IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received March 6, 2021, accepted March 21, 2021, date of publication March 29, 2021, date of current version April 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3069210

Applications of Machine Learning in Networking: A Survey of Current Issues and Future Challenges

M. A. RIDWAN[1], N. A. M. RADZI[2], (Senior Member, IEEE), F. ABDULLAH[2], (Senior Member, IEEE), AND Y. E. JALIL[2]

[1]Institute of Power Engineering, Universiti Tenaga Nasional, Kajang 43000, Malaysia
[2]Department of Electrical and Electronics Engineering, College of Engineering, Universiti Tenaga Nasional, Kajang 43000, Malaysia

Corresponding author: M. A. Ridwan (asyikin@uniten.edu.my)

This work was supported by the Universiti Tenaga Nasional Internal Grant RJO10517844057.

ABSTRACT Communication networks are expanding rapidly and becoming increasingly complex. As a consequence, the conventional rule-based algorithms or protocols may no longer perform at their best efficiencies in these networks. Machine learning (ML) has recently been applied to solve complex problems in many fields, including finance, health care, and business. ML algorithms can offer computational models that can solve complex communication network problems and consequently improve performance. This paper reviews the recent trends in the application of ML models in communication networks for prediction, intrusion detection, route and path assignment, Quality of Service improvement, and resource management. A review of the recent literature reveals extensive opportunities for researchers to exploit the advantages of ML in solving complex performance issues in a network, especially with the advancement of software-defined networks and 5G.

INDEX TERMS Machine learning algorithms, communication network, intrusion detection, routing, quality of service.

I. INTRODUCTION

In recent years, network traffic has grown exponentially along with increasing bandwidth and delay-sensitive applications, including voice over Internet protocol, 5G long-term evolutions, and on-demand videos. Accordingly, modern communication network systems need to be improved to manage traffic and fulfill their service level agreement (SLA) efficiently. However, the growth in the number of applications with different Quality of Service (QoS) requirements, the ever-increasing number of subscribers, and the complexities in resource allocation and management all introduce challenges that prevent network designers from optimizing network performance and achieving the highest efficiency.

Machine learning (ML) is a subset of Artificial Intelligence (AI) application that allows systems to learn automatically and provides predictions or solutions based on experience. ML brings intelligence to various broad sectors, such as image and speech recognition [1]–[4], guiding systems [5], and communication networks [6], [7]. ML is also known for its superior classification and optimization performance, has been used to intelligently address complex problems, and, in some cases, can produce better outputs compared with humans [8].

This paper surveys the recent trends in the application of ML algorithms in the communication literature published from 2017 to 2020. Instead of merely focusing on a single network such as software-defined networking (SDN), this paper presents an overview of how ML algorithms can be incorporated into various networks, including multi-domain, multilayer, optical, Internet of Things, and 5G networks. Given that networks are becoming exponentially complex and the gradual obsoletion of conventional algorithms over time, ML can be viewed as a revolutionary solution to the challenges being faced in networks, hence motivating this paper to survey the recent trends in the application of ML in solving such issues and to highlight its advantages over conventional solutions. The contributions of this paper are outlined as follows:

- This paper provides an overview of ML algorithms, including supervised and unsupervised learning models, deep learning (DL), and reinforcement learning (RL).

The associate editor coordinating the review of this manuscript and approving it for publication was Kathiravan Srinivasan.

- This paper reviews the related applications of ML between 2017 and 2020. These applications include congestion control, predictive model, intrusion detection system (IDS), routing, QoS improvement, and resource management.
- This paper discusses the issues encountered by conventional algorithms, the advantages of the current implementation of ML algorithms, and the current methods for solving network issues.
- This paper presents the future challenges and trends in the application of ML algorithms in networks.

The rest of this paper is organized as follows. Section II reviews the related papers. Section III provides an overview of ML algorithms, including supervised and unsupervised learning, DL, and RL. Section IV discusses the applications of ML in congestion control. Section V discusses the application of ML to predict the QoS in a network. Section VI discusses the application of ML-based IDS and elaborates the recent common intrusions in a network, the publicly available intrusion datasets, and future recommendations. Section VII discusses the application of ML for route and path assignments. Sections VIII and IX discuss the application of ML to improve QoS and network resource management, respectively. Section X elaborates the future challenges and trends in the application of ML algorithms in networks. Section XI concludes the paper. Fig. 1 summarizes the organization of this paper, and Table 1 lists the acronyms and notations used in this study.

TABLE 1. Definitions of acronyms and notations.

Acronym	Definition
AI	Artificial Intelligence
ANN	Artificial Neural Network
BF	Bellman-Ford
C-NN	Convolutional NN
C-ANN	Convolutional ANN
C-RAN	Centralized Radio Access Network
DBA	Deep-Belief Architecture
DL	Deep Learning
DNN	Deep Neural Networks
DoS	Denial-Of-Service
DRL	Deep Reinforcement Learning
DT	Decision Tree
ELM	Extreme Learning Machine
FSL	Few-Shot Learning
FSM	Finite State Machine-Based
GMMR	Gaussian Mixture Model Routing
GP	Gaussian Process
LA	Learning Automata
LSTM	Long Short-Term Memory
MARL	Multi-Agent Reinforcement Learning
MHWSN	Metro Heterogeneous Wireless Sensor Networks
ML	Machine Learning
NB	Naïve Bayes
NWN	Next-Generation Wireless Network
OppIoT	Opportunistic Internet Of Things Network
QoS	Quality-Of-Services
QoT	Quality-Of-Transmission
RL	Reinforcement Learning
RMSA	Routing, Modulation Format, and Spectrum Assignment
SDN	Software-Defined Network
SVM	Support Vector Machine
VIADL	Value Iteration Architecture-Based Deep Reinforcement Learning

II. RELATED WORKS

Numerous studies have reviewed the application of ML in networks. For instance, Miller *et al.* [9] comprehensively reviewed the application of deep neural networks (DNN) in defending networks from attacks. Mammeri *et al.* [8] reviewed the literature on the application of RL for routing. Several recent reviews on the application of DL in networks have also been conducted in [10]–[14]. Otoum *et al.* [10] comprehensively analyzed the application of ML and DL solutions for IDS in sensor networks. Shrestha and Mahmood [11] reviewed the available optimization methods that utilize different types of deep architectures to improve accuracy and reduce training time. Zhang *et al.* [12] reviewed mobile and wireless research based on DL. Usama *et al.* [13] provided an overview of the applications of unsupervised learning in the networking domain. Luong *et al.* [14] addressed the recent issues in the application of deep RL in communication and networking. These studies focus on specific ML algorithms or applications.

However, a broad range of ML algorithms come with their own advantages and disadvantages. ML algorithms have limitless applications in networks and can be implemented either in the present network or its future evolutions. Therefore, the application of a broad ML algorithm in a specific network has been recently reviewed. For instance, Yao *et al.* [15] reviewed the application of ML for load balance routing in a next-generation wireless network. Zhao *et al.* [16] incorporated different ML algorithms in SDN. Praveenkumar *et al.* [17] reviewed the application of different ML-based algorithms in wireless sensor networks. Although the ML algorithm covered in these papers are broad, they specialize in specific network applications or network types. By contrast, Boutaba *et al.* [18] comprehensively surveyed the various network applications of ML, such as for traffic predictions, traffic classifications, routing, congestion control, and resource management. This paper also provides insights for future research. While Boutaba *et al.* provides one of the most complete surveys on the application of ML, their review is only limited up to 2018.

Given that research on the application of ML in networking is still ongoing and has many opportunities for expansion, this paper provides a review of the recent trends in the application of ML in networks and an overview of the most recent advancements in this field. This paper also identifies gaps that can be filled in future studies on the incorporation of ML into networks. Our latest study shows that the most prominent applications of ML in networks include congestion control, network performance predictions, IDS, routing, QoS improvements, and resource management. Therefore, this survey will focus primarily on these six applications. Table 2 summarizes the recent related works along with their focus and contributions.

III. OVERVIEW OF ML MODELS

ML has been recently applied in various aspects of our lives, including financing, health care, robotics, customer

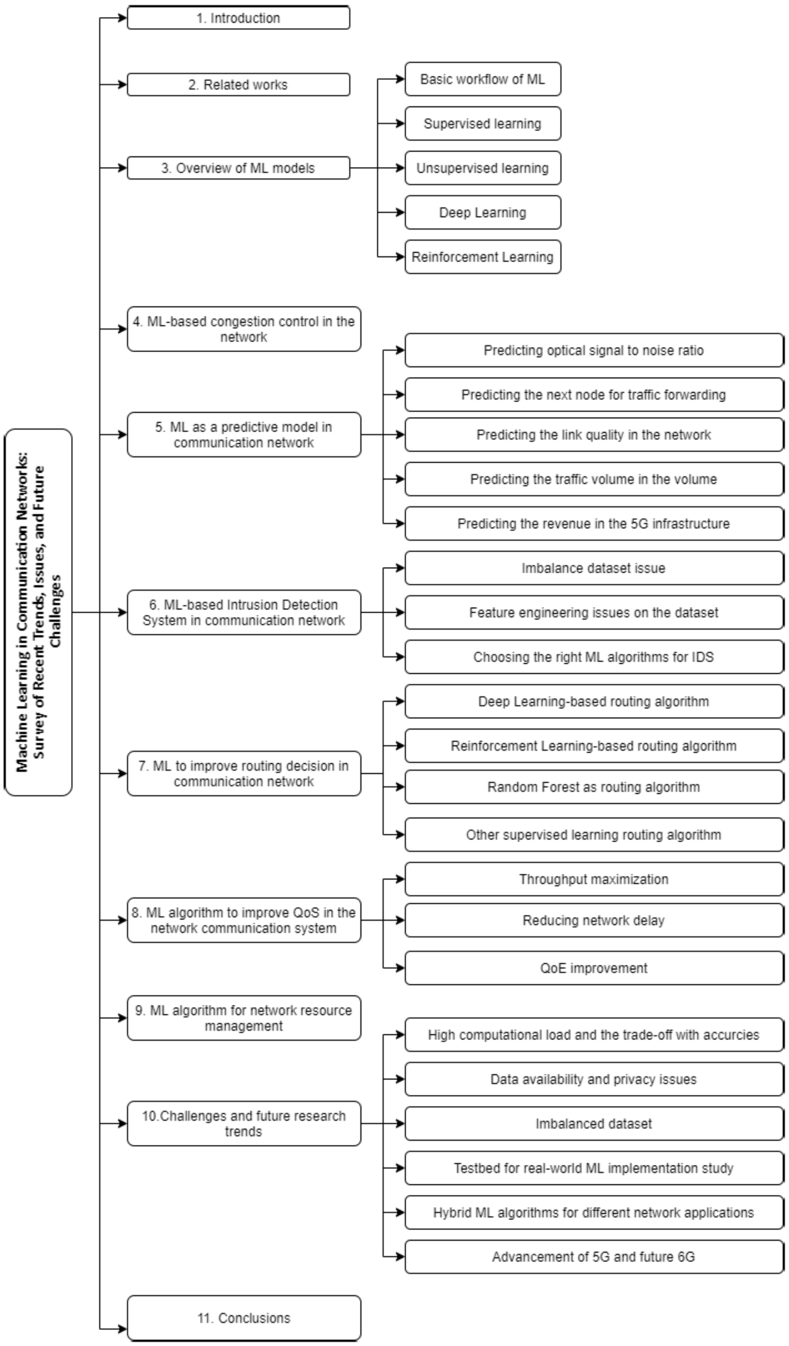

Machine Learning in Communication Networks: Survey of Recent Trends, Issues, and Future Challenges
1. Introduction
2. Related works
3. Overview of ML models
Basic workflow of ML
Supervised learning
Unsupervised learning
Deep Learning
Reinforcement Learning
4. ML-based congestion control in the network
5. ML as a predictive model in communication network
Predicting optical signal to noise ratio
Predicting the next node for traffic forwarding
Predicting the link quality in the network
Predicting the traffic volume in the volume
Predicting the revenue in the 5G infrastructure
6. ML-based Intrusion Detection System in communication network
Imbalance dataset issue
Feature engineering issues on the dataset
Choosing the right ML algorithms for IDS
7. ML to improve routing decision in communication network
Deep Learning-based routing algorithm
Reinforcement Learning-based routing algorithm
Random Forest as routing algorithm
Other supervised learning routing algorithm
8. ML algorithm to improve QoS in the network communication system
Throughput maximization
Reducing network delay
QoE improvement
9. ML algorithm for network resource management
10.Challenges and future research trends
High computational load and the trade-off with accurcies
Data availability and privacy issues
Imbalanced dataset
Testbed for real-world ML implementation study
Hybrid ML algorithms for different network applications
Advancement of 5G and future 6G
11. Conclusions

FIGURE 1. **Organization of this paper.**

service, and pattern recognition. ML algorithms can provide low-complexity solutions to performance issues by exploiting the historical data without the need for any complex re-programming [24]. With the expanding volume of complex data generated and the demand for intelligent data analytics, the use of the ML algorithm has become ubiquitous. Fueled by advancements in computing power, ML has received recognition in both research and industry for its potential to extract information efficiently from a large dataset [25]. Given the high predictive accuracy of ML approaches and the rapid speed by which an ML model can be generated, ML is being used every day by thousands of companies for the predicting the next best business [26].

ML tasks often depend on the nature of the training data. During the training process, the ML framework is trained to achieve a specific goal, such as making a decision, predicting a value, or performing a classification. Training allows the ML framework to discover potential relationships between the input and output data without any human intervention [27]. Another form of ML is the online ML algorithm,

TABLE 2. Summary of the Recent Related Works with Their Focus and Contributions.

Authors	Year	Focus and Contributions	ML Model	Applications
Miller et al. [9]	2020	Comprehensive review of defenses against attacks	Deep Neural Network	Adversarial Learning Attack Defense
Mammeri et al. [8]	2019	Reviews the literature regarding the application of RL to routing	Reinforcement Learning	Routing
Otoum et al. [10]	2019	Comprehensive analysis of the use of the machine and deep learning solutions for Intrusion Detection	Deep Learning	Sensor Network
Shrestha et al. [11]	2019	Reviews optimization methods to improve the accuracy of the training and to reduce training time	Deep Learning	Different Types of Deep Architectures
Zhang et al. [12]	2019	Provide a review of mobile and wireless research based on deep learning	Deep Learning	Mobile And Wireless Networking
Usama et al. [13]	2019	Provide an overview of applications of unsupervised learning in the domain of networking	Unsupervised ML	General Networking
Luong et al. [14]	2019	Review Deep RL approaches to address the recent emerging issues	Deep RL	Communication and Networking
Zhao et al. [16]	2019	Presents the network applications combined with SDN concepts based on ML	Broad ML Algorithms	Software-Defined Network
Chaabouni et al. [19]	2019	Comprehensive review of network IDS deploying different aspects of learning techniques for Internet of Things	Broad ML Algorithms	IDS for Internet of Things
Sultana et al. [20]	2019	Review various recent works on ML methods that leverage SDN to implement NIDS	Broad ML Algorithms	IDS for SDN
Chen et al. [21]	2019	Review the application of deep learning on the edge Edge computing	Deep Learning	Computing Network
Yao et al. [22]	2019	Presents load balance routing based on machine learning	Broad ML Algorithms	Next-Generation Wireless Network
Zappone et al. [23]	2019	Survey of the literature on deep learning for wireless communication networks	Deep Learning	Wireless Network
PraveenKumar et al. [17]	2018	Present various ML-based algorithms for Wireless Sensor Networks	Broad ML Algorithms	Wireless Sensor Networks

in which after every prediction, the model is updated for each new input feature [26].

A model can be built from the historical data or dataset by using the ML algorithm. The model is then evaluated to check whether the desired accuracy has been reached and to determine whether optimization is required to improve performance. New data are then fed into the model to make predictions. For online ML models, the new predicted results are updated into the model. The differences between ML algorithms and the rule-based algorithm are summarized as follows [26]:

- ML creates its model based on the dynamic input features and complexity of the data with no fixed rule.
- ML is often more accurate, automated, fast, customizable, and scalable than a manually constructed rule-based system.
- ML can be trained to identify trends and patterns from a large volume with multi-dimensional data to provide future predictions for a particular problem.

ML, which is a subset of AI technology that learns patterns from empirical data, has been applied in classification, regression, and control. The basic workflow of ML algorithms is depicted in Fig. 2. The dataset is fed into the ML platform to train the algorithms. The ML platform then builds the model, whose accuracy is subsequently evaluated. If the accuracy is not promising, then further optimization is required. This process is repeated until the accuracy of the algorithm converges. The trained ML algorithm is then further validated on a new data to ensure that the algorithm still provides good accuracies. This is also an important performance measure to prevent the algorithm from overfitting the training dataset. The ML algorithm can be trained with a labelled dataset that informs the machine about the correct answers in a process known as supervised learning. Many algorithms, including decision tree (DT), logistic regression, and k-nearest neighbor (KNN), use this approach

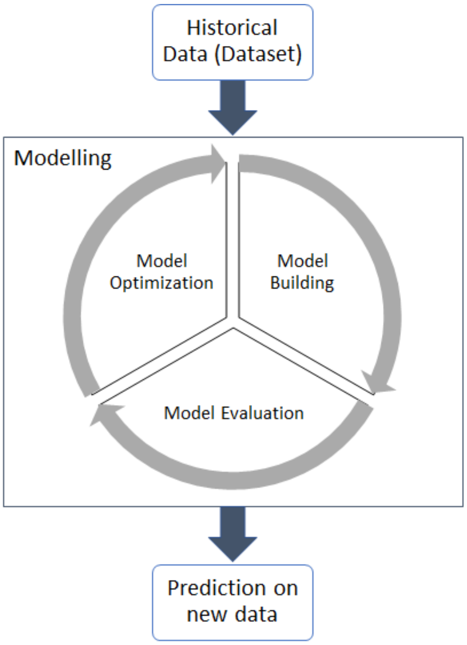


FIGURE 2. The basic workflow of real-world ML system [26].

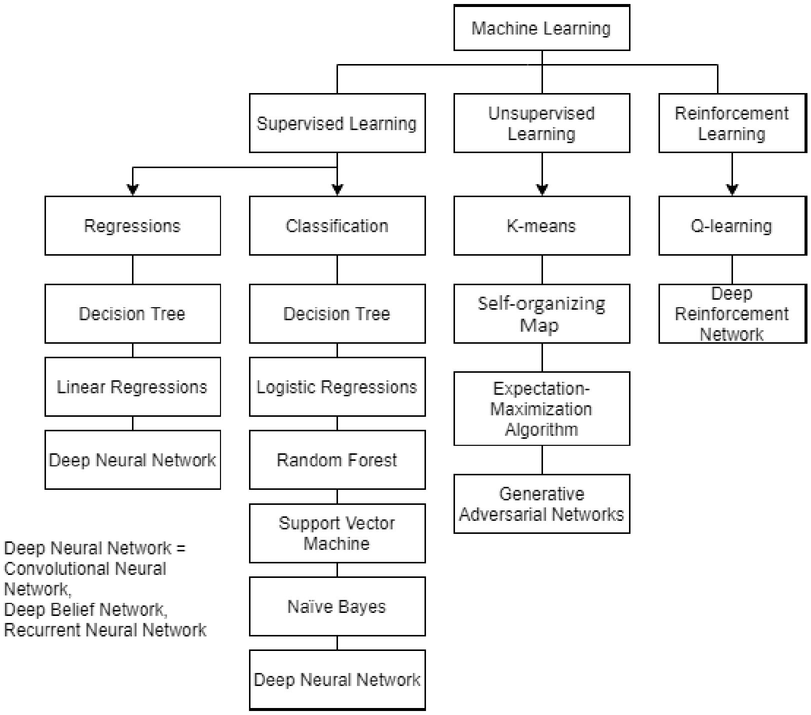


FIGURE 3. **ML model categories.**

to perform regression or classification. When dealing with labeled data, both input and desired outputs are known by the system. The supervised learning approach is commonly used when sufficient historical data are available. Some ML algorithms are also fed with unlabeled datasets. The model looks for associated or clustering patterns without the correct answers in the dataset. The main goal of unsupervised learning is to explore the data and infer some structure directly from unlabeled data. K-means, self-organizing map (SOM), expectation–maximization (EM), and generative adversarial networks are some examples of unsupervised learning algorithms. Unsupervised learning is essential in the event where the applications have no labeled data. Another form of the ML model is RL. RL allows an agent to take action and interact with the environment to maximize the total rewards. The basic workflow of RL will be elaborated further in this paper. Fig. 3 summarizes the subdivisions of the ML algorithm, including RL, supervised ML, and unsupervised ML. This section presents an overview of the classical supervised and unsupervised learning algorithms and briefly explains each ML algorithm. A well-trained ML algorithm should be able to perform predictions in a system with remarkable accuracy. Nonetheless, when the behaviors of the system change rapidly, ML algorithms may need to be re-trained to be adaptive to the new changes. To overcome this problem, another form of ML algorithm, namely, online ML is introduced. Online ML assumes an initial model that can generate predictions without any pre-deployment effort as soon as the system is up. This model can be updated every time a new occurrence in the system is observed. Online ML updates periodically while adapting to changes in the effort to improve the accuracy of the algorithm [28]. For additional information on each algorithm discussed in this section, readers can refer to [26], [29]–[43].

