regression

Linear regression is an algorithm that models the mapping function $f$ as a linear function. It assumes that there is a linear relationship between the input and output data:

$$y = x\beta + \varepsilon.$$  \hspace{1cm} (1.1)

In Equation 1.1, $x$ is the independent variable that represents the input variable and $y$ is the dependent variable that represents the output variable. The slope parameter, $\beta$, is termed as a regression coefficient, and $\varepsilon$ is the error in predicting the $y$ value. Here, $y$ is depicted as a function ($f$) of $x$. The test data are entered into this linear function to predict the output value. Fig. 1.1 shows the prediction of $y$ using a single input feature and the simple linear regression.

1.2.1.2 Logistic Regression

Input values are fed into the logistic regression model, which uses the logit function or the sigmoid function as shown in Equation 1.2 to produce output predictions that lie between 0 and 1. The logistic regression model can also be used to solve classification problems as the continuous-valued output values correspond to the probability of an instance being associated with a certain class:

$$P(y = \pm 1|x, \beta) = \sigma(y^T\beta) = \frac{1}{1 + e^{-y^T\beta}}.$$  \hspace{1cm} (1.2)

In Equation 1.2, $\beta = (\beta_0, \ldots, \beta_d)$ is a vector of dimension $d$, known as the model parameters, $y$ is the class label which is $\pm 1$ in the equation. The vector $x = (1, x_1, \ldots, x_d)$ are the covariates or input values (2).
1.2.1.3 KNNs
KNN is a classification model that labels the nearest patterns to a target pattern \( x \). The class label is assigned to points based on a similarity measure in data space. Equation 1.3 defines the KNN for a binary classification problem:

\[
f_{\text{KNN}}(x') = \begin{cases} 
1 & \text{if } \sum_{i} iN_{i}(x') \cdot y \geq 0 \\
-1 & \text{if } \sum_{i} iN_{i}(x') \cdot y < 0 
\end{cases}
\]  

(1.3)

Here, \( y \) can either take the value of 1 or -1. \( N_{i}(x') \) denotes the indices of \( K \) of the nearest patterns. \( K \) is the neighbourhood size. It is used to define the locality of the KNN. For smaller neighbourhood sizes (\( K \leq 2 \)), scattered patterns of different classes are obtained whereas, for larger neighbourhood sizes (\( K > 19 \)), minority groups are ignored (3).

1.2.1.4 Naïve Bayes
The Naïve Bayes algorithm performs best on large data sets. This algorithm functions on the assumption that the various features of the data set are independent of each other. Then the Naïve Bayes model finds the probability of a new test sample belonging to a certain class and uses this parameter to perform classification. The model predicts the probability that a new sample, \( x = (x_{1}, \ldots, x_{a}) \), belongs to some class \( y \), which is represented as \( P(y \mid x) \). Here, \( x_{i} \) is the value of the attribute \( X_{i} \), and \( y \in \{1, \ldots, c\} \) is the value of the output class \( Y \).
1.2.1.5 SVMs

SVMs use hyper-plane classifiers to separate the data points into their respective classes. The hyper-plane would be a point for or a one-dimensional data set, a line for a two-dimensional data set, a plane for a three-dimensional data set and a hyper-plane for any data set having a dimension higher than three. A linearly separable SVM classifier is denoted by Equation 1.4 (4):

\[ ax + by + c = 0. \]  

(1.4)

Here, \((x,y)\) are the data points. The slope of the linear classifier is given by \((a/b)\), and the intercept term is \((c/b)\).

1.2.1.6 Decision Trees and Random Forests

One of the most basic classifiers, a decision tree performs two tasks, learning and classification. Based on the training data set, the decision tree learns the split criterion. This phase is known as the learning phase. The phase that follows the training phase is the classification phase, during which the test data are classified using the trained tree. The tree has a structure resembling a flow chart and consists of three parts known as the leaf node, the branch and the internal nodes. Each branch of the tree represents the output obtained on a test condition. The bottommost node that holds the final predicted output class is called the leaf node. A decision tree is one of the simplest machine learning classifiers and is easy to comprehend. However, one of the challenges faced by the decision tree algorithm is that it is more likely to overfit the data. Therefore, it is a weak classifier. Hence, many decision trees are combined to form a stronger classifier known as a random forest. The random forest is an ensemble model that provides its final classification by choosing the most popular class predicted by the decision trees for a data sample as the final classification for that particular data sample (5).