A. DECISION TREE (DT)

The DT algorithm can be used to address both classification and regression problems. This algorithm considers all input attributes and features at the root and is divided into groups of splits at the first phase. The accuracy of each split is calculated by using a cost function, and the split with the least cost is selected. DT is recursive given that each split formed can be subdivided by using the same strategy. The next phase is tree pruning, which identifies and removes branches that reflect noise or outliers [29], [44]. Given this procedure, the DT algorithm is known as a greedy algorithm. DT has an excessive desire to lower the costs. The cost function seeks to find the most homogenous group of branches with the most similar responses. The maximum depth of DT refers to the length of the longest path from the root to a leaf. The depth is set to a value that balances the accuracy of the model and avoids overfitting the training data at the same time. The advantages of DT are as follows:

- simple to interpret and visualize;
- implicitly performs feature selection; and
- unaffected by the nonlinear relationship among parameters.

However, DT can be over-complex and overfit the training data. This algorithm is also prone to instability because a small variation in the data may result in the generation of a completely different tree [29], [44]. This model may also be unable to deal with complex systems with inconsistent attributes.

B. RANDOM FOREST (RF)

One of the downsides of DT is that the top levels of the tree have a huge impact on the output. If the new data does not follow the same distribution as the training dataset, DT may suffer inaccuracies. The RF model can help to mitigate such issues [26]. The DT model is the primary building block of the RF model. As the name implies, the RF model consists of a large number of individual DTs that operate as an ensemble [30], [33], [45], [46]. Each DT in the RF splits out a class prediction, and the class with the most votes is selected the model prediction as illustrated in Fig. 4. A large number of uncorrelated trees operating as a committee will outperform any of the individual constituent models. Each tree in the RF may give the wrong prediction, whereas the other trees may produce an accurate prediction. The RF constructs multiple DTs and eventually merges them to obtain an absolute and stable value, which is mainly used at the time of training and predicting the class [30], [33], [45], [46]. As a result, this group of trees provides an improved prediction. RF also forces additional variations into the model, which will ultimately reduce the correlation across branches due to diversification. The advantages of RF algorithms include their immunity to few correlation features and noisy datasets and their significant gains in accuracy [47]. Moreover, RF is robust against overfitting unlike DT. Apart from having a high-performance classification model, RF can calculate the importance and degree of influence of each variable used in the classification [48].

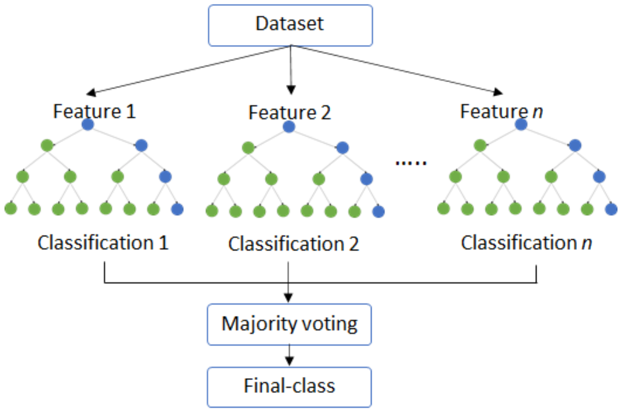


FIGURE 4. Illustration of Random Forest classifier [26].

C. SUPPORT VECTOR MACHINE (SVM)

SVM is highly preferred by many due to its significant accuracy yet low computation power. Similar to DT, SVM can be used to solve both regression and classification problems. SVM aims to find a hyperplane in an N-dimensional space that distinctly classifies the data points [31]–[33], [49] as illustrated in Fig. 5. To separate several classes of data points,

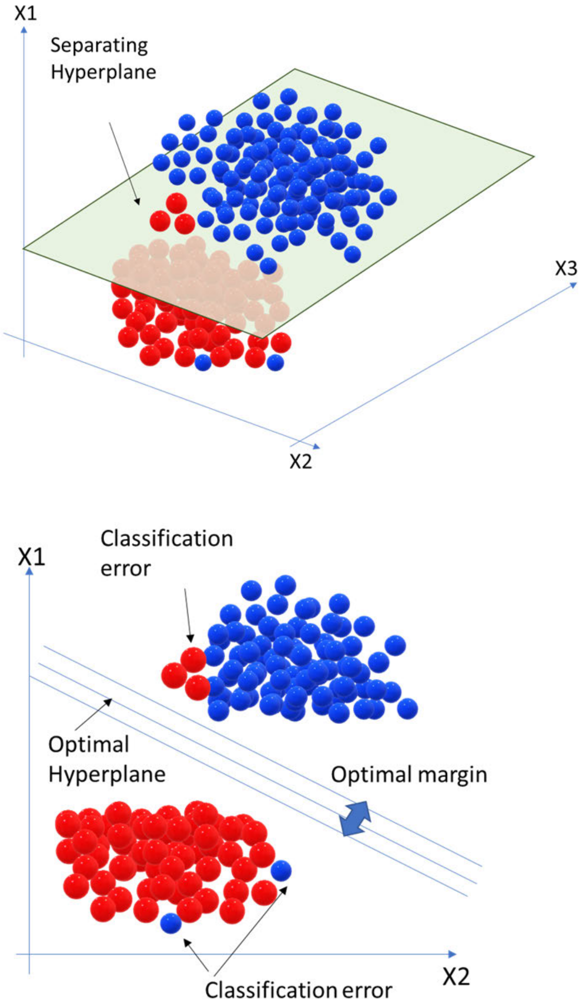


FIGURE 5. Illustration of SVM hyperplane in 3-dimension space and its optimal hyperplane and margin in 2-dimension space [26].

many hyperplanes can be constructed. The main objective of SVM is to find a plane with the maximum distance between the classes, where the hyperplane divides the blue and red data points with some classification error. Maximizing the optimal margin distance provides some reinforcement in order for future data points to be classified with improved accuracy. A higher number of features corresponds to a more complex construction of the hyperplane. SVM can also fit linear and nonlinear data and uses a the kernel technique, which is a mathematical construct that can ‘‘wrap’’ the space where the data are located. SVM can then find a better boundary in this wrapped space, thereby making the boundary nonlinear in the original space [26], [33], [49]. SVM has an excellent generalization performance, hence making this model suitable for small datasets with many features.

D. NAÏVE BAYES (NB)

An NB classifier is a probabilistic ML model that is used to solve classification problems. The operation of this classifier is based on the Bayes theorem presented in Equation (1).

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1}$$

where P(A|B) is a conditional probability of the likelihood of event A occurring given that event B is true, P(B|A) is the conditional probability of the likelihood of event B occurring given that event A is true, and P(A) and P(B) are the probabilities of observing events A and B, respectively. The NB algorithm is mostly used in sentiment analysis and recommendation systems. One of the most significant disadvantages of NB is requiring the input attributes to be independent, which is not always the case in real-life applications. Such disadvantage will ultimately hinder the performance of this classifier [33], [49], [50]. However, in the case where the input attributes can be independent, NB is preferred due to its simplicity and fast processing speed compared with other classifiers. Similar to SVM, the NB classifier has the advantage of requiring only a small amount of training data to estimate the parameters required for classification.

E. REGRESSION

Regression is a method of modeling a target value based on independent variables. This method is widely used for forecasting and determining the cause and effect relationship among variables. The most basic regression model is linear regression, which is a type of regression analysis where only one independent variable is present and where the relationship between the independent x and dependent y variables is linear as shown in Equation (2).

$$y = a_o + a_1 x \tag{2}$$

Equation (2) shows the cost function where the a_0 and a_1 are the slope and intercept of the linear regression model with respect to each input, respectively. This equation aims to calculate the best possible values for a_0 and a_1 that can achieve the best fit for all data points. The search problem is converted into a minimization problem where the goal is to improve the fit. The gradient descent method continuously updates both values to reduce the mean squared error (MSE). The linear regression eventually fits the data point straight line or plane to the target variable as illustrated in Fig. 6. While linear regression is used to predict a numerical value, logistic regression is used to perform classification. The best line or plane for splitting the data into the target classes is constructed for the logistic regression. This approach can be extended to additional dimensions, and the performance may degrade if the decision boundary that separates the classes is highly nonlinear. Another downfall is that logistic regression can sometimes overfit the data. A process called regularization is usually employed to limit the occurrence of overfitting [26].

F. K-MEANS

K-means is one of the oldest yet most widely used unsupervised clustering algorithms. This simple partitional clustering algorithm attempts to find K non-overlapping clusters. To create K-means clusters, the K initial centroids are initially selected. The basic working model of K-means is shown in Fig. 7. The K initial number of clusters are pre-defined by users. Afterward, every data point is assigned to the closest centroid, and each collection of points assigned to a centroid forms a cluster. This process is repeated until no point changes clusters. This clustering algorithm shows several advantages over other algorithms in terms of simplicity. However, K-means performs poorly when the clusters are non-globular and is highly sensitive to outliers [35].

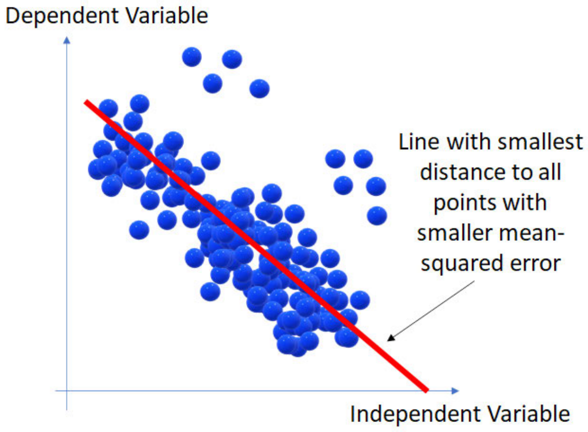


FIGURE 6. Sample Linear Regression best fit line [26].

G. SELF-ORGANIZING MAP (SOM)

SOM is an unsupervised learning algorithm that produces a low-dimensional, discretized representation of the input space of the training dataset. This algorithm is usually employed to reduce dimensionality and incorporates competitive learning [36]. First, each weight of the input nodes is initialized. Second, a vector is chosen randomly from the training data. Third, each node is evaluated by calculating those weights that are most likely to belong to the input vector. The winning node is known as the best matching unit (BMU). Fourth, the neighborhood of the BMU is calculated, and the number of neighbors decreases over time. The closer a node is to the BMU, the more weights are altered. This process is repeated until the convergence becomes valid. Two advantages of SOM are its high interpretability and capability to handle high-dimensional datasets. However, SOM also incurs high computational load, especially for large maps with dense training data [51].

H. EXPECTATION-MAXIMIZATION (EM)

EM can be applied as an unsupervised clustering algorithm to estimate the maximum likelihood in the presence of latent variables. This algorithm aims to enhance the tractability of the given incomplete data problem for ML estimation [37]. The flowchart of EM is shown in Fig. 8. The EM algorithm is an iterative approach that cycles between two modes. First,

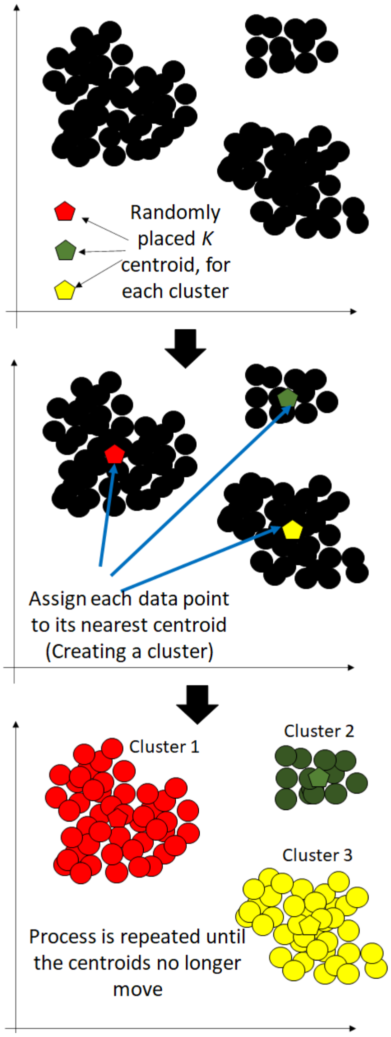

Randomly placed *K* centroid, for each cluster
Assign each data point to its nearest centroid (Creating a cluster)
Cluster 1
Cluster 2
Cluster 3
Process is repeated until the centroids no longer move

FIGURE 7. K-Means clustering [35].

a set of initial parameters is considered with incomplete data. The first mode, namely, estimation-step, attempts to estimate the missing data. Second, the maximization-step attempts to optimize the model parameters to explain the estimated value generated in the estimation-step. This process is repeated until convergence is reached. EM guarantees that the likelihood increases along with each iteration. However, EM has a slow convergence rate, thereby resulting in the development of modified versions of this algorithm [37].

I. GENERATIVE ADVERSARIAL NETWORKS (GAN)

GAN belongs to a set of generative models given its capability to generate new content. Specifically, GAN can generate an infinite number of similar samples based on a given dataset. GAN contains two NNs, namely, a generator and a discriminator, that compete against each other in a zero-sum game framework. The basic workflow of GAN is shown in Fig. 9. First, the generative network takes random noise as an input and generates samples as an output. The main goal of the generator is to generate samples that will "trick" the discriminator into thinking that it sees a real image. Second, the discriminator takes both real images from the dataset and the fake images generated by the generator and subsequently decides the legitimacy of the given image. GAN eliminates the need for direct data inputs by using a generative network and can generate sharp distributions that are superior to Markov chains. However, GAN has a long training time, thereby enhancing its complexity for real-world applications [38].

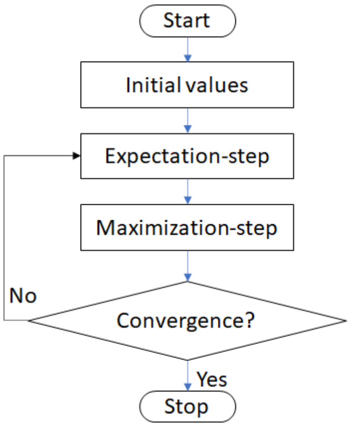


FIGURE 8. The Flowchart of EM [37].

J. DEEP LEARNING (DL)

As shown in Fig. 10, DL is a subset of ML that attempts to mimic the function of the human brain. DL has been regarded as the next paradigm to revolutionize user experience and has widely attracted the attention of networking researchers given its ability to alleviate the burden resulting from exponentially growing traffic and increasing complexity. Researchers have also investigated the application of DL in alleviating the ever-increasing communication overhead. DL mimics the biological nervous system and performs computation through multi-layer transformation as depicted in Fig. 11. The primary benefit of DL over traditional ML is its automatic feature extraction, whereby the expensive hand-crafted feature engineering can be circumvented. By contrast, traditional supervised ML is only useful when sufficient labeled data are available. However, most current systems generate unlabeled or semi-labeled data. DL provides a solution by extracting unlabeled data to find useful patterns [12], [33], [39]–[41].

DLs have been applied in many domains, including computer vision, natural language processing, and big data analysis. DL can also be used to perform both supervised and unsupervised learning and can result in accurate, prompt actions due to its efficiency in extracting features from the input and finding the relationships among multiple metrices by training with massive data [52]. DL comprises an artificial neural network (ANN) that makes the core computational unit focus on uncovering the underlying patterns or connections within a dataset similar to what the human brain does when

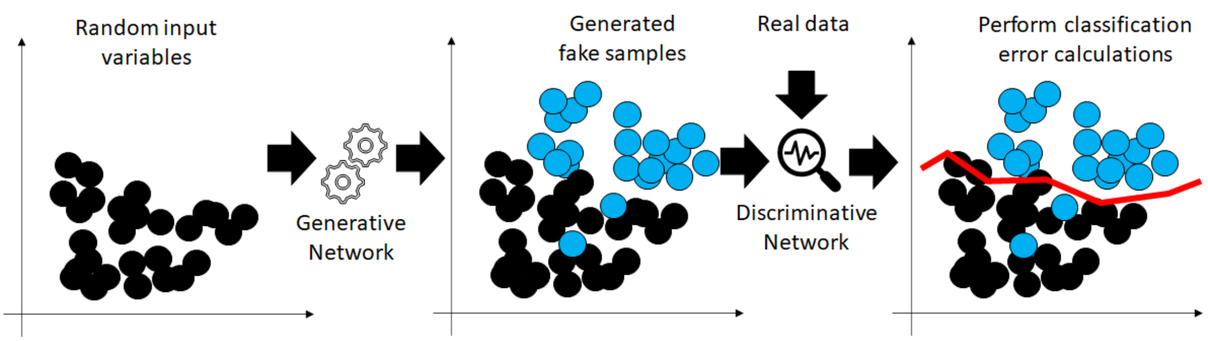


FIGURE 9. Basic workflow of GAN [38].

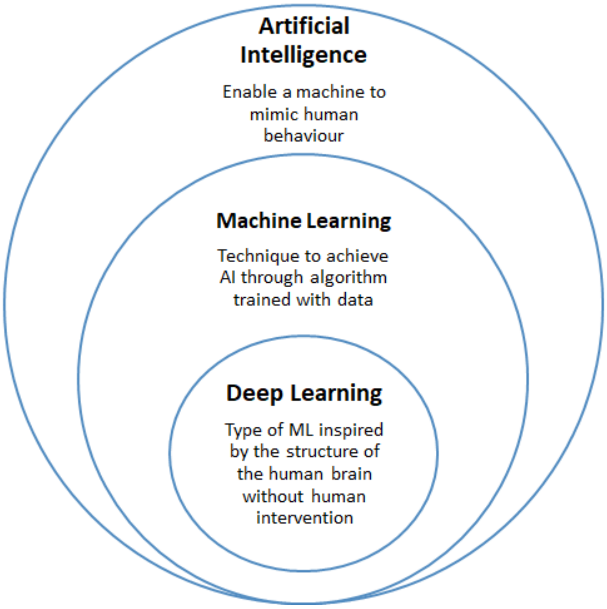

Artificial Intelligence
Enable a machine to mimic human behaviour
Machine Learning
Technique to achieve AI through algorithm trained with data
Deep Learning
Type of ML inspired by the structure of the human brain without human intervention

FIGURE 10. Difference between AI, ML and DL [12].

making a decision. The structure of the DL is similar to how neurons are arranged in the human brain.

DLs have several layers of ANN that carry out the ML process. The first layer is the input layer, which comprises a series of neurons, processes the raw input data, and passes the information to the second layer with some weights. The second layer is known as the hidden layer, which processes the information further by adding additional weights. All inputs are eventually summed up and added with another pre-determined number called bias before being passed to the activation function where the final output is either 1 or 0. At the final layer, the predicted output is compared with the actual output. If the predicted output is not matching with the actual output, then the ANN will perform the backpropagation process, whereby the process is repeated after adjusting the weights to minimize errors. This process is continuously performed across all layers of the ANN until the desired results are obtained. The advantages of the ANN include the following:

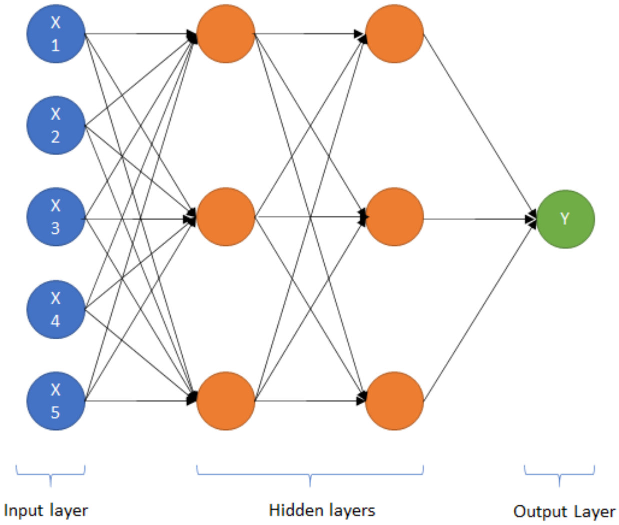


FIGURE 11. Artificial Neural Networks with two hidden layers.

- easily models multi-complex tasks [27];
- requires a small number of stored variables yet yields high accuracy [53]; and
- a well-trained ANN can be thought of as an "expert" in dealing with human-related data [54]

There are various types of DL algorithms and the most used in the field of networking including Convolutional NN (CNN), Long Short-Term Memory networks (LSTM), and Deep belief network. While no one network is considered perfect, some DL algorithms are better suited to perform specific tasks.

K. REINFORCEMENT LEARNING (RL)

Another form of ML model is RL. The workflow of RL is illustrated in Fig. 12. RL is trained iteratively from the data collected from the model itself. The goal of RL is to learn from the environment and find the best strategies for a given agent. In contrast to the supervised ML model, RL does not learn from a given dataset. Instead, an RL agent learns from the significance of its activities and chooses its action based on past information and a new choice.

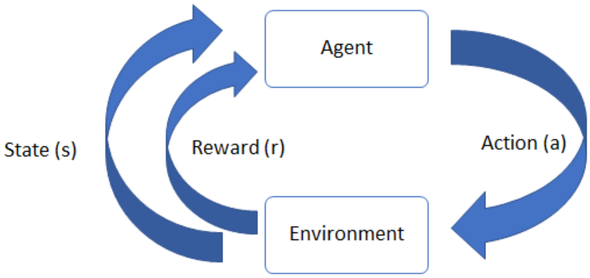


FIGURE 12. **The basic workflow of the RL algorithm.**

In this way, RL is essentially a trial and error learning technique [27], [33], [42], [43]. RL learns the reward from a particular action and gives a feedback loop to the algorithm. The agent then changes its action based on the previous reward. The agent continues to interact with the environment by learning both the action and reward until the reward saturates or reaches a pre-defined threshold [8], [23], [33]. The agent then learns how good or bad its action was based on the rewards received from the environment. This procedure resembles the decision-making process of humans.

The study of RL involves the construction of a mathematical framework to solve a given problem. To find a good policy, valued-based methods, such as Q-learning, are used to measure how the action of an agent performs in a certain state. The RL algorithm has two general tasks, namely, policy evaluation and policy improvement, the former of which calculates the cost related to the current policy, whereas the latter assesses the obtained cost and updates the current policy. The RL policy and value iteration algorithms repeatedly perform policy evaluation and improvements until an optimal solution is found [33], [55].

However, real-world problems can be extremely complex, thereby preventing a typical RL algorithm from providing an effective solution. Moreover, throughout the RL learning process, despite reaching convergence, determining the best policy may consume much time because the RL algorithm needs to explore and gain knowledge of the entire system. This shortcoming will ultimately make RL unsuitable for large-scale networks [14]. To overcome these problems, researchers have introduced an enhanced version of RL, namely, deep RL (DRL) [56], which exploits the advantages of DNNs to train the learning process and improve the learning speed of the RL algorithm. DRL also achieves an autonomous decision making and significantly improves the learning speed.

The following sections discuss the applications of supervised, unsupervised, and RL algorithms in a network.

IV. ML-BASED CONGESTION CONTROL IN THE NETWORK

Congestion presents a key concern for network providers given its degrading effects on overall network performance. Without proper congestion control and management, the network may encounter delays and underutilize its available resources. Congestion control ensures network stability, fair resource utilization, and acceptable packet loss ratio [57]. Different network environments deploy their own sets of congestion control mechanisms. Conventional routing protocols do not learn from their past experiences regarding network abnormalities, including network congestion. The perpetual growth of network traffic places a significant amount of stress on the network, thereby leading to challenges in resource allocation and management. As a result, the QoS of network traffic is affected because most networks are still operating on routing frameworks that have been designed decades ago [58]. The conventional routing protocols are originally designed for a fixed network that calculates the shortest path based on distance vectors or link costs. In the end, the network may suffer from excessive traffic load that will degrade its performance entirely. When such situation reoccurs in the future, conventional routing strategies typically make the same mistake; these strategies ought to make the same routing decisions all over again, thereby leading to an uncontrollable increase in delay and packet loss rate. The predictive ML model can be used to address such congestion.

Complex network applications that deal with massive dynamic bandwidth requirements that optimize the routing of several end-to-end connections are typically complex. To overcome such complexity, Troia *et al.* [59] implemented an intelligent network optimization model called ML routing computation that drives the provisioning of paths in the SDN network. SDN has emerged as one of the most promising technologies for implementing centralized and programmable control planes. By exploiting the programmability features of SDN, the logistic regression classifier model is implemented in the network due to its simplicity. As a logically centralized control plane, SDN can obtain network information, including topology, bandwidth request, link load, and network device status. The classifier captures the traffic metrics, including the number of bytes in each traffic flow, from the switches. A real-time routing decision is then made upon detecting changes in the network traffic matrix. The ML-based routing scheme proposed by Troia *et al.* provides smarter routes by reducing network congestion. In contrast to conventional routing where the shortest path is the desired route, this routing scheme can avoid bottlenecks and congestion in advance. This scheme only takes 80 ms to capture the network conditions, obtain the routing configurations, and provide new flow rules to the switches.

Congestion is one of the most prominent issues in ensuring QoS in wireless mesh networks. One of the most common congestion avoidance protocols is the transmission control protocol (TCP). However, TCP suffers from performance degradation [60]. Moreover, the source node cannot explicitly determine whether a packet loss has occurred due to buffering overflow or temporary link failure by the TCP [61]. As mentioned previously, conventional routing protocol does not learn from previous experiences in handling congestion, thereby resulting in a recursive problem unless a proper congestion control mechanism is implemented in the system.

In a wireless mesh network, the congestion window size is not precisely adjusted and selected by using the previous cross-layer handling link asymmetry scheme (CHLA-QSCACAR). Accordingly, Yuvaraj and Thangaraj [61] proposed an improved cross-layer handling link asymmetry scheme with enhanced QoS-based congestion avoidance that ensures its adaptiveness to the congestion window size by using ML approaches. In this work, input features, including the congestion factor, window decrease factor, aggressive factor, data rate, and packet loss rate, are fed into the SVM model. Afterward, the next congestion window size is predicted and used to adjust the congestion window in the following transmission. Yuvarai *et al.* proved that CHLA-QSCACAR can improve the throughput by 10.48% and end-to-end delay by 11.47% as well as reduce the interference and noise ratio by 19.35% and routing overhead by 12.5%. Results show that by using ML to aid the current link handling scheme, the network congestion can be further improved.

The implementation of DL for congestion control has been limited to the baseline routing protocol. As elaborated in Section III, DL works with a set of initialization weights for all neurons in the input layer of the ANN. In most cases, to solve the congestion issues in the network, the DL applies the open shortest path first (OSPF) as the baseline, which lacks the required intelligence to deal with newly occurring situations [58]. To address this shortcoming, Tang *et al.* [58] proposed a novel real-time DL-based intelligent network traffic control method by exploiting the deep convolutional NN (deep CNN) with uniquely characterized input and output to represent the considered wireless mesh network backbone. At the initial phase, instead of using OSPF as the baseline, all routers in the network calculate and record the possible paths for each destination node. All paths are arranged as a minimum priority queue depending on their metric values, including their hop number and distance. After obtaining enough training data, the training is accomplished via a periodic real-time updating phase. The valid path combinations are then intelligently chosen and executed by the proposed deep CNN model. The proposed scheme is compared with OSPF, intermediate system to intermediate system, and routing information protocol, and the simulation results prove the superiority of the DL-based routing scheme, which avoids 98.7% of the congestion cases compared with other routing protocols.

V. ML AS A PREDICTIVE MODEL IN COMMUNICATION NETWORK

The prediction of network parameters, such as path or link quality, delay, throughput, optical signal-to-noise ratio, and incoming traffic, plays an important role in network operations and management. ML aims to learn from historical data or the environment and make prediction of the network parameters to improve the efficiency of the entire network system.

A. PREDICTING THE NETWORKS OPTICAL SIGNAL TO NOISE RATIO

Recent works in [62]–[64] have used ANN, SVM, and Gaussian process regressions, respectively, to predict the OSNR in an optical network. The predicted OSNR for each source-to-destination path are then used by the system to determine the best path for the incoming traffic.

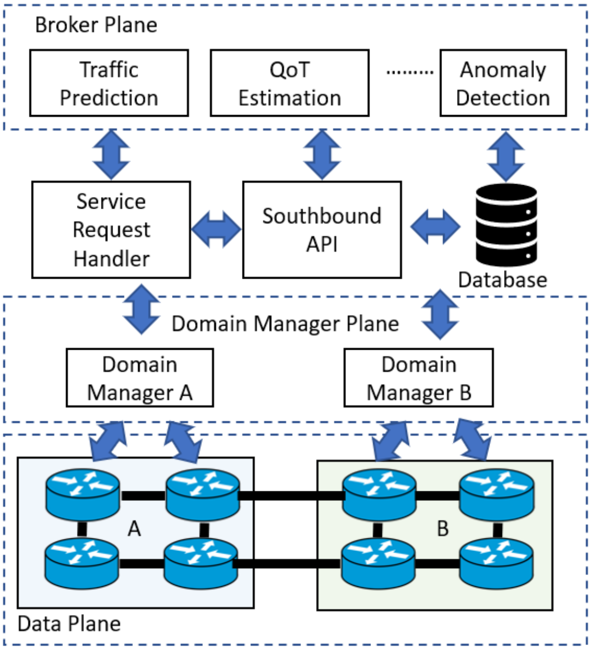


FIGURE 13. QoT estimation in multi-domain Elastic Optical Networks as proposed in [62].

Estimating the Quality-of-Transmission (QoT) in an elastic optical network is an incredibly challenging task that can lead to inaccuracies, especially when supporting high-capacity and dynamic traffic demands across multiple autonomous systems. In addition, making QoT predictions is extremely challenging in a multi-domain network. In each domain, only a minimal amount of inter-domain information is disclosed to the domain manager (DM), thereby creating a considerable disadvantage when using the ML algorithm for QoT prediction given that this algorithm heavily relies on the availability of large quantities of performance monitoring data to learn and make predictions. To overcome the privacy issues in a multi-domain network, Proietti *et al.* [62] proposed an alien wavelength performance monitoring technique and used ML algorithms to estimate the network QoT for the light path provisioning of intra-inter-domain traffic as shown in Fig. 13. Alien wavelength refers to those light paths that are not under the direct control of the domain. For each domain, the DM monitors performance in the intra-domain connections and alien wavelengths. For every light path request, the DM immediately calculates the cognitive routing, modulation format, and spectrum assignment (RMSA) solution that satisfies the QoT requirement. Afterward, DM calls the domain-level QoT estimation model to estimate the QoT of each candidate path solution provided by the RMSA. The DM then chooses the RSMA that satisfies the QoT requirements. The broker plane, which connects all DMs in the network, receives the intra-domain candidate light paths together with their estimated QoT. The broker plane then builds a multi-domain virtual topology and further calculates the inter-domain end-to-end RMSA on the broker-level QoT estimation model. With this model, the DM can disclose most information and only relay the estimated QoT by the domain-level estimator to the broker plane. The QoT estimation model used in the system is the ANN model. The input data are obtained via a testbed to determine the BER and OSNR of active connections and alien wavelengths. The ANN then predicts the QoT of candidate RMSA solutions to the broker plane in order to create a virtual topology for end-to-end routing. The proposed cognitive function achieves an OSNR prediction accuracy of 95%. The ML-based predictive model shows that the BER at the egress node is 2.7×10^{-3}, while without the QoT estimator, the BER can reach as high as 8.0×10^{-3}, thereby validating the effectiveness of the ANN-based QoT estimator.

Locatelli *et al.* [63] proposed an in-band OSNR estimation process that relies on the ML algorithm to accurately estimate the OSNR from the in-band optical spectrum in short-distance scenarios. The generated spectral data for different configurations and resolutions are obtained via simulations. These data are used to train two ML regression-based algorithms, namely, Gaussian process (GP) regressions and SVM model. The accuracy of GP and SVM are compared based on their mean square error (MSE), and the MSE obtained from these models are 8.5×10^{-3} and 21.6×10^{-3}, respectively. GP shows a better accuracy than SVM in this system.

The traffic management of a core IP/Optical backbone of a large Internet service provider must deal with dynamic traffic changes under various network conditions. Choudhury *et al.* [64] proposed a hybrid ML model to predict the traffic volume for each traffic engineering tunnel at future time horizons and subsequently predict the optical performance of new wavelengths in a multi-vendor environment. After compiling all the available data for every optical path in the network, the ML algorithm predicts the path performance, and the path with the least OSNR value is chosen to route the incoming traffic. The dataset is used to train regression models, including Ridge, GP, gradient boosted trees, and RF regression trees. The simulations show that GP and RF algorithms have the lowest MSE value of 0.81. The proposed scheme can improve the efficiency and reduce the cost by 9% compared with a non-ML-based scheme due to the predictions made by the ML algorithm, which can help avoid traffic loss and increase both feasibility and efficiency by changing the IP layer topology before the traffic surge.

B. PREDICTING THE NEXT NODE FOR THE TRAFFIC FORWARDING

The conventional method for improving network QoS considers a limited number of metrics due to the challenges in manually ascertaining the relationship among multiple metrics to reduce the analysis and computation complexity. The advancement of ML algorithms, such as DL, has enabled an effective extraction of features from the input and identification of relationships among multiple metrics through training with massive data. However, the existing DL-based predictive strategies build intelligence based on a fixed topology where the existing nodes or links information is assumed to be static. A problem arises when the network topology changes, where the prediction accuracy of the DL algorithm sharply decreases. The network may have various topologies that cannot be fully covered in the training process. A study in [65] proposed a value iteration architecture-based deep reinforcement learning (VIADL) routing strategy that uses the adjacency matrix of the network node as a learning parameter. VIADL can repeatedly predict the next node until the destination is reached. This approach makes the system topology independent by focusing only on predicting the next node in contrast to deep-belief architecture (DBA), which predicts the complete path from the source to the destination. The proposed scheme guarantees a stable network performance when the network topology changes. The throughput of the proposed scheme is maintained at 144 Mbps, whereas that of DBA is reduced to 133 Mbps when dealing with network failure.

Guo *et al.* [66] reviewed the stochastic shortest path routing (SSPR) approaches that are only applicable in situations where the edge lengths are fixed. The SSPR problem has become a key topic in the literature after the emergence of 5G technology. A learning-automata-based (LA) algorithm called SSPR-hieraStructure LRI, whose workflow is similar to that of RL algorithm that learns from the environment and grants a reward for every action, is proposed to solve the SSPR problem. This approach finds the shortest path with an optimal node in each layer instead of focusing on the shortest path alone. The LA selects one of its actions to activate the next LA, and this process is repeated until no LA remains to be activated. The activated LAs, which can be viewed as selected paths, are collectively sent to the environment. If the current length is shorter than a predefined threshold, then a reward is received. Simulation results show that the proposed algorithm converges faster with the highest probability compared with other state-of-the-art LA-based SSPR algorithms.

As its name implies, the delay tolerant network (DTN) does not have a strict delay requirement and forwards its traffic opportunistically. Example DTNs include the connection between Mars orbiter satellites to ground stations on Earth, non-essential wireless sensor networks, and networks in rural or disaster areas. Some challenges being faced by DTNs include the frequent interruptions among nodes, uncertain available paths, long trip times, and asymmetric links. Dudukovich and Papachristou [67] proposed an ML-based approach to predict a set of neighboring nodes that have a high probability of delivering a message to the desired location based on historical message delivery information. The performance of this approach is compared with that of Naïve Bayes, DT, and KNN based on hamming loss, zero-one loss, F1 score, and Jaccard similarity score. Simulation results prove that in synthetic mobility scenarios and real-world cases, DT-based classifiers obtain the best performance parameters.

C. PREDICTING THE LINK QUALITY IN THE NETWORK

The computational load must be considered in any optimization approach to solve a problem. Even if the algorithm achieves the best accuracy, when the computational load is too demanding, implementing the algorithm may not be feasible. In [68], an ML approach that efficiently accomplishes the routing and wavelength assignment (RWA) for an input traffic matrix in an optical network is proposed. Despite the cost-minimized solution provided by the integer linear program (ILP), this ML approach suffers from high computational complexity and requires minutes or even hours to solve medium-sized network topology problems. Other AI-based approaches, such as genetic algorithms or RL, also suffer from high computational costs with slow convergence. Martin *et al.* [68] proposed a classifier that is trained with labeled RWA configurations that are already solved by the ILP. After training, the classifier can provide the network configuration for newly incoming traffic matrices in an online fashion. The RWA configurations are computed within a few milliseconds, thereby allowing a dynamic network adaptation and reconfiguration in response to frequently changing traffic patterns. Therefore, instead of performing ILP calculations for each incoming traffic, the network learns from historical data that have been solved by ILP and assigns a path accordingly. Numerical results show that this approach can reduce the computational time by up to 93% compared with the ILP method.

The computational load is proportional to the problem that the ML model is designed to solve. Liu *et al.* [25] reduced the problem size by formulating a small-sized optimization problem that can indirectly solve the original problem. While they find that many links in the network are not involved in the optimization solution, these links are still considered. However, part of the computing efforts is wasted by repeatedly solving similar conditions. The DL-based classifier aims to predict those links that need to be included in the optimization problem. Only the predicted useful links are kept in the formulation by applying the threshold. This approach successfully reduces the computation cost by 50% without affecting the optimality, thereby significantly improving the efficiency of solving network optimization problems.

Bote-Lorenzo *et al.* [28] used the online ML algorithm to predict the link quality in community wireless mesh networks (CWMN). They also claimed that no previous study has examined the application of this algorithm in predicting the link quality of large-scale CWMN. Real data from the FunkFeuer Wien CWMN dataset with 500 nodes and 2000 links are used to train online ML algorithms, including online perceptron, online regression trees with options, and fast incremental model trees with drift detection and adaptive model rules. The performance evaluation results show that the online perceptron algorithm outperforms the rest in terms of accuracy with light computational demand. Such performance is further evaluated by using offline supervised ML algorithms, including SVM, KNN, regression trees, and GP. Evaluation results show that the performance of SVM is on par with that of the online perceptron algorithm. However, the online ML approach requires only 0.1% of the computational load generated by SVM, thereby suggesting that the online ML approach is superior over the offline ML approach in this case.

D. PREDICTING THE TRAFFIC VOLUME IN THE NETWORK

In virtual network topology, a high number of transponders should be installed given that the network must be able to cope with the maximum daily traffic forecast during the planning period. However, the design is often over-provisioning, and most of the available capacity in the network will remain underutilized throughout the day. Morales *et al.* [69] proposed an ML-based virtual network topology reconfiguration (VENTURE) to predict traffic usage. The VENTURE framework is shown in Fig. 14. First, the data are collected for each origin to destination (OD) pairs and stored in the modeled data repository. Second, a prediction module based on ML algorithms generates the predicted OD traffic matrix for the next period. Third, the decision-maker module decides whether the current virtual network topology needs to be reconfigured by the VENTURE optimizer. After the algorithm finds a solution, the network controller implements the changes in the network. VENTURE can maximize the utilization of available transponders by reconfiguring the virtual topology to follow the predicted traffic direction changes and dynamically manage the capacity. VENTURE also saves up to 40% of transponders compared with the threshold-based method.

E. PREDICTING REVENUE IN THE 5G INFRASTRUCTURE

With the high data rates, extensive coverage, and sub-millisecond delays promised by 5G networks, this novel technology is expected to boost upon deployment. In the 5G infrastructure, network slicing can be one of the traded goods, including the network resources such as spectrum and transport network. Network slicing involves a set of network function virtualizations and divides the infrastructure into several slices, where each slice can be tailored to meet specific service requirements. The network capacity broker algorithm must decide whether to admit or reject a new network slice request that can meet the service guarantee and maximize the revenue of the network provider. Bega *et al.* [53] proposed an ML approach for 5G infrastructure market optimization and

TABLE 3. Summary of recent ML-based Predictive Model with their Previous Issues on Conventional Rule-based Method, ML Algorithms and Advantages of the Proposed ML-based Solution.