1.2.2 Unsupervised Learning

Unsupervised learning involves the discovery of previously unknown patterns from unlabeled data. Unlike supervised algorithms, unsupervised algorithms can help address a wider range of problems as it is easier to obtain unlabeled data. Unsupervised ML algorithms can fall under three types, which include clustering algorithms, visualization and dimensionality reduction algorithms and association rule learning algorithms. This classification is based on the type of task performed by the algorithm.

Clustering algorithms find a structure in the uncategorized data. Similar data points are grouped together. For instance, segregating consumers into groups, with the help of clustering models, would help businesses target their customers better and get the best return on investment. k-means and hierarchical cluster analysis (HCA) are common clustering algorithms.

Visualization and dimensionality reduction algorithms perform related tasks. A visualization algorithm is used to model and plot unstructured data as a two- or three-dimensional representation. This helps in identifying unsuspected patterns in
the data. For example, visualizing the most spoken languages in the world would require such visualization algorithms. Dimensionality reduction is a technique that merges correlated features into a single feature and, as a result, simplifies the available data without losing too much information. The mileage of a car may be correlated with the age of the car. Therefore, using dimensionality reduction or feature extraction these correlated features can be merged into a single feature named the wear and tear of the car. By utilizing this technique, the model will train faster and lesser memory space is required to hold the data. Principal component analysis (PCA) and kernel PCA are popular visualization and dimensionality reduction algorithms.

The goal of association rule learning algorithms is to explore large data files and discover interesting patterns and new relations between the various features of the data. Association rule learning algorithms may be applied in supermarkets, whereby the algorithm may reveal that people who buy bread are more likely to buy bread spreads also in that purchase. Therefore, it would be ideal to place these two products next to each other. Some association rule learning algorithms include apriori and ECLAT.

### 1.2.2.1 k-Means

k-means is one of the primitive clustering algorithms that can be used for grouping problems. A group of randomly selected centroids are used as the initial centres of \( k \) clusters. \( k \) represents the number of clusters required. This parameter is also set before running the k-means algorithm. The algorithm then performs a series of calculations that influence the new set of \( k \) centroids for the next iteration. After completing the defined number of iterations, \( k \) clusters are obtained. k-means is computationally faster than HCA and produces tighter and more spherical clusters. However, it is challenging to determine the perfect \( k \) value (6).

### 1.2.2.2 HCA

HCA is a clustering algorithm that can be classified as agglomerative HCA and divisive HCA. In agglomerative clustering, each input data sample is assumed to be an individual cluster. Similar clusters then merge into one another until ‘\( k \)’ distinct clusters are obtained. This happens after every iteration is complete. The clusters are grouped based on a proximity matrix that is updated every time the iteration is complete. The divisive HCA algorithm initially considers all the data points to belong to a single cluster. Data points that are not similar are then separated from the cluster. This algorithm is not as widely used as the agglomerative clustering technique.

### 1.2.2.3 PCA

The PCA dimensionality reduction technique is used to convert data sets having a large number of features into ones with fewer features. It can only be applied to linear data sets which are data sets that are linearly separable. The data set, having fewer attributes, would still contain most of the information. Data sets having a smaller number of features are easier to explore and analyse. ML algorithms train faster on the data sets that have undergone dimensionality reduction. In PCA, to ensure that every variable has an equal weight in contributing to the analysis, the variables are
initially standardized. Then, the covariance matrix of the data set is constructed. The principal components of the data are determined from the eigenvalues and eigenvectors of the covariance matrix. The principal components are the updated set of features that can be represented as a linear combination of the original set of features. This technique greatly increases the classification accuracy of a model (7).