Authors	Year	Previous Issues on Conventional Rule-based Method	ML Algorithm	Advantages of the Proposed ML-based Solution
Liu et al. [25]	2020	Waste of computing effort to solve a repetitive problem.	DBN	Reduced computation cost by up to 50 %
Bega et al. [53]	2020	SLAs with network slices tenants usually impose stringent requirements on the quality perceived by the users	DRL	Performed close to the optimal point under a wide range of configurations
Locatelli et al. [63]	2019	Recent works utilize synthetic data to generate dataset are not realistic	GP and SVM	Achieved a maximum error of 1.1 dB in all the experimental scenarios
Fadlullah et al. [65]	2019	High computational cost to address RWA problems	VIADL	Guarantees more stable network performance when the network topology changes
Martin et al. [68]	2019	Cost-minimization approach suffers from high computational complexity	LR and DNN	Reduction of computational time up to 93%
Proietti et al [62]	2018	Difficulties to estimate the OSNR	ANN	Achieved accuracy of 95% optical signal to noise prediction
Choudhury et al. [64]	2018	Most DL-based algorithms only consider the static topology	GP and RF regression trees	Efficiency and cost reduction with ML-based prediction by 9%
Dudukovich et al. [67]	2018	Interruptions, high error rates, asymmetrical links issue in DTN	NB, DT and KNN	Decision tree-based classification provide a positive impact on performance
Bote-Lorenzo et al. [28]	2018	The use of online ML algorithms to make link quality predictions in a CWMN has not been previously explored	Online perceptron	Achieved a high accuracy with a low computational load

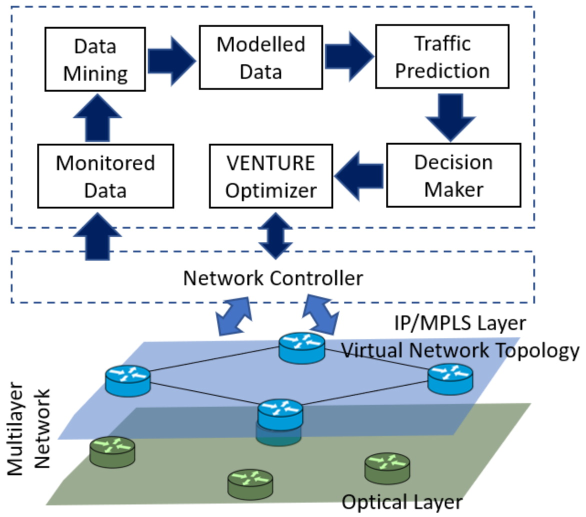


FIGURE 14. Proposed VENTURE for traffic volume prediction in multilayer network [69].

developed an analytical model for the admissibility region that uses the NN-deep RL algorithm to maximize the revenue of the infrastructure provider. In this model, an agent interacts with the environment and makes a decision at a given state, and an estimated reward is then given for each action. The reward in this case is revenue. Upon receiving a request, two NN algorithms predict the revenue for each state when the selected action is either accepted or rejected and then make decisions based on the predicted value. The performance of the proposed algorithm is close to the optimal point under a wide range of configurations, and this algorithm substantially outperforms the native approaches, such as smart heuristics with fast convergence. This algorithm can also be scaled to a large scenario and may prove useful in practical settings.

A summary of recent works on the ML-based predictive model can be found in Table 3 along with the issues and advantages of their proposed algorithms.

VI. ML-BASED IDS IN A COMMUNICATION NETWORK

The mass dependencies on computers and networks, especially those applications that require strict protection applications such as banking, securities, and private cloud data, only increase the vulnerability of users to security and privacy threats. Guaranteeing security in such a complex technological environment presents a huge challenge that needs to be tackled intelligently. IDS is an important tool for ensuring the security of the network information system. Security mechanisms, including wired equivalent protection and WiFi protected access, have been mainly used to secure and protect networks. However, these mechanisms demonstrate many flaws when exposed to threats, such as Denial-of-Service (DoS), network discovery, and brute force attacks [70]. Intrusion can also lead to huge financial losses and compromise critical infrastructure. Some real-life intrusions in the network are listed in Table 4.

To protect the network from such vulnerabilities, researchers have developed a state-of-the-art IDS that integrates ML algorithms. Given that intrusion detection can be considered a classification problem, ML is viewed as a promising IDS candidate in the network. ML-based IDS provides a learning-based system that classifies possible attacks based on the behavior of the incoming packet. The advantages of ML-based IDS over conventional signature-based IDS are as follows [71]:

- flexible rather than rule based;
- low computational load;

TABLE 4. Real-life Network Intrusions.

Year	Network Intrusion Descriptions
2007	Distributed Denial-of-Service attack launched against Estonian website [75]
2008	Amazon started receiving increasing number of authentication request that slows down the server [76]
2013	DDoS attacked Dropbox with substantial loss of service for more than 15 hours affecting all users across the globe [77]
2014	Suspected DDoS hit Facebook [78]
2018	Top three banks in the Netherlands was targeted in multiple cyber-attacks, blocking access to the website and internet banking services [79]
2019	One of the world's largest aluminum companies, Norsk Hydro, was attacked by a new variant of ransomware called LockerGoga, forcing some of its automated production lines in Europe and the US to shut down [80]
2019	The Labor Party suffered a DDoS attack that affecting the party website, online campaigning tools, and platforms after they were flooded with millions of requests [81]

- ability to detect novel attacks;
- ability to capture complex properties of the attack behavior; and
- clustering ML-based IDS does not require updates on the dataset.

TABLE 5. Common Attacks in the Network.

Intrusion	Descriptions
Denial-of-Service (DoS)	The administrative server will be flooded with many service request and will fail to provide services to other legitimate users
Scanning attack (Probe)	Attacker attempt to locate the target systems in the network and subsequently exploit known vulnerabilities
Remote-to-local (R2L)	Attacker use to gain local access to the vulnerable machine provided the attacker can send packets to the victim machine over a network
User-to-root (U2R)	Exploits are used to gain the root access to a machine by an unprivileged local user

ML-based IDS aims to provide a general representation of known attacks from historical data [72]. Some of the known intrusions in the network are summarized in Table 5.

IDS can be categorized into a host- and network-based IDS, both of which can be further categorized into signature-, anomaly-, and hybrid-based IDS. Anomaly-based IDS learns the network behavior under normal operations and classifies any abnormality as an intrusion. Signature-based IDS learns from the historical dataset of a known intrusion and classifies a similar occurrence as a network intrusion. However, signature-based IDS achieves a favorable detection accuracy only for well-known attacks. Moreover, the administrator needs to update the IDS database regularly, and this IDS is prone to false alarms whenever new legitimate traffic enters the network domain.

Anomaly-based IDS can be further categorized into statistical techniques, ML-based techniques, and finite state machine-based (FSM) techniques [73]. FSM produces a behavioral model comprising states, transitions, and actions. Anomaly-ML-based IDS can discover zero-day attacks or intrusions that are previously unknown. However, anomaly-ML-based IDS suffers from a high false-positive because of the limitation of ML algorithms in accurately distinguishing normal from intrusion behavior.

Similar to any ML algorithm, data are essential for IDS. Computer network security data can usually be obtained either directly or by using an existing public dataset. Direct access is one method of acquiring cyber data either by simulations or using a testbed. The required network packet data can be captured through Wireshark or Win Dump. This direct approach is flexible and straightforward depending on the preferences of the researcher. However, this approach is only suitable for collecting short-term and limited amounts of data on a network with a limited scale. When trying to obtain long-term and large amounts of data, the cost of data collection will increase proportionally. Therefore, using an existing public dataset can shorten the data collection time and subsequently improve research efficiency [74]. However, feature selection needs to be performed because these datasets usually contain massive amounts of data, and some features may have a low correlation for ML algorithms. Moreover, some datasets may be outdated, such as the KDD-Cup99 and NSL-KDD, thereby introducing challenges in the detection of the newest attacks.

A. PUBLICLY AVAILABLE DATASET FOR IDS

The DARPA intrusion detection dataset is collected and published by The Cyber System and Technology Group of the MIT Lincoln Laboratory for evaluating IDS [82]. The latest DARPA dataset is the 2000 DARPA intrusion detection scenario-specific dataset that includes LLDOS 1.0, LLDOS 2.0.2, and Windows NT attack scenario data.

The KDD CUP 99 dataset, which is based on the DARPA 1998 dataset [83], is one of the most used training datasets in the literature that contains 4,900,000 replicated attacks, 22 attack types, and 41 fixed feature attributes.

The NSL-KDD dataset is a new version of the KDD CUP 99 dataset that addresses some limitations of its predecessor [84]. This dataset removes unnecessary records and duplicates from the training data and offers a highly homogenous distribution by ensuring that the number of records in the training sets is proportionally distributed [83].

The CICIDS-2017 dataset, which was created in 2017, includes real-world attacks that are recorded during the year of its introduction. This dataset was created by analyzing network traffic by using information from timestamps, source and destination IPs, source and destination ports, protocols, and attacks [85]. This dataset comprises 86 features, complete with network and traffic structure, tagged data, recorded network traffic, and protocols of frequent attacks that are distributed proportionally.

CSE-CICIDS-2018 was introduced in 2018/2019 by the Canadian Institute for Cybersecurity [86]. As an enhanced version of the previous CICIDS-2017 dataset, CSE-CICIDS-2018 contains a limited number of duplicated data, excludes uncertain data, and can be exported in CSV format, thereby making this dataset ready for use without pre-processing.

B. IMBALANCE DATASET

One problem with publicly available datasets is that they contain data with enormous sizes. Given that these datasets capture network activity for as long as several weeks, most ML algorithms that utilize shallow learning methods, such as KNN, SVM, and SOM, may suffer from long training times [87], [88]. Shallow learning algorithms heavily depend on feature engineering and feature selection and demonstrate poor performance in detecting unlabeled network attacks with high false alarm rates. These algorithms also cannot effectively classify large-scale data in actual complex network application environments.

In addition, most datasets containing different types of attack traffic are imbalanced. Fig. 15 shows the percentage portion between normal and DoS attack class type between well-known public intrusion dataset. Such imbalance prevents traditional classifiers from achieving high detection rates. These classifiers tend to favor the class with the highest volume in a dataset [88]. Meanwhile, minority class types usually have low prediction and detection rates. This issue is generally unfavorable given that the other intrusion types in the dataset are equally harmful to the network. Intruders may take advantage of this loophole and focus on minority attack types. While other attack types such as bot, infiltration, brute force, and SQL injection attacks only account for less than 7% of the public dataset [83].

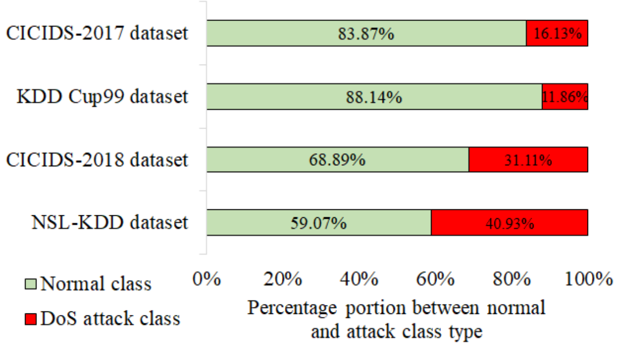


FIGURE 15. Percentage portion comparison between different public intrusion datasets.

Several methods have been used to deal with imbalanced datasets. One of these methods is resampling, which can be divided into the following:

- oversampling, which generates samples of the minority class;
- undersampling, which drops the sample of the majority class;
- hybrid sampling, which combines oversampling and undersampling;
- exploiting ensemble-based algorithms to help alleviate the influence of imbalanced class distribution; and
- using loss functions in DL algorithms.

Several studies have recently attempted to overcome such imbalance. For instance, Yang *et al.* [88] proposed the hybrid supervised and DNN adversarial variational auto-encoder with regularization (SAVAER) approach, which can accurately detect various network attacks, thereby making this approach suitable for new networks. The decoder of SAVAER is used to synthesize low-frequency and unknown attack samples of a specific label, thereby increasing the diversity of training samples and balancing the training dataset. As a result, the detection rate for low-frequency and unknown attacks is improved. SAVAER has been compared with both DNN and DT. In the NSL-KDD dataset, the minority attack class types are the U2R and R2L attacks. Results show that SAVAER has a 44.5% detection accuracy for U2R attacks, whereas DT and DNN only report detection accuracies of 8.5% and 5 %, respectively. Meanwhile, for R2L attacks, SAVAER, DT, and DNN obtain detection accuracies of 53.59%, 7.12%, and 7.66%, respectively. Therefore, SAVAER can successfully improve the detection of the minority attack class in the NSL-KDD dataset.

Karatas *et al.* [83] argued that looking at the overall accuracy of the ML-based IDS does not yield precise comparisons due to the imbalanced distribution of attacks in the dataset. Instead, the accuracies related to each attack type should be examined separately. Accordingly, they attempt to remove the effect of asymmetry between classes in the dataset by improving the average accuracy of the system. The imbalance ratio in the dataset is reduced by using a synthetic data generation model called synthetic minority oversampling technique (SMOTE) as depicted in Fig. 16. The SMOTE function creates new samples by considering the differences between the feature vectors and their nearest neighbor and by multiplying the difference by a random number between 0 and 1. After running this function, the imbalance ratio in the newly transformed dataset is reduced from 53887 to 9.98, which is deemed acceptable. However, the sampling model increases the dataset size by 17%, thereby extending the training time of the system. Nevertheless, the newly transformed dataset obtained by the SMOTE function demonstrates a 72.35% improvement in detection accuracy for three minority attacks, namely, brute force, infiltration, and SQL injection attacks.

Yu and Bian [89] proposed an intrusion detection method based on few-shot learning (FSL). FSL is one of the solutions when only a limited amount of training data is available [90]. This method aims to learn from a small amount of labeled data. FSL is one of the algorithms that can effectively solve the problem of limited network intrusion detection data. However, FSL requires a balanced dataset. To ensure that the dataset is balanced for each training session, the same number of samples from each attack class type is chosen and sampled by N times in sequence according to the sampling order. This work focuses on two training phases, namely, binary classification, where the dataset is categorized into normal and attack type classes, and multi-class classification. For binary classification, 100 samples are chosen from each class and sampled five times. Meanwhile, for multi-class classification,

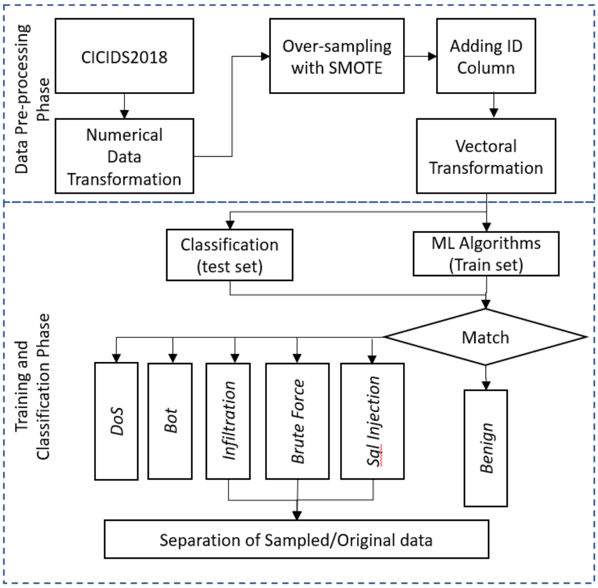

Data Pre-processing Phase
CICIDS2018
Numerical Data Transformation
Over-sampling with SMOTE
Adding ID Column
Vectoral Transformation
Training and Classification Phase
Classification (test set)
ML Algorithms (Train set)
Match
DoS
Bot
Infiltration
Brute Force
Sql Injection
Benign
Separation of Sampled/Original data

FIGURE 16. **Workflow of SMOTE function proposed in [83].**

50 samples are selected for each class and sampled 10 times. Experiment results show that the resampling works do not affect the performance of the IDS. FSL also outperforms the other ML algorithms, such as NB, NB-Tree, RF, and SVM. The proposed scheme achieves a 92.34% accuracy with only 1% of the NSL-KDD and UNSW-NB15 dataset. Meanwhile, in multi-class classification, FSL achieves a 92.33% accuracy with only less than 2% of the KDDTrain+ dataset.

Gao *et al.* [91] investigated the imbalanced dataset problem by using the ensemble learning model. Ensemble learning integrates the advantages of each ML algorithm for different attack classes to achieve optimal results. All classification algorithms are initially trained via cross-validation by using the training data, and the algorithm with the highest accuracy and operation performance is selected for voting. Afterward, each algorithm is boosted via feature selection, unbalanced sampling, adding class weights, and multi-layer detection to further improve the detection accuracy. The class with the highest number of votes is selected as the final prediction of traffic class type. Experimental results prove that the proposed ensemble voting approach achieves the highest accuracy of 85.2%, whereas the multi-tree and DNN models only achieve accuracies of 84.23% and 81.61%, respectively.

C. FEATURE ENGINEERING ISSUES IN DATASET

The high-dimensional input features of publicly available IDS datasets pose one critical challenge that needs to be addressed. Nagaraja *et al.* [92] proposed a Gaussian distance function to reduce the dimension of the original input dataset into a new transformation space. By using the KDD and NSL-KDD datasets, the original 41 attributes are reduced to 35 after feature transformation. By dimensionally reducing the dataset, the precision value for the minority attack class is improved from 58% to 68% for U2R attacks and from 97% to 98% for R2L attacks.

Kasongo *et al.* [70] proposed a DL-based IDS that uses feed-forward DNN (FFDNN) coupled with a filter-based feature selection algorithm. Before using FFDNN for feature extraction, DNN has a learning rate of 0.05 with 41 features. After the filtering process, the number of features is reduced to 21, thereby reducing the learning rate to 0.02 and achieving an accuracy of 86.19% when using the NSL KDDTest+ dataset. Other ML algorithms, such as RF, SVM, and NB, achieve accuracies of 85.27%, 79.55%, and 75.51%, respectively, when using the same dataset. These results prove that feature extraction or reduction is essential to reduce the learning load of ML algorithms.

However, extracting features from a dataset is a challenging task. ML algorithms can achieve satisfactory detection levels when sufficient training data are available, and sophisticated hand-engineering features are built to achieve sufficient generality and to accurately detect both attack variants and novel types of attacks. However, with the emergence of DL, these hand-engineering features have been replaced with a trainable multi-layer network. Andresini *et al.* [93] exploited the DL feature by proposing a novel DNN architecture for training intrusion detection models. They combined supervised and unsupervised multi-channel feature learning to find the feature dependencies in both channels. In the supervised stage, two sets of autoencoders for normal and attack flows are separately learned. The autoencoders in the first set are trained on normal samples, can contribute to the recovery of denoised normal samples, and can detect attack samples as anomalies. Meanwhile, the autoencoders in the second set perform the same process for attack samples. The multi-channel parametric convolution is then adopted in the supervised stage to learn the effect of each channel. The idea is to exploit the possible existing patterns among channels to improve intrusion detection performance. The proposed approach outperforms the other ML algorithms, including NN, ANN, CNN, and anatomically CNN, when using the CICIDS 2017 test set with 97.9% accuracy.

To address the misclassification issues in IDS, Su *et al.* [94] proposed BAT-MC, a novel methodology for IDS that integrates the Bat and DL algorithms. The Bat algorithm combines the bidirectional long-short-term memory (BLSTM) with the attention mechanism. The network traffic is repeatedly collected at fixed time intervals to generate a network traffic matrix. Multiple convolutional layers then pre-process the data. The BLSTM layer extracts the features of the traffic bytes of each packet. The attention mechanism is then used to perform feature learning on the sequence data comprising the packet vector to obtain fine-grained features. The proposed BAT-MC method can achieve an 84.25 % detection accuracy, which is approximately 4.12% and 2.96% higher than those of existing CNN and RNN models when using the NSL-KDD dataset.

D. CHOSSING THE RIGHT ML ALGORITHMS FOR IDS

No single ML algorithm can be considered superior over others when solving specific problems. Such superiority all comes down to the core of the problems, the availability of data, the volume of datasets, and other considerations, including the overhead cost or transparency. Each algorithm has its advantages and disadvantages. While some algorithms may perform well on one type of attack, they may demonstrate a poor performance on other attack types.

Ahmad *et al.* [95] compared the performance of SVM, RF, and extreme learning machine (ELM) for IDS using the NSL-KDD dataset. ELM is a feedforward NN, but it works differently from a standard NN as it does not require gradient-based backpropagation to work. The results from [95] shows that SVM performs poorly when using a huge dataset, thereby making this algorithm unsuitable for IDS given that the dataset is large by nature. Meanwhile, the RF algorithm has a low real-time prediction speed given that this algorithm comprises multiple trees that may require much time to develop. The ELM algorithm outperforms the other algorithms in terms of accuracy, precision, and recall rate on full data samples with 65,535 records of activities. SVM shows the best performance but only when using a quarter of the data samples, thereby proving the claim that SVM only performs best in small datasets.

Recent works, such as Andresini *et al.* [93] and Su *et al.* [94], have exploited DL algorithms to reduce the cost of performing features engineering on a dataset, which is particularly challenging for datasets with a massive volume. However, DL algorithms require a longer training time compared with other ML algorithms [91], thereby resulting in long detection delays in the practical application scenario of a broadband network and subsequently affecting the response time of attack detection. Although DL can handle high data throughput, improving accuracy and reducing the false-positive alarm rate remain crucial given the ever-growing size of datasets used in IDS research [70].

Otoum *et al.* [10] comprehensively analyzed the performance of ML algorithms that apply DL-based solutions for IDS systems in wireless sensor networks (WSN) with KDD'99 as the attack dataset. Analysis results show that the DL- and ML-based IDSs have accuracies of 99.91% and 99.83%, respectively. However, the ML-based IDS framework has approximately half the detection time of the DL-based IDS framework (31.4 s vs. 17.1 s, with testing times of 1.62 s and 0.86 s). However, ML-based IDS requires features engineering, which can be complex and time consuming.

E. MALWARE ATTACKS IN MOBILE DEVICES

Mobile devices are vulnerable to attacks, such as malware attacks. For instance, devices operating on Android, which is among the most popular open-source operating systems (OS) in the world, have accounted for 54% of malware variants reported in 2017 [96] Third-party application stores have also been reported to host 99.9% of the available malicious applications [96]. To address the vulnerabilities of mobile devices, researchers have developed an ML-based Android OS malware detection approach. For instance, Ananya *et al.* [96] proposed a novel feature selection method called selection of relevant attributes for improving locally extracted features using classical feature selectors (SAILS), which can improve the performance of classifiers, including RF, LR, classification and regression trees, XGBoost, and DNN, compared with conventional feature selection methods. Their evaluation results show that SAILS achieves an accuracy improvement of up to 95% compared with conventional feature extraction methods, including mutual information, distinguishing feature selector, and Galavotti–Sebastiani–Simi. However, when SAILS is tested against adversarial attacks, its accuracy decreases to as low as 24.79%. These results suggest that hackers can still bypass detection when the classifier blind spot is exploited, and this challenge needs to be addressed to further improve the security of mobile devices.

Taheri *et al.* [97] proposed another malware detection method that uses Hamming distance to classify samples into benign and malware samples. The conceptual workflow of this proposed scheme is depicted in Fig 17. First, the static features of the data samples are selected from the dataset, and the RF feature selection algorithm is used to select a certain percentage of features between 10% to 100%. The selected features are then converted into vectors and further converted into binary vectors. Second, the ML model is generated by using the proposed classification detection algorithm based on Hamming distance. This model achieves a malware detection accuracy of up to 99%, which is comparable with that of existing state-of-the-art solutions.

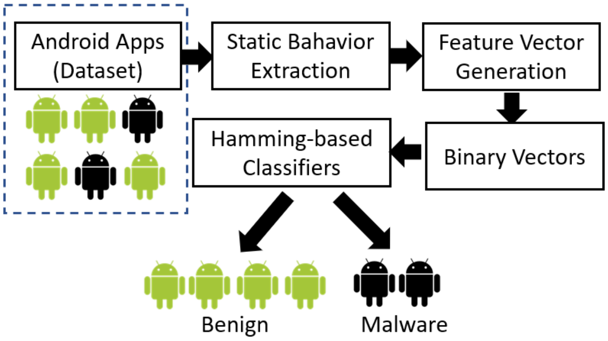

Android Apps (Dataset)
Static Bahavior Extraction
Feature Vector Generation
Hamming-based Classifiers
Binary Vectors
Benign
Malware

FIGURE 17. Workflow of Hamming distance classification proposed by [97].

Table 6 summarizes the recent works on the application of ML algorithms for IDS with their issues and their accuracies to detect intrusions.

VII. ML FOR IMPROVING ROUTING DECISIONS IN COMMUNICATION NETWORKS

Network traffic routing is one of the fundamentals in networking where a path is selected for packet transmission.

TABLE 6. Summary of recent ML-based IDS with their Previous Limitation of ML-based IDS, Training Datasets, ML Algorithms and Detection Accuracies.

Authors	Year	Previous Limitation of ML-based IDS	Dataset	ML Algorithm	Accuracy
Kasongo et al. [70]	2020	The accuracy of existing methodologies for IDS using ML is profoundly affected when the feature space grows	NSL-KDD	Feed-forward DNN	86.19%
Karatas et al. [83]	2020	Accuracy related to each attack type should be examined separately	CICIDS 2018	RF	99.35%
Yang et al. [88]	2020	Difficult for traditional classifiers to achieve high detection rates on imbalanced data sets	NSL-KDD	SAVAER-DNN	89.36%
Yu et al. [89]	2020	Unbalanced dataset and the small sample size of minority categories	NSL-KDD	FSL	92.33%
Andresini et al . [93]	2020	Sophisticated hand-engineering features are required for the dataset	CICIDS 2017	DNN	97.90%
Su et al. [94]	2020	Traditional ML-based IDS have relatively low accuracy and rely heavily on the manual design of traffic features	NSL-KDD	BAT-MC	84.25%
Taheri et al. [97]	2020	The accuracy of existing methodologies for IDS using ML is profoundly affected when the feature space grows	Drebin, Genome and Contagio	Modified KNN	99.00%
Ananya et al. [96]	2020	Classical feature selection algorithm yields low detection accuracy due to the presence of noisy attributes	Drebin	LR, RF, DNN, XGBoost, and CART	99.00%
Otoum et al. [10]	2019	The performance difference of ML and DL-based IDS in WSN	KDD-99	RBM	99.00%
Gao et al. [91]	2019	Each algorithm has its advantages and disadvantages	NSL-KDD	Ensemble Learning	85.20%
Ahmad et al. [95]	2018	The performance of SVM decreases with extensive data, and RF algorithm reduces real-time prediction speed	KDD-99	ELM	99.50%

With proper routing management, the route with cost minimization and fulfils the QoS requirements can be determined. Traffic routing using ML approaches is a challenging task that must be able to cope with complex and dynamic topologies, different types of traffic, and unique QoS requirements. The input and output of ML algorithms for the routing optimization problem can be described as traffic and route matrices [98]. ML algorithms should learn the correlation between traffic inputs and link conditions to predict or determine a path for the incoming traffic. Recent studies that attempt to improve routing decisions in a network are mostly NN based, such as in [22], [24], [99]–[101], followed by works that adopt RL, such as in [34], [100], [102]. Those recent studies that exploit other ML algorithms are further elaborated in this section.

A. DL-BASED ROUTING ALGORITHM

Sensors are intensively deployed in mobile heterogeneous wireless sensor networks (MHWSN) to improve data monitoring accuracy [103]. However, intensive deployment nodes can cause multiple nodes to perceive the same anomaly, thereby making the data highly redundant. To effectively address these redundancy issues, a data fusion algorithm based on an ELM optimized by the Bat algorithm for MHWSNs is proposed in [99]. ELM is another type of single hidden layer feedforward-NN. Given that ELM consists of only one hidden node layer, the output weight and thresholds are calculated via one-step operations, thereby increasing the learning speed of ELM by several thousand times compared with back-propagation (BP) NN, RBF NN, and SVM. By contrast, the Bat algorithm was inspired by the echolocation ability of bats that provides them with strong global search abilities. In this work, the Bat algorithm optimizes the input learning weight and threshold of the ELM algorithm, and only the optimal nodes are chosen to be transmitted. Simulation results prove that the BAT-ELM-based data fusion algorithm can effectively reduce network traffic, conserve network energy, improve networking efficiency, and significantly extend network lifetime. Compared with other protocols such as the stable election protocol, BP NN, and ELM-based NN, the proposed BAT-ELM algorithm has a higher node survival rate, which reaches 87% at the 400th iteration. Meanwhile, BP NN and ELM-based NN have node survival rates of 55.0% and 51.7%, respectively. The BAT-ELEM algorithm also has the highest node reduction and the best load performance among the compared algorithms. Combining the ML algorithm with other optimization algorithms, such as the Bat algorithm in this case, can further improve the overall efficiency.

Building a mathematical model that accurately describes the behavior of WSNs in a complex environment is a challenging task [103]. The work in [24] presents a case study where ML classifiers are hybridized to develop the multi-criteria Topsis-based ensemble (MCTOPE) framework. Instead of merely evaluating the accuracy of ML algorithms, this framework generates scores based on the diversity of classifiers, errors, accuracy, and area under the ROC curve of the classifier. The validity of the MCTOPE framework is tested by using six datasets from the UCI machine learning repository, and results prove that the ensemble SVM and NN classifiers are superior over single ML-based classifiers.

The next-generation wireless network (NGWN) is an interface of network services and operations that can support the access of multiple standards, such as 5G, Wi-Fi, and cognitive radio networks. However, the traffic in the current communication infrastructure is rapidly increasing to the extent that the router speed may be unable to accommodate such traffic.

Depending on conventional routing schemes that are purely based on standard rules and have limited computing capacity cannot satisfy and serve the real-time load balance requests of the NWGN [22]. Accordingly, Yao *et al.* [22] proposed a load balance routing scheme based on NN to predict the network queue status, which is one of the metrics used for making intelligent routing decisions. The proposed algorithm is then compared with shortest-path-based algorithms, such as Bellman–Ford (BF) and its Queue-Utilization variant, in terms of throughput and delay. Results show that the proposed algorithm achieves a higher throughput yet suffers a 20% longer delay compared with the BF algorithm. However, the proposed algorithm can predict the next-hop path with the lowest buffer and alleviate the load balancing issue.

Routing in the opportunistic Internet of Things network (OppIoT) is an incredibly challenging task because the network is intermittently connected and end-to-end paths from the source to the destination are almost non-existent due to the absence of a fixed infrastructure. With limited information, designing a routing protocol with a high message delivery success rate is considered ambitious. A routing strategy called epidemic routing will flood the network with messages. Despite having a higher probability of message delivery, this strategy has a high overhead load. Simulation results show that the delivery probability of this strategy is merely 38.54%. An ML-based solution called ML-based probabilistic routing protocol using the history of encounters and transitivity (MLProph) adopts a binary classification of the delivered or undelivered message to train on, thereby resulting in a class imbalance problem. To improve routing in OppIoT, Vashishth *et al.* [100] utilized cascade learning, which is a form of ensemble-based ML that combines logistic regression with NN classifiers. The logistic algorithm initially generates two probabilities that are either delivered or undelivered by using MLProph as the input. The probabilities from the regression model are fed as inputs into the NN classifiers to generate a delivery probability value from the learning solutions. The proposed algorithm outperforms the existing ML-based protocols, including MLProph, KNNR, history-based prediction routing (HBPR), and ProPHET, in terms of message delivery probability, average hop count, number of packets dropped, and network overhead ratio.

With the exploding traffic volume and complex environments in wired grid networks, controlling network traffic becomes an increasingly complex task when designing a routing strategy. The conventional routing methods are incapable of dealing with such complexity, and using fixed metrics to determine a routing protocol cannot cope with the complex environment. Tensor is a multi-dimensional matrix that provides a very concise mathematical framework for arranging the values of various parameters. A study in [101] proposed tensor-based deep belief architectures (TBDA), which uses tensor as an input in training an NN algorithm. The traffic patterns from the edge router are fed to TDBA, and a path to all edge routers is subsequently constructed. All paths are then attached to the headers of the corresponding packets, and the router simply forwards these packets according to the labeled paths. TDBA outperforms the OSPF protocol with a zero packet loss rate. The average delay per hop for TDBA remains constant, whereas that for OSPF gradually increases over time.

In a multi-domain optical network, having a distributed collaborative routing where each domain has its own controller can improve domain privacy but will create a complex signaling problem for inter-domain routing. Meanwhile, a centralized routing system simplifies the signaling yet compromises domain privacy. To overcome these privacy and complexity issues, Zhong *et al.* [104] proposed a data analytical method that learns historical route trajectories and trains a DL model that can directly return a feasible inter-domain route upon request. Training data, such as traffic requests, historical routes, and inter-domain link capacities, are publicly available. However, the complicated relationship among these data is deeply hidden inside the layers of the NN, thereby preserving domain privacy. The public global information in multi-domain networks is collected from a traffic engineering database and is fed to the NN for training purposes. Several local paths in each domain are computed, and these local paths comprise an end-to-end inter-domain path trajectory. Compared with the backward recursive path computation element-based computation approach for assigning the end-to-end path, the signaling volume of the proposed scheme is reduced with 98% prediction accuracy.

Li *et al.* [105] proposed an NN approach for optical circuit switching networks with fixed-alternate routing. The ELM framework is used to improve the training of the ML algorithm. This approach uses the enhanced ELM framework that adopts a random-search-based selection phase to determine those hidden nodes that significantly reduce the estimation error. As a result, the number of ANN hidden nodes is significantly reduced, and the estimation accuracy is improved. The ELM framework also rapidly estimates the blocking probability for all paths and recommends the best path with the lowest blocking probability to the network management system. Moreover, the enhanced ELM provides highly accurate blocking probability estimates by reducing the required number of hidden nodes by one-third compared with the previous baseline ELM training algorithm.

In SDN, the bursty nature of packet traffic introduces load imbalance in a network. To address this problem, Yao *et al.* [106] proposed a pair of ML-aided load balance routing schemes that consider queue utilization (QU) to reduce packet loss ratio and to improve throughput for better load-balance routing. The QU for the next time slot is predicted by ANN algorithms to cope with the network congestion resulting from sudden traffic bursts. The predicted value is then used for intelligent routing decision making. The proposed scheme achieves a higher packet loss ratio and throughput yet with a 20% longer delay compared with the shortest path approach.

A study in [107] employed metaheuristic dynamic optical routing to address the over-provisioning problem in SDN that leads to reduced energy efficiency and high operational expenses. This approach uses ANN to forecast the traffic load, predict the tidal traffic variation, calculate the best resource allocation, and reduce energy consumption. The ANN was trained on a public dataset from Milan with traffic of voice data and a short messaging service. The effectiveness of this ML-based dynamic routing scheme is proven to match almost entirely the behavior of a network that performs an optical routing reconfiguration. The proposed scheme also yields an optimality gap exceeding 3%, whereas the static-based routing scheme reduces the optimality gap below 0.2%.

Fig. 18 summarizes the sample recent network issues that are proven feasible to be solved using the DL-based algorithm.

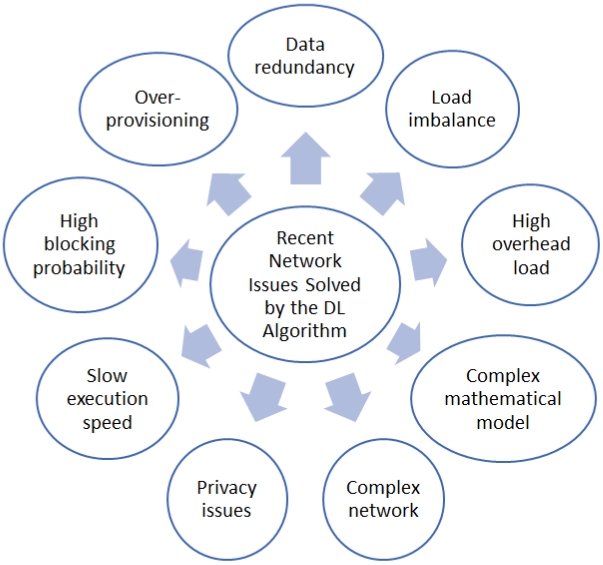


FIGURE 18. **Recent network issues solved by DL-based routing algorithm.**

B. RL-BASED ROUTING PROTOCOL

As discussed in Section III, RL employs an agent to learn the surrounding environment without supervision. RL uses a trial and error approach to learn the optimal action policy that maximizes the reward. Routing protocol via RL chooses an action that establishes a route from a specific path in the network to the destination, and the reward can be given in term of their delay, congestion level, packets loss rate, link reliability, retransmission count and many more. This process reiterates until the reward converges. For complex situations, a single agent may be insufficient to achieve a global optimization. In this case, multi-agent RL (MARL) employs multiple agents in the learning process, and each node exchanges local knowledge and decisions with other nodes in the network to achieve a better optimization. Nevertheless, this approach has high complexity and computation load that require attention.

Several works have used the self-learning RL-based algorithm to compute the most optimal path. Murudkar *et al.* [34] proposed the user specific-optimal capacity shortest path routing that uses RL to determine the resource-based optimum-capacity shortest path for a user between a source and destination pair in the 5G network. Given that the shortest path is not always the optimum one and fails to satisfy QoS requirements, this work considers the available capacity at the network nodes and the distance between a source and destination pair. By implementing Q-learning, the RL algorithm determines the shortest path while avoiding congested network nodes with high physical resource blocks (PRB) to satisfy throughput or bitrate requirements. If the PRB exceeds 70%, then the RL will classify the nodes as busy; otherwise, the RL classifies these nodes as available. Simulation results show that the proposed RL algorithm rapidly determines the shortest path with optimum capacity.

MARL routing algorithms can achieve better optimization yet incur a high communication overhead, slowly converge under dynamic networks, and lack QoS support. Reference [102] proposed an enhanced version of Q-routing, namely, the Q2-routing algorithm, which merges the existing wireless routing techniques and further enhances them by using the MARL domain for an ad-hoc wireless network. Q2-routing is a hybrid routing algorithm where the nodes make routing decisions by choosing the neighbor associated with the optimal Q-value for a given destination as the next hop. This algorithm is similar to Q-routing but with an additional modified reward function to satisfy the QoS requirements. During the learning process, only the training traffic is sent to obtain the Q-values on the available path until converging within a predetermined threshold. Afterward, the rate of sending learning traffic is sharply reduced, and the transmission of data traffic commences. The proposed Q2-routing algorithm outperforms the ad-hoc QoS-aware on-demand distance vector algorithm and can adapt to changes in network conditions.

C. RF AS A ROUTING ALGORITHM

In a circuit network, achieving an accurate timing estimation is difficult when the routing has not yet been performed. Moreover, performing the computations is usually expensive, and frequently evaluating optimization solutions is considered impractical. Given the lack of routing information, the over-pessimistic pre-routing prediction approach is adopted. However, this approach results in an over-design that subsequently wastes optimization time. To overcome this issue, Barboza *et al.* [108] proposed an ML-based pre-routing timing prediction that mostly avoids pessimism by using the RF algorithm. They compared this algorithm with lasso regression, ANN regression, and commercial-based estimation tools, and their experimental results show that the proposed pre-routing prediction achieves accuracy near the post-routing sign-off analysis. Moreover, compared with commercial estimation tools, this approach reduces the false positive rate by about two-thirds when reporting timing violations. RF also obtains the lowest MSE with the highest correlation. Using a commercial tool for the estimation is considered very pessimistic and large spread with more significant true negative error and false alarm. The proposed model has better accuracy than the ANN regression algorithm.

Recall that ML requires massive amounts of data to well-train and optimize the network. However, in real network deployment scenarios, having a perfect knowledge of the network is impossible. A study in [109] claims that in an elastic optical network, the complete information, including types of fiber and amplifiers, is not always known, thereby reducing the accuracy of the existing analytical model. Such lack of information may also result in the underutilization of network resources. Salani *et al.* [109] proposed the integration of RF-based estimation for routing and spectrum assignment to ensure QoT in an elastic optical network. All known network parameters, including traffic requests, alternative route configurations, and modulation formats, are obtained as inputs for the classifier. The output of these classifiers yields a probability that the light path configuration will satisfy a pre-determined threshold on the BER measured at the receiver. The learning process is iterative, where new information in the adjacent channels are fed into the classifier. Compared with the margined analytical model, the proposed scheme achieves up to 30% savings in the spectrum occupation.

D. OTHER SUPERVISED LEARNING ML-BASED ROUTING ALGORITHM

Circuit-switched networks are typically fixed route oriented, thereby limiting their routing performance due to inflexibility in route selection. One routing protocol in a circuit-switched network is the least loaded routing (LL) protocol. However, this protocol may have poor performance due to capacity overconsumption under high load situations, which affects its overall efficiency [110]. A novel online-based supervised NB classifier is then proposed in [110] to improve the performance of LL routing. The supervised NB classifier predicts the future circuit blocking probability between each node pair. After a service is either fulfilled or blocked, the network snapshot is stored as historical data for route selection in future service connections. The performance of the proposed scheme is compared with those of the least-load and short-path conventional routing protocols, and this scheme reports the lowest blocking probability, smallest number of extra hops, and lowest network capacity overconsumption.

Vashishth *et al.* [100] proposed a DL-based algorithm to improve the routing protocol in OppIoT, and their simulation results highlight the superiority of the proposed algorithm over other routing schemes. Meanwhile, the approach in [111] for routing in OppIoT utilizes Gaussian mixture model routing (GMMR), which combines the advantages of context-aware and context-free routing protocols. Specifically, context-free routing protocols utilize minimal network resources as they do not expend computational power in gathering and analyzing network information. While this approach increases the message delivery probability, the network may suffer from congestion and message dropping. By contrast, context-aware routing protocols gather knowledge about devices and network conditions to select the next best intermediate relay for a message yet require a high computation power. In GMMR, the trained GMM classifier creates clusters and assigns devices to each of these clusters. Afterward, the message is forwarded to every device belonging only to the same cluster as the message destination. This approach reduces the computational load by only involving the node within clusters and simultaneously increases the message delivery probability. Similar to the evaluation in [111], the performance of GMMR is compared with those of MLProph, ProPhet, KNNR, and HBPR. Simulation results show that GMMR outperforms these routing protocols in terms of average hop count, overhead ratio, delivery probability, and number of dropped messages.