1.2.2.4 Kernel PCA
Kernel PCA is an extension of PCA. It is a dimensionality reduction technique that can be applied to nonlinear data sets. A kernel function is used to project the data set into a feature space where it is linearly separable. The kernel function acts as a replacement to the covariance matrix calculated in PCA. It is used to calculate the eigenvalues and the eigenvectors that are required to obtain the principal components of a given data set. The most commonly used kernels are the polynomial kernel and the Gaussian kernel. Polynomial kernels are used for data sets modelled with nonlinear decision boundaries that are polynomial in shape, whereas, for data points that are distinguished based on the distance from a centre point, Gaussian kernels would be the preferred kernel function. Kernel PCA has an advantage over PCA as real-time data are more likely to be non-linear in nature (7).

1.2.2.5 Apriori
The apriori algorithm is a popular algorithm used in data mining to determine the relationship between different products. These relations are termed as association rules. The various items in the data set are mined and the set of items or the item set that occurs most frequently is determined using the apriori algorithm. The main factors that are used in the apriori algorithm are support, confidence and lift. The support is the probability that two items in the data set (A and B) occur together. Confidence is the conditional probability of B, given A. Lift is the ratio of support to confidence. Using these parameters and a breadth-first search approach, the apriori algorithm can determine the frequent item sets in the data set (8).

1.2.2.6 Equivalence Class Transformation Algorithm (ECLAT)
On the other hand, the ECLAT utilizes a depth-first search approach to determine the frequent item sets in a given data set. The input to this algorithm is a transaction database. A set of transactions is collectively defined as a transaction database and a transaction is an itemset. The algorithm discovers frequent item sets and association rules from the transaction database. As the ECLAT algorithm uses a depth-first search in the database, it is faster than the apriori algorithm and has a lower memory requirement (8).

1.2.3 Semi-Supervised Learning
In semi-supervised learning, algorithms can handle a combination predominantly consisting of unlabelled data and a much smaller amount of labelled data. This is particularly useful in the medical field as it usually takes a lot of time and the expertise of medical professionals to label medical scan images. Semi-supervised learning algorithms would require only a few labelled images, thus saving a lot of time and effort.
1.2.4 Analysis of ML

ML algorithms are widely used in the wireless networking domain. For instance, logistic regression models are used in determining the probability of failure of a network or a process. This is a regression problem. Classification problems such as predicting root-to-local (R2L) or denial-of-service (DoS) attacks in the networking domain can also leverage ML algorithms (9). The ML-based solutions to networking problems can also make use of feature engineering techniques like dimensionality reduction. Hence, ML in the networking domain can be used to speed up and efficiently perform fundamental networking tasks including traffic prediction, network security and packet routing. In spite of the multiple advantages of ML in networking, ML algorithms still have limitations and face challenges. ML algorithms require hand-picked features to train the network, and this tends to influence the performance of the model. Another major drawback of ML algorithms is that these algorithms require a huge amount of data for training. Fewer available data give rise to the problem of overfitting. More training data could also mean higher computation costs. Hence, DL models were introduced to overcome these challenges.

1.3 DL

DL is another branch of AI. Unlike ML, DL doesn’t treat all the features equally. DL first learns which all features significantly impact the outcome and based on that the DL creates a combination of all features for the learning process. This property of DL demands a lot of data. A DL model has at least one or more hidden layers. The hidden layers fall between the input and output layers. Hidden layers are intermediate layers through which the DL algorithm learns which combination of features can be used to get the best consistent results. DL is widely used in various supervised classification and regression problems. The training of the deep learning algorithms happens via back propagation, whereby the algorithm learns the parameters for each layer from the immediate next layer and so on. Some of the well-known DL algorithms are recurrent neural networks (RNNs), convolution neural networks (CNNs) and general adversarial neural networks (GANs). Generally, these models have many different data-processing blocks before the hidden layers. Some of the commonly used blocks are convolution, pooling and normalization.