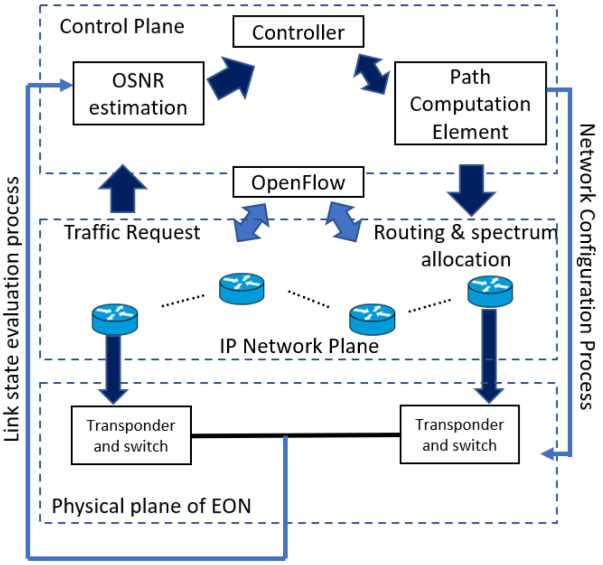


FIGURE 19. Link state-aware system model based on EON [112].

Zhou *et al.* [112] proposed a link state-aware routing strategy (LSA) that considers physical layer impairments to satisfy QoT requirements under different link states. As shown in Fig. 19, in the control plane, a link-state evaluation process is performed, followed by a network configuration process. In the link-state evaluation process, the physical signal is collected periodically in the physical plane of EON and then used to estimate the domain parameters, including chromatic dispersion and OSNR. Afterward, the link state is estimated by using the LightGBM algorithm, which is based on gradient-boosting DT. Results show that when the link in the network is degraded, the proposed LSA routing algorithm can still achieve an improved network throughput with a reduced traffic failure probability of 24% and bandwidth blocking probability of 10%.

Traffic classification is another approach used by the ML-based routing algorithm. Pasca *et al.* [113] proposed an application-aware multipath flow routing framework integrating ML in the SDN (AMPS), which evaluates the characteristics of each possible path based on the accessible parameters, including bandwidth and delay, and then assigns a path based on the QoS requirements. The paths are updated into the forwarding flow rule table. The data are collected by conducting experiments with 10 clients using different applications, such as Skype, Facebook, YouTube, and Dropbox. The collected data are fed into state-of-the-art classifiers, including NB, NB kernel estimation, DT, Bayesian network, and SVM. Numerical results show that the DT algorithm yields the highest classification accuracy of 98% compared with other classifiers. The AMPS-based routing also offers less jitter and high throughput for high-priority applications by choosing low-latency paths.

Recent studies that exploit the advantages of ML-based algorithms to improve network routing protocols have proven that ML-based routing can greatly improve network performance and efficiency compared with conventional routing schemes. The limitations of conventional routing protocols as described in recent papers are summarized in Fig. 20.

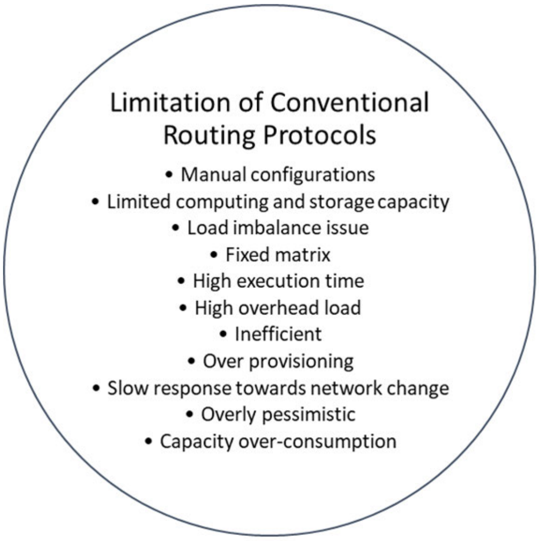


FIGURE 20. **Limitation of conventional routing protocols.**

The recent ML-based routing approaches proposed in the literature have promising applications in addressing complex network problems. However, with the superiority of ML-based routing, the computational load must also be considered if the technology is to be implemented in a real-world network environment. Several studies, such as those in [108] and [109], argue that collecting data for training the ML algorithm is not an easy task in practice. Furthermore, some of these studies have been based on assumptions that are not realistic enough to be implemented in a network. These are some of the challenges that need to be overcome in order to achieve the best trade-off between the best-performing ML model and computation complexity. Recent studies on ML-based routing and path assignments are summarized in Table 7.

VIII. ML ALGORITHM FOR IMPROVING QoS IN A NETWORK COMMUNICATION SYSTEM

Achieving a good QoS by managing network delay, jitter, bandwidth, and packet loss ratio is the main objective of any network provider. Having knowledge about the impact of network performance on user experience is also crucial given that such knowledge determines the success or failure of a service. Therefore, monitoring and controlling QoS parameters is essential to deliver high-quality services. However, with the increased traffic volume in a network, satisfying the QoS requirements of each incoming traffic becomes a challenge [114]. Using conventional algorithms for improving QoS parameters in a network may also be impractical due to network complexity. Therefore, an automated strategy should be developed to measure QoS as realistically as possible [115]. Researchers are still improving and developing novel algorithms, particularly ML-based algorithms, to maximize throughput, reduce delay, and comply with traffic QoS requirements.

A. THROUGHPUT MAXIMIZATION

Recent works, such as in [116], have exploited the advantages of ML-based models with an aim to improve network throughput. In a 5G wireless network, the conflict graph is widely considered an adequate representation of the underlying interference constraint in the network and a powerful tool for interference management. However, most studies that construct conflict graphs are based on accurate geographical distance information, which is not easy to collect in practice. Cao *et al.* [117] then proposed an accurate and practical ML-based approach for constructing a conflict graph. Specifically, the inter-user interference relations are constructed by analyzing the data collected from the network with minimum prior knowledge assumed for training the ANN algorithm. This approach mines large-scale uplink signal-to-interference-plus-noise ratio data and resource block allocation data that are readily accessible in a practical network. The ANN can automatically alleviate the influence of data fluctuations caused by rapid fading on the predicted model. From the constructed graph, the throughput maximization problem is decoupled into a user clustering subproblem and a subchannel allocation subproblem. The proposed Min k-Cut-based clustering algorithm splits the network into several clusters to further reduce the interference caused by spectrum reuse. A supplementary allocation algorithm is then developed to improve spectrum efficiency by fully utilizing the remaining unallocated subchannels. This approach improves the system efficiency by up to 125.19% and inevitably improves the network throughput.

An experiment in [116] has successfully proven that the RL-based algorithm can optimize the network-on-chip run time performance. This work presents a variety of

TABLE 7. Summary of recent ML-based Routing with their Previous Issues on Conventional Rule-based Method, ML Algorithms and Advantages of the Proposed ML-based Solution.

Authors	Year	Network	Previous Issues on Conventional Rule-based Methods	ML Algorithm	Advantages of the Proposed ML-based Solution
Cao et al. [99]	2020	MHWSN	Intensive deployment nodes can cause redundancy	ELM	Effectively reduce network traffic, network energy, network efficiency and significantly prolong network's lifetime
Zhou et al. [112]	2020	EON	Previous work does not consider the physical layer impairments	LightGBM	Reduce traffic failure probability by more than 20%
Yao et al. [22]	2019	NGWN	Traditional routing algorithms cannot always serve the NGWN effectively	DL	Improved packet loss ratio, throughput and delay
Vashishth et al. [111]	2019	OppIoT	Fixed source to destination paths are non-existent making routing very challenging	GMMR	Improved message delivery probability, average hop count, packets dropped and network overhead ratio
Murudkar et al. [34]	2019	5G	Challenging spectrum resource optimization	RL	The shortest path with optimum capacity is rapidly determined.
Zhong et al. [104]	2019	Multi-Domain Optical Network	A trade-off between complexity and privacy in the multi-domain network	DL	98% prediction accuracy is achieved for traffic request
Li et al. [105]	2019	Optical Circuit Switching Network	Estimating blocking probability is time-consuming for large scale network	ELM	Provides more accurate blocking probability estimates
Yao et al. [106]	2019	SDN	High computational complexity for QoS routing algorithm	DL	Achieved better global convergence and more effective optimization of the network
Barboza et al. [108]	2019	Circuit Network	Overly pessimistic pre-routing prediction	RF	Reached accuracy near post-routing sign-off analysis
Salani et al. [109]	2019	EON	Inaccessible of perfect transmission knowledge to train the ML model	RF	Achieved savings in spectrum occupation up to 30%
Li et al. [110]	2019	Circuit-switched Network	Fixed route oriented significantly limits the routing performance and flexibility	NB	Up to 90% of the time can be saved for the learning process which significantly accelerated the simulation studies
Hooda et al. [24]	2018	WSN	Building a perfect mathematical model that accurately describes the behavior of the system is a quite challenging task	Ensemble of SVM and NN	Achieve classification accuracy of 99.6%
Hendriks et al. [102]	2018	Ad-hoc Wireless Network	Traditional routing algorithms have a high overhead load to be used in an ad hoc environment	RL	Outperform well-known ad-hoc routing algorithms in dynamic environments under QoS constraints
Vashishth et al. [100]	2018	OppIoT	Context-free routing protocols suffer from high network overhead ratio and congestion	GMMR	Improved delivery probability, network overhead ratio, average hop count and number of messages dropped

RL-based algorithms, including Q-learning, state-action-reward-state-action (SARSA), and expected-SARSA algorithms, to keep record of the current network state with its corresponding reward, which is throughput in this case. Selecting routing algorithms that denote the action of RL will yield a reward based on the learned information with the goal of maximizing throughput. Experimental results show that with a 0.3 packet injection rate per node, the random routing strategies saturate due to deadlock. By contrast, the RL-based strategy delivers near-optimal choices across all states.

Azzouni *et al.* [118] introduced an ANN-based algorithm called NeuRoute to maximize throughput at minimum cost for the unicast dynamic routing of SDN. NeuRoute comprises three modules, namely, the traffic matrix estimator, traffic matrix predictor, and traffic routing unit. The traffic matrix estimator initially estimates the traffic matrix, the traffic matrix predictor takes the fixed size set of archived traffic matrices and input to predict the traffic matrix at the next cycle, and the traffic routing unit eventually selects the optimal routes based on the predicted traffic matrix. The traffic matrix estimator continuously gathers data from the network and feeds them into the traffic matrix predictor and traffic routing unit to adjust the weights and improve accuracy until the convergence point is reached. The model successfully selects the near-optimal path learned from the model with an estimated error of 0.05% and execution time of within 30 ms compared with the baseline heuristic approach, which has an execution time of 120 ms.

B. REDUCING NETWORK DELAY

To minimize the delay in a cognitive radio network (CRN), Pourpeighambar *et al.* [119] proposed a distributed cooperative multi-agent routing problem in a multi-hop CRN that is modeled by using a decentralized partially observable Markov decision process (DEC-POMDP). The goal of this approach is to minimize the end-to-end delay while keeping the interference to the primary user (PU) below a certain threshold. In CRN, the secondary users (SU) or cognitive users are allowed to access the licensed spectrum subject to the condition that the interference caused by the transmission of the SUs to the PU does not exceed a predefined threshold. Routing in CRN is incredibly challenging due to the stochastic behavior of PUs, which distinguishes this network from a traditional multi-hop wireless network. The main challenge here is to implement a routing protocol that is adaptive to the available spectrum windows in the CRN. DEC-POMDP was used in [120] to model the routing problem. Afterward, a gradient-based learning algorithm was implemented to solve this problem. Simulation results show that the proposed scheme maintains the end-to-end delay experienced by packets at a low level and outperforms the related approaches, including OPERA and the fictitious learning approach, in terms of interference control.

The existing QoS-aware routing schemes cannot be used for a front-haul centralized radio access network (C-RAN) because these schemes ignore frame-level queueing. To deal with queuing delay, Nakayama *et al.* [120] proposed a routing scheme that reduces the worst-case end-to-end delay of all front-haul flows and guarantees that all flows satisfy the latency requirements by using the Markov chain Monte Carlo algorithm. In the proposed work, the path computation element (PCE) collects information by using the IS-IS routing algorithm, and then the PCE generates a set of candidate paths for each front-haul flow by using the k-shortest path algorithm to determine those paths that satisfy the latency requirements. Afterward, the algorithm selects the paths by using the learned solution and determines whether these paths satisfy the constraints. The proposed work successfully reduces the delay of all flows below the latency requirements. However, when the shortest path approach is employed, the maximum delay exceeds the threshold due to queuing delays. These results prove that the ML-based algorithm can solve the queuing delay issues in C-RAN.

Stampa *et al.* [121] designed and evaluated a deep RL agent that can optimize routing according to a predefined target metric, which is the delay requirement in SDN. The deep RL model adapts automatically to the current traffic conditions and utilizes a tailored configuration to minimize the network delay. With the traffic matrix and bandwidth request as the states, path allocations as the action, and minimizing delays as the reward, the deep RL model can determine the optical behavior policy. This model also consistently computes the overall traffic intensities, and the delays are lower than the 100,000 randomly generated routing benchmark on average.

C. QUALITY OF EXPERIENCE (QoE) IMPROVEMENT

Several challenges in cloud-RAN (CRAN) need to be addressed in the application of unmanned aerial vehicles (UAV), including the effectiveness of tracking user behavior, caching, and resource management. Previous studies on UAV have only considered non-linear systems and assume that the users are static. Chen *et al.* [122] proposed a novel framework for deploying cache-enabled UAVs by incorporating a conceptor-based echo state network (ESN) to maximize the QoE of users and minimize the transmit power of UAVs at the same time. ESN is a branch of RNN that is used to predict the content request distribution and mobility patterns of users. ESN allows the cloud to split the behavior of each user into different patterns and learn these patterns independently to improve the accuracy of predictions. The ESN-based scheme successfully improves the average transmit power and QoE by 33.3% and 59.6%, respectively.

The related works discussed in this section prove the superiority of the ML algorithm in improving network QoS parameters, increasing network throughput, and minimizing delay across various networks. The ML-based algorithm also successfully addresses the QoS issues that are faced by conventional schemes, such as the shortest path or randomly generated path approach. The ML algorithm can also cope with network complexities, stringent delays, and throughput requirements.

IX. ML ALGORITHM FOR NETWORK RESOURCE MANAGEMENT

Network resource management refers to the process of managing and allocating the available resources for the networking process. An efficient resource management is achieved when the available network resources are fully utilized and able to achieve the desired QoS requirements [123]. In a communication network, the switches, routers, bandwidth, and spectrums are considered network resources. Traditional resource management approaches are typically static based, which, in the long run, will lead to an underutilization problem where the allocated resources, such as bandwidth, are higher than what is requested. Such inefficient resource allocation results in inevitable delays and poor network efficiency. Admission control and resource allocation are two broad categories that contribute to network resource management [18]. Admission control aims to optimize the utilization of resources by monitoring and managing resources in the network and accepts or rejects the incoming traffic based on network availability. Continuously accepting a new request increases the revenue of the network provider yet degrades the QoS of the existing service that violates the SLA. Therefore, admission control maximizes the number of accepted requests without violating the SLA. Resource allocation is a decision problem that manages resources, such as bandwidth, to achieve a long-term objective. By exploiting its advantages, the ML model can learn and predict resource management provisioning.

In a 5G SDN-based vehicular network, resource management is considered a complex and challenging objective that can facilitate the achievement of the expected outcome. However, one feature of SDN is its ability to extract network information from a centralized controller. Such information allows the detection of resource capacity and network requirements from the global perspective [16]. Moreover, extracting network information provides a considerable advantage in solving the resource allocation problem by using ML approaches, which can learn seamlessly from the available data. Tayyaba *et al.* [124] proposed a resource allocation

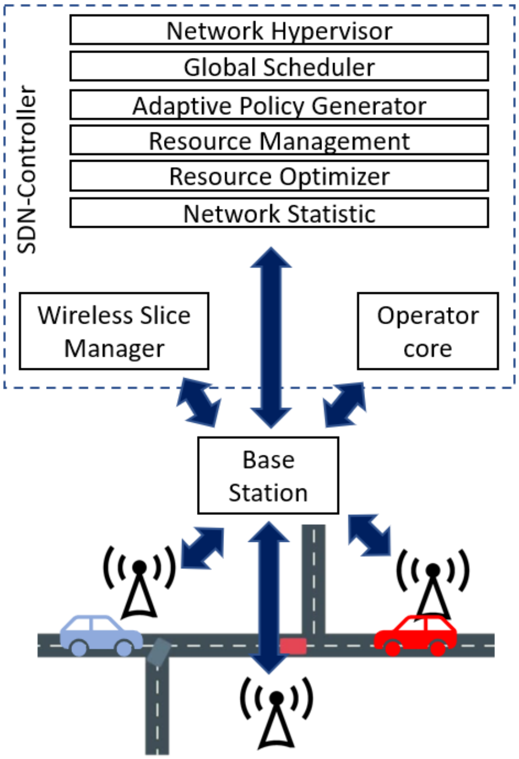


FIGURE 21. SDN-based 5G architecture for VANET service provisioning [124].

policy framework for an SDN-based vehicular network in the context of 5G connectivity as depicted in Fig. 21. In this work, the policy framework can optimize resource allocation according to the changing demands and dynamic nature of the vehicular network. Every flow request from vehicles is assigned with a priority based on the criticality of the application demand. The flows are then classified based on applications, such as road safety, infotainment applications, or comfort, by using an ML-based classifier. The training data are obtained by using a Mininet emulator to train classifiers, including LSTM, deep ANN, and CNN. Simulation results show that LSTM, CNN, and DNN achieve detection accuracies of 99.36%, 95%, and 92.58%, respectively, thereby proving that ML-based approaches successfully allocate network resources to high-priority applications with up to 99% accuracy.

Effective resource management is typically considered a challenge in mixed-integer nonlinear programming (MINLP). MINLP involves optimization problems with continuous and discrete variables and nonlinear functions in the objective function. A study in [125] shows that recent ML-based methods for addressing resource management problems in a wireless network require a tremendous amount of training samples and are unable to address constrained problems. When the network parameters change (task mismatch), ML-based approaches tend to demonstrate poor performance. To address this problem, Shen *et al.* [125] proposed the learning to optimize resource management (LORM) framework that can reduce the sample complexity and address the feasibility problem. LORM learns the optimal pruning policy in the branch-and-bound algorithm for MINLP by utilizing an efficient yet straightforward method called imitation learning. To address the task mismatch problem, a transfer learning method via self-imitation (LORM-TL) has been proposed. This approach can rapidly adapt a pre-trained ML model to the new task while requiring only few additional unlabelled training samples. The proposed resource management policy is compared with specialized state-of-the-art algorithms, including relaxed MINLP, iterative group sparse beamforming (GSBF), and branch-and-bound algorithms. The LORM algorithm successfully outperforms GSBF and relaxed MINLP and achieves a near-optimal performance within a running time that is twice shorter than that of GSBF.

The works in [124], [125] have proven the effectiveness of using ANN algorithms to learn from the environment, perform classification for resource network management, and accurately allocate network resources to high-priority applications. The combination of RL and DL has recently been proven as a promising alternative solution to various resource management problems in practical settings [126]. Meanwhile, NN-based algorithms encounter steady performance problems in terms of accuracy and convergence [127].

The recent works discussed in this paper have proven the superiority of ML algorithms in solving complex issues across most networks ranging from wired to wireless networks. Table 8 summarizes the recent works on the application of ML for satisfying QoS improvements and achieving resource management. The issues being faced in a network cannot be solved by using conventional methods, which are not adaptive and are not specifically built to solve complex problems without making unrealistic assumptions.

X. CHALLENGES AND FUTURE RESEARCH TRENDS

The emergence of ML-assisted solutions in networking seems promising given that ML can learn a complex system based on the fed historical or live data and make predictions based on these data. The survey of recent studies shows that ML algorithms have broad applications in solving various complex network problems. Despite the superiority of ML, many related challenges need to be addressed. One of these problems is winning the confidence of network providers to incorporate ML into their networks. This section discusses some challenges and potential directions in ML research that can help readers identify research gaps in the application of ML in networking.

A. HIGH COMPUTATIONAL LOAD AND TRADE-OFF WITH ML ACCURACIES

DT, RF, NB, and SVM are just some ML algorithms that are preferred by network administrators due to their simplicity and better interpretability compared with DL.

TABLE 8. Summary of recent ML-based QoS Improvement and Resource Management with their Previous Issues on Conventional Rule-based Method, ML Algorithms and Advantages of the Proposed ML-based Solution.

Authors	Year	Previous Issues on Conventional Rule-based Method	ML Algorithm	Advantages of the Proposed ML-based Solution
Cao et al. [117]	2020	Construction of conflict graphs is challenging to obtain in reality	ANN	Improves the aggregate throughput
Tayyaba et al. [124]	2020	Resource management in the vehicular environment is complex to satisfy the demand	LSTM	Outperformed the rest of classification techniques with promising results
Shen et al. [125]	2020	Exponential complexity of the recent global optimization algorithms	ANN	Achieves near-optimal performance with reduced in computation time
ChunKao et al. [117]	2019	Deadlock issues in NoC	RL	Successfully learn a near-optimal solution across different environment states
Azzouni et al. [118]	2017	Over-provisioning issues	ANN	The estimated error is less than 0.05
Pourpeighambar et al. [119]	2017	Routing in the CRNs is incredibly challenging due to the stochastic behavior of PUs	DEC-POMDP	Maintains the end-to-end delay experienced by the packets at a low level and substantially outperforms related work at interference control
Nakayama et al. [120]	2017	Existing QoS-aware routing scheme does not consider the frame-level queuing delay	Markov Chain Monte Carlo	Delay of all flows is below the threshold and satisfies the requirements
Stampa et al. [121]	2017	Limited routing optimization capabilities	Deep RL	Achieve better delay compared to the non-DRL agent routing scheme

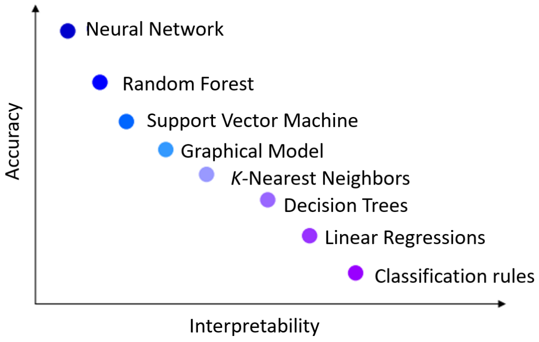


FIGURE 22. Accuracy versus interpretability [129].

The relationship between accuracy and interpretability is depicted in Fig. 22, which shows that a higher accuracy corresponds to a lower interpretability. One downside of the aforementioned ML algorithms is their instability. Specifically, a small change in the training dataset can result in a massive change in ML algorithms. These conventional ML algorithms may also be inapplicable in solving complex problems with a high-dimensional state and action space in large-scale environments. These algorithms also have low training speed and face overfitting issues that influence their effectiveness [51]. To overcome these problems, DL has emerged in the realm of networking. Previous studies have exploited DL to solve complex network issues. The number of applications with different QoS requirements has increased exponentially in recent years. As a result, the network depth and number of network parameters have also exploded. Given its multi-layer structure, DL is considered a practical approach for accurately extracting important information from raw data without requiring tedious feature extraction works, which represent the most time-consuming phase of conventional ML algorithms [128]. Recent advancements in GPU and hardware accelerators have resulted in the development of different DL-based solutions for various network problems. However, DL also has limitations, such as the demand for a significant amount of computation power, memory, energy, and resources. When DL algorithms are incorporated into a centralized network without resource constraints such as SDN, DL can be implemented with the aid of a resourceful computation platform. However, in distributed networks where the edge devices or sensors have limited storage and power (such as in the case of IoT), implementing DL presents an enormous challenge. Fig. 23 shows the run-time and accuracy comparison between different ML algorithms using a sample dataset. DL algorithm has the highest accuracy compared to the rest, but it has the highest run-time which correspondence to higher requirements of computational load. While DT algorithm has the lowest run-time, but has a slightly lower accuracy. Nonetheless, DT algorithms has better interpretability as depicted in Fig. 22. With that, network provider will need to look for the trade-off between computational load and accuracy for their system. Future research should then investigate the implementation of DL on resource-limited devices. However, it is important to note that different ML algorithms will behave differently depending on the training dataset.

B. DATA AVAILABILITY AND PRIVACY ISSUES

ML algorithms heavily rely on the availability of large quantities of performance monitoring data to learn and make predictions. One issue that needs to be considered by network providers is the availability of data for training ML algorithms to solve network problems. However, these data are often inaccessible due to privacy issues. This problem is particularly severe in cases where the required data for training ML algorithms are extracted from different domains or vendors. Proietti *et al.* [62] addressed the data privacy issues in multi-domain networks by using a multi-domain virtual topology to estimate the QoT for light path provisioning.

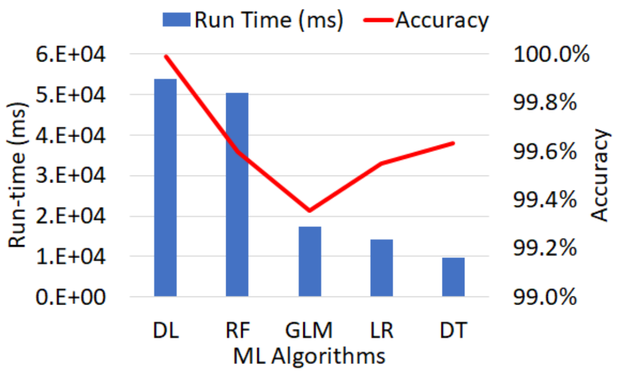


FIGURE 23. Run-time and accuracy comparison between different ML algorithms using a sample dataset.

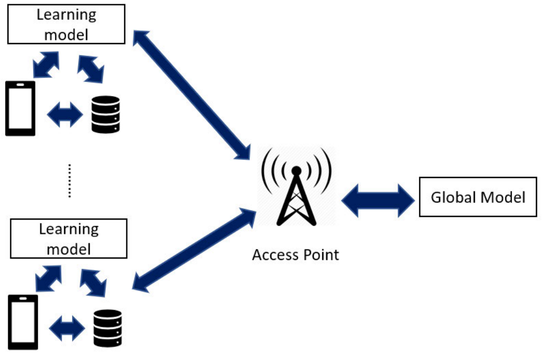


FIGURE 24. Basic framework of FL [130].

Federated learning (FL) has recently become popular in the field of networking. FL is a new distributed ML technique that usually operates in a wireless edge network. Each edge device contributes to the learning model by independently computing the gradient based on local training data. The basic workflow of FL is depicted in Fig. 24. First, users perform local computing by using their own data to minimize a predefined empirical risk function and then updates the trained weights to the access point (AP). Second, AP collects updates from users and consults the FL unit to produce an improved global model. The output from the FL is eventually redistributed to the users, who, in turn, will conduct further training by using the global model as reference [130]. Decoupling data acquisition and computation at the AP is viewed as a promising solution that can maintain the privacy of ML-based solutions in a network.

However, implementing FL in a network remains a challenge. For instance, wireless edge networks have limited bandwidth, and only a small portion of edge devices can be scheduled for updates in each iteration. Given the shared nature of the wireless medium, the transmission is subjected to interference and is therefore not guaranteed [130], [131]. The design challenges in FL, namely, resource and data challenges, should also be addressed. In terms of resources, edge devices have different computation power with limited storage. In terms of data, edge devices generate large and redundant raw data, and the FL paradigm needs to use these data to create meaningful solutions. Although one of the main advantages of FL is privacy preservation, the authors in [132] argued that during the training process, the data transmitted to the AP can still be reverse engineered by a malicious central server to reveal sensitive personal information.

The resource limitations of edge devices may also negatively affect the training of high computational learning algorithms, such as DL. Studies on the application of FL for solving network privacy issues are still in their infancy. Recent studies on FL, such as by Chen *et al.* [131] and Yang *et al.* [130], reveal scheduling policies as one of those issues that need to be tackled. With the future emergence of wireless edge networks, such as IoT, and 5G networks that involve a higher number of edge devices, FL can be seen as a very promising algorithm for improving user experience without invading their privacy.

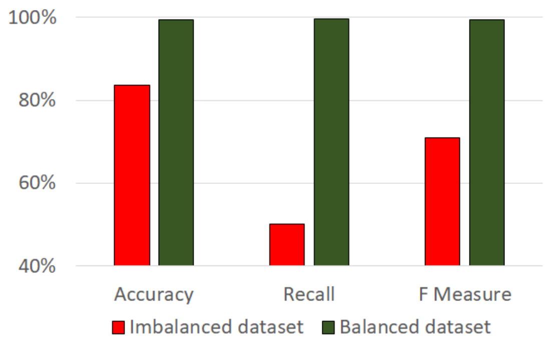


FIGURE 25. Performance comparison between balanced and unbalanced sample dataset.

C. IMBALANCED DATASET

Before feeding data for training ML algorithms, the dataset should be checked for any imbalanced problem. This problem has been widely reported in the literature, especially in the application of IDS in a network. In an imbalanced dataset, the feature inputs are favored over the other outputs in the dataset. For instance, in IDS, most of the publicly available intrusion datasets are heavily imbalanced toward the benign class with only a small percentage for the attack class output. As a result, the ML provides a prediction that is biased toward the benign class, thereby reducing the accuracy of ML-based IDS. Results in Fig. 25 shows the performance of an ML algorithm using a balanced and imbalanced sample dataset. The results show significant improvement of the ML algorithm when trained using balanced dataset algorithm in terms of accuracy, recall and F-measure compared to the ML algorithm trained using imbalanced dataset. Although several solutions have been proposed for this problem, including synthesizing low-frequency samples [88] and synthetic oversampling [83], previous studies are mostly limited to using publicly available datasets. Studies on an imbalanced dataset that is collected in real time remain limited, and a real-time sampling of datasets presents a challenge. Many edge devices, including IoT devices, sensors, or smartphones, only have a small number of data samples, thereby limiting the application of ML. Learning from small and imbalanced data may affect the performance of ML algorithms. Therefore, future research must investigate this problem. The amount of data is also expected to increase further in the future due to the increasing number of edge devices.

D. TESTBED FOR A REAL-WORLD ML IMPLEMENTATION STUDY

Most studies on the application of ML in networking are simulation based. Simulations are essential for evaluating the performance of ML-based schemes. However, simulations are restricted to several assumptions and are run in controlled environments. Future studies on the application of ML-based algorithms in a real-world environment should utilize real-time data to evaluate the performance of these algorithms and to serve as proof of concept before network providers can decide whether or not to implement ML in their networks. Several testbeds have been developed in the literature to evaluate the performance of ML-based schemes in real environments. For instance, Cheng *et al.* [133] studied indoor localization by using frequency modulation and digital video broadcasting terrestrial signals, Nithin *et al.* [134] used a face tracking robot to assess the performance of ML techniques, and Liu *et al.* [135] developed the first testbed for cognitive end-to-end optical service provisioning. However, only few studies have tested the performance of these approaches by using a lab-scale testbed. With the advancement of computing platforms and GPUs, the real-world implementation of ML can be evaluated on a testbed. In addition, software-defined radios or routers into which ML algorithms can be programmed can facilitate performance evaluations and can be considered a future trend in ML research.

E. HYBRID ML ALGORITHMS FOR DIFFERENT NETWORK APPLICATIONS

The data extracted from networks can be used for applications other than for training ML algorithms. For instance, when inspecting the incoming traffic in the ingress router, the input data can be used to perform IDS and to classify the traffic as either benign or attack. Afterward, by cascading with data from the network domain, IDS can be combined with different applications, including resource management, congestion control, or routing. The available data in the network can also be used to train ML algorithms for different applications. For instance, Choudhury *et al.* [64] proposed a hybrid ML model that predicts the traffic volume and optical performance of a new wavelength in multi-vendor environments. However, studies on hybrid ML algorithms are still in their infancy. Therefore, the hybridization of ML algorithms presents a promising direction for future research.

F. ADVANCEMENT OF 5G AND FUTURE 6G

5G is the next-generation mobile communication technology that aims to offer better network capacity and data rates compared with the previous LTE technology. 5G applications can be divided into enhanced mobile broadband (eMBB), ultra-reliable low-latency communication (URLLC), and massive machine-type communication (mMTC). Each of these applications faces a unique set of challenges. ML optimization has the potential to support 5G requirements. Wireless network virtualization (WNV) is expected to become one of the main trends in 5G systems that provides better QoE for users. Given that WNV relies on SDN, the programmability of SDN will introduce opportunities of applying autonomous and ML algorithms in a 5G environment.

For eMBB applications that offer high peak rates, the target throughput can reach up to 20 Gbps for downlink, which is 20 times higher than that in the previous LTE technology. A significant amount of spectrum resources [136], such as centimeter and millimeter waves, is needed to exploit the full potential of eMBB. Meanwhile, massive or large multiple-input multiple-output (MIMO) is essential to improve spectral efficiency in 5G [136]. ML can be also be used for channel or direction of arrival estimations in MIMO technology. Classification using ML can produce channel state information that facilitates the selection of optimal antenna indices [136], [137]. ML can also be used to predict future demand from users and to dynamically perform resource management, determine the topology setup, and identify a suitable bitrate based on connectivity performance to further enhance user experience [138]. Similar to other networks, 5G is also vulnerable to malicious activities. Therefore, ML-based anomaly detection in the wireless spectrum is crucial. ML-based IDS has been proven to be a promising solution for intrusion detection with outstanding accuracy. Future research should then focus on the detection of anomalies in the 5G wireless spectrum.

In the 5G network environment, a large amount of sensors, actuators, electronic appliances, drones, and smart devices are wirelessly connected to the Internet and to one another via mMTC [139]. These devices generate sporadic traffic among many geographically spread equipment, thereby introducing connection density and network energy efficiency issues. One promising solution to these connection density issues is incorporating an intelligent proactive caching [140]. Caching refers to the intelligent buffering of data at the nodes based on their demand rate with an aim to reduce the delay and power consumption in data routing. ML can also be exploited for processing, classifying, and manipulating content to improve the caching process in 5G IoT environments. Given the potential of ML, an intelligent caching of data at the base station allows a significant offloading of heavy traffic from the network backhaul. At the same time, the latency of popular and on-demand content can be reduced. The use of ML for proactive caching in the emerging big data era also presents a potential future research direction. Given the limited computational power and battery life of edge devices, various tasks are often performed at the base station or in the cloud. However, doing so will introduce several challenges, including limited capacity, especially in highly dense networks. An ML-assisted solution, such as incorporating FL and dynamic intelligent scheduling, can be viewed as a potential trend in 5G research.

Numerous network services, including healthcare, remote surgery, and mission-critical applications, will be made possible soon with the advancement of URLLC. However, URLLC requirements come with their own set of challenges. In terms of reliability, the expected packet error rates are less than10^{-5}, whereas the end-to-end latency is within 1 ms. Such is made possible after the introduction of the revolutionary concept of end-to-end network slicing (NS) [140]. Specifically, NS allows multiple logical networks or slices to operate in a shared physical infrastructure. ML can also be exploited for intelligent network slicing to allocate computation and storage resources. At the same time, ML can assist in isolating data traffic from other slices to create a true end-to-end virtual network. The application of ML for NS is just one solution for realizing URLLC. Other promising ML-assisted solutions, such as intelligent network function virtualization, are also worth exploring.

Although several works have already exploited the advantages of ML algorithms in 5G, some room for improvement is always present. The challenges in incorporating ML into 5G networks are still unsolved, such as the availability of quality datasets, poor interpretability of ML algorithms, privacy and security issues, and limited learning resources at edge networks [129]. These challenges need to be addressed in future research before network providers decide to incorporate ML-assisted solutions into their complex and fast-paced networks. Researchers have recently begun to venture toward the 6G realm such as in [140], [141], which is expected to have volumetric spectral and energy efficiency that is a hundred times higher than that of 5G. 6G is also expected to have a very complex structure due to its high connectivity. A tremendous amount of data may be collected from users, and approaches with strong processing and learning capabilities, such as ML algorithms, may show potential in managing complex networks at different levels and applications.

XI. CONCLUSION

This study surveys the recent applications of ML algorithms in networking, such as congestion control, predictive network model, intrusion detection system, route and path allocation, QoS improvement, and resource management. The fundamental workflow of state-of-the-art ML models, such as supervised, unsupervised, and semi-supervised learning, are also discussed. Apart from explaining the above applications, the recent issues and related works on ML are also discussed. As the volume of network traffic grows exponentially, a flexible and intelligent network management is essential to cater to the bandwidth-hungry and stringent delay demand. Although conventional approaches can solve networking issues to some extent, they may be unable to cater to the complexity of future networks. Some limitations of these approaches include their manual configurations with a fixed matrix, limited computing capacity, long execution time with high overhead load, and slow response to network changes. ML has recently emerged as a disruptive technology that fills the computational complexity and performance gap to solve problems in a network. ML has gained considerable popularity due to its ability to provide frameworks for solving problems that involve large-scale data processing, classification, and intelligent decision making. ML algorithms can learn from the complexity of networks and provide decisions dynamically according to the changes in these networks. This study summarizes the simulation and experimental results that prove the superiority of ML-based algorithms over conventional approaches.

However, network providers need to address some other issues before implementing ML algorithms in their networks. As discussed above, despite its outstanding accuracy, ML suffers from a high computational load, especially those algorithms with iterative-based learning, such as the ANN, RL, or online-based learning algorithms. A trade-off between computation load and accuracy must be taken into consideration. A high computational load can also increase the cost for real-world implementation. Moreover, historical data are essential elements of ML algorithms. Network providers must consider solving network problems by using easily accessible data to train ML algorithms. Otherwise, the accuracy of these ML algorithms may be compromised. Imbalanced dataset issues and sophisticated feature engineering works, as discussed for IDS, must also be taken into consideration. The ML algorithm may also issue false predictions when high-priority applications, especially those that involve protection, are involved. This algorithm must work in a way that when a false prediction occurs, the applications can still satisfy the QoS requirements. Whether the ML algorithm can improve the efficiency and quality of the network in practice is a problem worth exploring. The advancement of network technologies that are expected to support higher data volumes per unit area with lower latency, such as 5G networks, can motivate researchers to continue exploring the possibility of exploiting the advantages of ML algorithms. Besides, computing technology has also been improving along with the introduction of better processing units and programmability features. In sum, ML algorithms hold much application potential in network communication systems.

REFERENCES

[1] K. Nugroho, E. Noersasongko, Purwanto, Muljono, and H. A. Santoso, ‘‘Javanese gender speech recognition using deep learning and singular value decomposition,’’ in *Proc. Int. Seminar Appl. Technol. Inf. Commun. (iSemantic)*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, 2019, pp. 251–254.

[2] Z. Han and J. Wang, ‘‘Speech emotion recognition based on deep learning and kernel nonlinear PSVM,’’ in *Proc. 31st Chin. Control Decis. Conf. (CCDC)*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, 2019, pp. 1426–1430.

[3] A. Sarkar, S. Dasgupta, S. K. Naskar, and S. Bandyopadhyay, "Says who? Deep learning models for joint speech recognition, segmentation and diarization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Apr. 2018, pp. 5229–5233.

[4] M. Satsangi, M. Yadav, and P. S. Sudhish, "License plate recognition: A comparative study on thresholding, OCR and machine learning approaches," in *Proc. Int. Conf. Bioinf. Syst. Biol. (BSB)*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Oct. 2018, pp. 58–63.

[5] V. D'Orazio, J. Honaker, R. Prasady, and M. Shoemate, "Modeling and forecasting armed conflict: AutoML with human-guided machine learning," in *Proc. IEEE Int. Conf. Big Data*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Dec. 2019, pp. 4714–4723.

[6] X. Zhou, M. Sun, G. Y. Li, and B. F. Juang, "Intelligent wireless communications enabled by cognitive radio and machine learning," *China Commun.*, vol. 15, no. 12, pp. 16–48, 2018. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.uniten.edu.my/document/8594714

[7] D. Côté, "Using machine learning in communication networks [invited]," *J. Opt. Commun. Netw.*, vol. 10, no. 10, pp. D100–D109, Oct. 2018.

[8] Z. Mammeri, "Reinforcement learning based routing in networks: Review and classification of approaches," *IEEE Access*, vol. 7, pp. 55916–55950, 2019.

[9] D. J. Miller, Z. Xiang, and G. Kesidis, "Adversarial learning targeting deep neural network classification: A comprehensive review of defenses against attacks," *Proc. IEEE*, vol. 108, no. 3, pp. 402–433, Mar. 2020.

[10] S. Otoum, B. Kantarci, and H. T. Mouftah, "On the feasibility of deep learning in sensor network intrusion detection," *IEEE Netw. Lett.*, vol. 1, no. 2, pp. 68–71, Jun. 2019.

[11] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019.

[12] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, 3rd Quart., 2019.

[13] M. Usama, J. Qadir, A. Raza, H. Arif, K.-L.-A. Yau, Y. Elkhatib, A. Hussain, and A. Al-Fuqaha, "Unsupervised machine learning for networking: Techniques, applications and research challenges," *IEEE Access*, vol. 7, pp. 65579–65615, 2019.

[14] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, 2019.

[15] H. Yao, D. Fu, P. Zhang, M. Li, and Y. Liu, "MSML: A novel multilevel semi-supervised machine learning framework for intrusion detection system," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1949–1959, Apr. 2019.