The convolution block use kernels (or filters) to convolute multiple features at a time depending on the kernel size to get the spatial information about the data.

The pooling block is used to decrease the feature set size by either taking average or max of multiple features. This helps increase the computation speed of the algorithm and, at the same time, preserve the information.

Normalization is used to normalize the data in a feature. This is because due to multiple processing steps, the data may change significantly, and if one feature has relatively higher numbers than another feature, then the feature with a higher number dominant the results. To avoid this, we normalize the data across features so that all features are weighted equally before they enter into the hidden layers.
1.3.1 CNNs

CNNs use convolution block as one of the major functions to get the most prominent combination of features to get the results. This approach enables the algorithm to successfully capture the temporal and special dependencies between different features. The architecture of CNN facilitates the reduction of the size of the features which are easier to process, and it gets the results without losing any information.

1.3.2 RNNs

RNNs learn just like CNNs, but RNNs also remember the learning from prior inputs (10). This context-based learning approach from RNNs makes them suitable for any sequential data as the model can remember the previous inputs and the parameters learnt from them. Hence, this architecture is one of the best choices to make when dealing with series data as this model uses the data from the past to predict the present output values.

1.3.3 GANs

GANs are used for data augmentation. GANs can produce new data points with the probability distribution of the existing data points over $N$ dimensional space. The GAN model has two parts: (1) generator and (2) discriminator. The generator is used to create fake data points in addition to the existing data points based on random inputs, and the discriminator is used to classify the fake points from the existing data points. This process is repeated by updating the weights of the generator such that it increases the classification error and the weights of the discriminator such that it decreases the classification error until we get the fake points to have almost the same distribution of the original existing data points. In this way, the GAN model is able to generate new data points which have almost same probability distribution as the existing data points.

1.3.4 Analysis of DL

DL is preferred over ML because DL automates the feature selection, and the extraction process is automated as well. In ML, the features are hand-picked manually and fed to the model. DL removes this process with the help of blocks and hidden layers, whereby the model learns what combination of the feature works well for the data set considered. But at the same time, DL also has its downside. To run a DL model, a huge amount of data is required. The amount of data is proportional to the feature extraction efficiency of the DL model. So if the data set size is small, then ML algorithms perform better than DL algorithms.

1.4 Conclusion

ML and DL models have greatly influenced the way automation happens in today’s world. The development of these algorithms has enabled automation in every field,
including networks for wireless devices. By implementing ML or DL in a wireless networking domain, various processes which involved manual, time-consuming and unreliable processes are now being more refined and automated. This way a lot of manual errors and time delays are rectified. By removing manual works and automating them, the operational efficiency to find and resolve network issues is minimal and cost-effective. Hence, the evolution of these ML and DL architectures have greatly contributed to the development of the modern network domain.

REFERENCES


Overview of Machine Learning and Deep Learning Approaches


ML and DL Approaches for Intelligent Wireless Sensor Networks


Machine Learning-Based Optimal Wi-Fi HaLow Standard for Dense IoT Networks


Machine Learning Approaches in Big Data Analytics Optimization for Wireless Sensor Networks


Improved Video Steganography for Secured Communication Using Clustering and Chaotic Mapping


Test Dataset for Container Videos [video file], https://media.xiph.org/video/derf/y4m/.
Target Prophecy in an Underwater Environment Using a KNN Algorithm


**Design of Wireless Sensor Networks Using Fog Computing for the Optimal Provisioning of Analytics as a Service**


Communication Links and Networks (CAMAD), 325–329, 1 December.


**DLA-RL**


Deep Learning-Based Modulation Detector for an MIMO System


Deep Learning with an LSTM-Based Defence Mechanism for DDoS Attacks in WSNs


A Knowledge Investigation Framework for Crowdsourcing Analysis for e-Commerce Networks


Intelligent Stackelberg Game Theory with Threshold-Based VM Allocation Strategy for Detecting Malicious Co-Resident Virtual Nodes in Cloud Computing Networks