[16] Y. Zhao, Y. Li, X. Zhang, G. Geng, W. Zhang, and Y. Sun, "A survey of networking applications applying the software defined networking concept based on machine learning," *IEEE Access*, vol. 7, pp. 95397–95417, 2019.

[17] D. P. Kumar, T. Amgoth, and C. S. R. Annavarapu, "Machine learning algorithms for wireless sensor networks: A survey," *Inf. Fusion*, vol. 49, pp. 1–25, Sep. 2019, doi: 10.1016/j.inffus.2018.09.013.

[18] R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo, "A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities," *J. Internet Services Appl.*, vol. 9, no. 1, pp. 1–99, Dec. 2018.

[19] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for IoT security based on learning techniques," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2671–2701, 3rd Quart., 2019.

[20] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer-Peer Netw. Appl.*, vol. 12, no. 2, pp. 493–501, Mar. 2019.

[21] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proc. IEEE*, vol. 107, no. 8, pp. 1655–1674, Aug. 2019.

[22] H. Yao, X. Yuan, P. Zhang, J. Wang, C. Jiang, and M. Guizani, "A machine learning approach of load balance routing to support next-generation wireless networks," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 1317–1322.

[23] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-based, AI-based, or both?" *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 7331–7376, Oct. 2019.

[24] N. Hooda, S. Bawa, and P. S. Rana, "MCTOPE ensemble machine learning framework: A case study of routing protocol prediction," in *Proc. IEEE 3rd Int. Conf. Comput., Commun. Secur. (ICCCS)*, Oct. 2018, pp. 92–99.

[25] L. Liu, B. Yin, S. Zhang, X. Cao, and Y. Cheng, "Deep learning meets wireless network optimization: Identify critical links," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 167–180, Jan. 2020.

[26] H. Brink, J. W. Richards, and M. Fetherolf, *Real-World Machine Learning*. Shelter Island, NY, USA: Manning, 2017.

[27] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, A. M. Umar, O. U. Linus, H. Arshad, A. A. Kazaure, U. Gana, and M. U. Kiru, "Comprehensive review of artificial neural network applications to pattern recognition," *IEEE Access*, vol. 7, pp. 158820–158846, 2019.

[28] M. L. Bote-Lorenzo, E. Gómez-Sánchez, C. Mediavilla-Pastor, and J. I. Asensio-Pérez, "Online machine learning algorithms to predict link quality in community wireless mesh networks," *Comput. Netw.*, vol. 132, pp. 68–80, Feb. 2018, doi: 10.1016/j.comnet.2018.01.005.

[29] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[30] G. Biau and E. Scornet, "A random forest guided tour," *TEST*, vol. 25, no. 2, pp. 197–227, Jun. 2016.

[31] W. S. Noble, "What is a support vector machine?" *Nature Biotechnol.*, vol. 24, no. 12, pp. 1565–1567, Dec. 2006.

[32] D. Meyer, F. Leisch, and K. Hornik, "The support vector machine under test," *Neurocomputing*, vol. 55, nos. 1–2, pp. 169–186, Sep. 2003.

[33] A. Géron, *Hands-On Machine Learning With Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd ed. Canada: O'Reilly Media, 2019.

[34] C. V. Murudkar and R. D. Gitlin, "Optimal-capacity, shortest path routing in self-organizing 5G networks using machine learning," in *Proc. IEEE 20th Wireless Microw. Technol. Conf. (WAMICON)*, Apr. 2019, pp. 1–5.

[35] J. Wu, *Advances in K-Means Clustering*, 1st ed. Heidelberg, Germany: Springer, 2012.

[36] T. Kohonen, *Self-Organizing mapsNo Title*, 3rd ed. Berlin, Germany: Springer-Verlag, 2001.

[37] G. J. McLachlan, *The EM Algorithm and Extensions*, 2nd ed. Hoboken, NJ, USA: Wiley, 2008.

[38] N. K. Manaswi, *Generative Adversarial Networks With Industrial Use Cases*, 1st ed. New Delhi, India: BPB Publications, 2020.

[39] F. Chollet, *Deep Learning With Python*, 1st ed. Shelter Island, NY, USA: Manning Publications, 2017.

[40] L. Deng and D. Yu, "Deep learning: Methods and applications," *Found. Trends Signal Process.*, vol. 7, nos. 3–4, pp. 197–387, Jun. 2014.

[41] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[42] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, no. 1, pp. 237–285, Jan. 1996.

[43] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[44] L. Rokach, *Data Mining With Decision Trees*, 2nd ed. Singapore: World Scientific, 2015.

[45] R. Genuer, *Random Forests With R (Use R!)*, 1st ed. Paris, France: Springer, 2020.

[46] A. Liaw and M. Wiener, "Classification and regression by randomForest," *R Project*, vol. 3, no. 3, pp. 18–22, 2003.

[47] M. Haddouchi and A. Berrado, "A survey of methods and tools used for interpreting random forest," in *Proc. 1st Int. Conf. Smart Syst. Data Sci. (ICSSD)*, Oct. 2019, pp. 1–6.

[48] K. Torizuka, H. Oi, F. Saitoh, and S. Ishizu, "Benefit segmentation of online customer reviews using random forest," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manage. (IEEM)*, Dec. 2018, pp. 487–491.

[49] M. Murty and R. Raghava, *Support Vector Machines and Perceptrons: Learning, Optimization, Classification, and Application to Social Networks*, 1st ed. India: Springer, 2016.

[50] T. Weise, "An empirical study of the naive Bayes classifier," in *Proc. Int. Joint Conf. Artif. Intell. Amer. Assoc. Artif. Intell.*, Seattle, WA, USA, 2009, pp. 41–46.

[51] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, C. Wang, and Y. Liu, "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 393–430, 1st Quart., 2019.

[52] Z. M. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, ''State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems,'' *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2432–2455, May 2017.

[53] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, and X. Costa-Perez, ''A machine learning approach to 5G infrastructure market optimization,'' *IEEE Trans. Mobile Comput.*, vol. 19, no. 3, pp. 498–512, Mar. 2020.

[54] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, ''Artificial neural networks-based machine learning for wireless networks: A tutorial,'' *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3039–3071, Jul. 2019.

[55] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, ''Optimal and autonomous control using reinforcement learning: A survey,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2042–2062, Jun. 2018.

[56] D. Zhang, X. Han, and C. Deng, ''Review on the research and practice of deep learning and reinforcement learning in smart grids,'' *CSEE J. Power Energy Syst.*, vol. 4, no. 3, pp. 362–370, Sep. 2018.

[57] R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo, ''A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities,'' *J. Internet Services Appl.*, vol. 9, no. 1, pp. 1–99, Dec. 2018.

[58] F. Tang, B. Mao, Z. M. Fadlullah, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, ''On removing routing protocol from future wireless networks: A real-time deep learning approach for intelligent traffic control,'' *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 154–160, Feb. 2018.

[59] S. Troia, A. Rodriguez, I. Martín, J. A. Hernández, O. G. De Dios, R. Alvizu, F. Musumeci, and G. Maier, ''Machine-learning-assisted routing in SDN-based optical networks,'' in *Proc. Eur. Conf. Opt. Commun. (ECOC)*, Sep. 2018, pp. 3–5.

[60] P. Geurts, I. El Khayat, and G. Leduc, ''A machine learning approach to improve congestion control over wireless computer networks,'' in *Proc. 4th IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2004, pp. 383–386.

[61] N. Yuvaraj and P. Thangaraj, ''Machine learning based adaptive congestion window adjustment for congestion aware routing in cross layer approach handling of wireless mesh network,'' *Cluster Comput.*, vol. 22, no. S4, pp. 9929–9939, Jul. 2019, doi: 10.1007/s10586-018-2357-y.

[62] R. Proietti, X. Chen, K. Zhang, G. Liu, M. Shamsabardeh, A. Castro, L. Velasco, Z. Zhu, and S. J. B. Yoo, ''Experimental demonstration of machine-learning-aided QoT estimation in multi-domain elastic optical networks with alien wavelengths,'' *J. Opt. Commun. Netw.*, vol. 11, no. 1, pp. A1–A10, Jan. 2019.

[63] F. Locatelli, K. Christodoulopoulos, M. S. Moreolo, J. M. Fabrega, and S. Spadaro, ''Machine learning-based in-band OSNR estimation from optical spectra,'' *IEEE Photon. Technol. Lett.*, vol. 31, no. 24, pp. 1929–1932, Dec. 15, 2019.

[64] G. Choudhury, D. Lynch, G. Thakur, and S. Tse, ''Two use cases of machine learning for SDN-enabled IP/optical networks: Traffic matrix prediction and optical path performance prediction [invited],'' *J. Opt. Commun. Netw.*, vol. 10, no. 10, p. D52, 2018.

[65] Z. Fadlullah and B. Mao, ''Value iteration architecture based deep learning for intelligent routing exploiting heterogeneous computing platforms,'' *IEEE Trans. Comput.*, vol. 68, no. 6, pp. 939–950, Jun. 2019.

[66] P. Guo, W. Cheng, and Y. Wang, ''Hybrid evolutionary algorithm with extreme machine learning fitness function evaluation for two-stage capacitated facility location problems,'' *Expert Syst. Appl.*, vol. 71, pp. 57–68, Apr. 2017.

[67] R. Dudukovich and C. Papachristou, ''Delay tolerant network routing as a machine learning classification problem,'' in *Proc. NASA/ESA Conf. Adapt. Hardw. Syst. (AHS)*, Aug. 2018, pp. 96–103.

[68] I. Martín, S. Troia, J. A. Hernández, A. Rodríguez, F. Musumeci, G. Maier, R. Alvizu, and O. G. D. Dios, ''Machine learning-based routing and wavelength assignment in software-defined optical networks,'' *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 3, pp. 871–883, Sep. 2019.

[69] F. Morales, M. Ruiz, L. Gifre, L. M. Contreras, V. López, and L. Velasco, ''Virtual network topology adaptability based on data analytics for traffic prediction,'' *J. Opt. Commun. Netw.*, vol. 9, no. 1, pp. 35–45, 2017.

[70] S. M. Kasongo and Y. Sun, ''A deep learning method with wrapper based feature extraction for wireless intrusion detection system,'' *Comput. Secur.*, vol. 92, pp. 38597–38607, May 2020.

[71] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, ''A detailed investigation and analysis of using machine learning techniques for intrusion detection,'' *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 686–728, 1st Quart., 2019.

[72] C. Sinclair, L. Pierce, and S. Matzner, ''An application of machine learning to network intrusion detection,'' in *Proc. 15th Annu. Comput. Secur. Appl. Conf. (ACSAC)*, 1999, pp. 371–377.

[73] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, ''Anomaly-based network intrusion detection: Techniques, systems and challenges,'' *Comput. Secur.*, vol. 28, pp. 18–29, Feb. 2009.

[74] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, ''Machine learning and deep learning methods for cybersecurity,'' *IEEE Access*, vol. 6, pp. 35365–35381, 2018.

[75] *BBC NEWS | Technology | Estonia Fines Man for Cyber War*. Accessed: Jun. 1, 2019. [Online]. Available: http://news.bbc.co.uk/2/hi/technology/7208511.stm

[76] L. Dignan. *Amazon Explains Its S3 Outage*. Accessed: Jun. 1, 2019. [Online]. Available: http://www.zdnet.com/article/amazon-explains-its-s3-outage/

[77] L. Marinos, ''ENISA threat landscape 2013—Overview of current and emerging cyber-threats,'' Eur. Union Agency Netw. Inf. Secur., Heraklion, Greece, Tech. Rep., Dec. 2013, doi: 10.2788/14231.

[78] (2014). *Ello Social Network Hit by Suspected BLOODY DDoS Attack | DDoSAttacks.net*. [Online]. Available: https://ddosattacks.net/ello-social-network-hit-by-suspected-bloody-dDoS-attack/

[79] AFP. (2018). *Top Dutch Banks, Revenue Service Hit by Cyber Attacks*. [Online]. Available: https://www.securityweek.com/top-dutch-banks-hit-cyber-attacks

[80] M. Tozier. (2019). *Norsk Hydro Recovers From Lockergoga Cyberattack*. [Online]. Available: https://www.autonews.com/suppliers/norsk-hydro-rebuilds-after-cyberattack

[81] D. Palmer. (2019). *Labour Party Confirms Cyber Attack was DDoS*. [Online]. Available: https://www.zdnet.com/article/large-scale-cyberattack-hits-labour-party-systems/

[82] R. Lippmann, D. Fried, I. Graf, J. Haines, K. Kendall, D. McClung, D. Weber, S. Webster, D. Wyschogrod, R. Cunningham, and M. Zissman, ''Evaluating intrusion detection systems without attacking your friends: The 1998 DARPA intrusion detection evaluation,'' in *Proc. DARPA Inf. Survivability Conf. Expo (DISCEX)*, vol. 2. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, 1999, pp. 12–26. [Online]. Available: http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA526274

[83] G. Karatas, O. Demir, and O. K. Sahingoz, ''Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset,'' *IEEE Access*, vol. 8, pp. 32150–32162, 2020.

[84] S. Revathi and A. Malathi, ''A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection,'' *Int. J. Eng. Res. Technol.*, vol. 2, no. 12, pp. 1848–1853, 2013. [Online]. Available: https://www.ijert.org

[85] A. Gharib, I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, ''An evaluation framework for intrusion detection dataset,'' in *Proc. Int. Conf. Inf. Sci. Secur. (ICISS)*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Dec. 2016, pp. 1–6.

[86] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, ''Toward generating a new intrusion detection dataset and intrusion traffic characterization,'' in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy (ICISSP)*, 2018, pp. 108–116.

[87] M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, ''Deep learning approach combining sparse autoencoder with SVM for network intrusion detection,'' *IEEE Access*, vol. 6, pp. 52843–52856, 2018.

[88] Y. Yang, K. Zheng, B. Wu, Y. Yang, and X. Wang, ''Network intrusion detection based on supervised adversarial variational auto-encoder with regularization,'' *IEEE Access*, vol. 8, pp. 42169–42184, 2020.

[89] Y. Yu and N. Bian, ''An intrusion detection method using few-shot learning,'' *IEEE Access*, vol. 8, pp. 49730–49740, 2020.

[90] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, ''Learning to compare: Relation network for few-shot learning,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1199–1208.

[91] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, ''An adaptive ensemble machine learning model for intrusion detection,'' *IEEE Access*, vol. 7, pp. 82512–82521, 2019.

[92] A. Nagaraja, U. Boregowda, K. Khatatneh, R. Vangipuram, R. Nuvvusetty, and V. S. Kiran, ''Similarity based feature transformation for network anomaly detection,'' *IEEE Access*, vol. 8, pp. 39184–39196, 2020.

[93] G. Andresini, A. Appice, N. D. Mauro, C. Loglisci, and D. Malerba, ''Multi-channel deep feature learning for intrusion detection,'' *IEEE Access*, vol. 8, pp. 53346–53359, 2020.

[94] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, ''BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset,'' *IEEE Access*, vol. 8, pp. 29575–29585, 2020.

[95] I. Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, ''Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection,'' *IEEE Access*, vol. 6, pp. 33789–33795, 2018.

[96] A. Ananya, A. Aswathy, T. R. Amal, P. G. Swathy, P. Vinod, and S. Mohammad, ''SysDroid: A dynamic ML-based Android malware analyzer using system call traces,'' *Cluster Comput.*, vol. 23, no. 4, pp. 2789–2808, Dec. 2020, doi: 10.1007/s10586-019-03045-6.

[97] R. Taheri, M. Ghahramani, R. Javidan, M. Shojafar, Z. Pooranian, and M. Conti, ''Similarity-based Android malware detection using Hamming distance of static binary features,'' *Future Gener. Comput. Syst.*, vol. 105, pp. 230–247, Apr. 2020.

[98] K. Yu, L.-Z. Tan, X.-J. Wu, and Z.-Y. Gai, ''Machine learning driven network routing,'' in *Proc. 6th Int. Conf. Syst. Informat. (ICSAI)*, Nov. 2019, pp. 705–712.

[99] L. Cao, Y. Cai, Y. Yue, S. Cai, and B. Hang, ''A novel data fusion strategy based on extreme learning machine optimized by bat algorithm for mobile heterogeneous wireless sensor networks,'' *IEEE Access*, vol. 8, pp. 16057–16072, 2020.

[100] V. Vashishth, A. Chhabra, and D. K. Sharma, ''A machine learning approach using classifier cascades for optimal routing in opportunistic Internet of Things networks,'' in *Proc. 16th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Jun. 2019, pp. 1–9.

[101] B. Mao, Z. M. Fadlullah, F. Tang, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, ''A tensor based deep learning technique for intelligent packet routing,'' in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6.

[102] T. Hendriks, M. Camelo, and S. Latré, ''Q^2-routing: A QoS-aware Q-routing algorithm for wireless ad hoc networks,'' in *Proc. Int. Conf. Wireless Mobile Comput., Netw. Commun.*, 2018, pp. 108–115.

[103] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, ''Machine learning in wireless sensor networks: Algorithms, strategies, and applications,'' *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1996–2018, Apr. 2014.

[104] Z. Zhong, N. Hua, Z. Yuan, Y. Li, and X. Zheng, ''Routing without routing algorithms: An AI-based routing paradigm for multi-domain optical networks,'' in *Proc. Opt. Fiber Commun. Conf. (OFC)*, 2019, pp. 1–3.

[105] S. Li, H. C. Leung, E. W. M. Wong, and C. S. Leung, ''Enhancement of extreme learning machine for estimating blocking probability of OCS networks with fixed-alternate routing,'' *IEEE Access*, vol. 7, pp. 52319–52330, 2019.

[106] H. Yao, X. Yuan, P. Zhang, J. Wang, C. Jiang, and M. Guizani, ''Machine learning aided load balance routing scheme considering queue utilization,'' *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7987–7999, Aug. 2019.

[107] R. Alvizu, S. Troia, G. Maier, and A. Pattavina, ''Matheuristic with machine-learning-based prediction for software-defined mobile metro-core networks,'' *J. Opt. Commun. Netw.*, vol. 9, no. 9, pp. D19–D30, 2017.

[108] E. C. Barboza, N. Shukla, Y. Chen, and J. Hu, ''Machine learning-based pre-routing timing prediction with reduced pessimism,'' in *Proc. 56th Annu. Design Automat. Conf.*, Jun. 2019, pp. 1–6.

[109] M. Salani, C. Rottondi, and M. Tornatore, ''Routing and spectrum assignment integrating machine-learning-based QoT estimation in elastic optical networks,'' in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2019, pp. 1738–1746.

[110] L. Li, Y. Zhang, W. Chen, S. K. Bose, M. Zukerman, and G. Shen, ''Naïve Bayes classifier-assisted least loaded routing for circuit-switched networks,'' *IEEE Access*, vol. 7, pp. 11854–11867, 2019.

[111] V. Vashishth, A. Chhabra, and D. K. Sharma, ''GMMR: A Gaussian mixture model based unsupervised machine learning approach for optimal routing in opportunistic IoT networks,'' *Comput. Commun.*, vol. 134, pp. 138–148, Jan. 2019, doi: 10.1016/j.comcom.2018.12.001.

[112] Y. Zhou, Q. Sun, and S. Lin, ''Link state aware dynamic routing and spectrum allocation strategy in elastic optical networks,'' *IEEE Access*, vol. 8, pp. 45071–45083, 2020.

[113] S. T. V. Pasca, S. S. P. Kodali, and K. Kataoka, ''AMPS: Application aware multipath flow routing using machine learning in SDN,'' in *Proc. 23rd Nat. Conf. Commun. (NCC)*, Mar. 2017, pp. 1–6.

[114] P. Wang, S.-C. Lin, and M. Luo, ''A framework for QoS-aware traffic classification using semi-supervised machine learning in SDNs,'' in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jun. 2016, pp. 760–765.

[115] M. S. Mushtaq, B. Augustin, and A. Mellouk, ''Empirical study based on machine learning approach to assess the QoS/QoE correlation,'' in *Proc. 17th Eur. Conf. Netw. Opt. Commun. (NOC), 7th Conf. Opt. Cabling Infrastruct. (OCI)*, Apr. 2017, pp. 1–7.

[116] S.-C. Kao, C.-H.-H. Yang, P.-Y. Chen, X. Ma, and T. Krishna, ''Reinforcement learning based interconnection routing for adaptive traffic optimization,'' in *Proc. 13th IEEE/ACM Int. Symp. Netw.-Chip (NOCS)*, no. 3, 2019, pp. 4–5.

[117] J. Cao, T. Peng, X. Liu, W. Dong, R. Duan, Y. Yuan, W. Wang, and S. Cui, ''Resource allocation for ultradense networks with machine-learning-based interference graph construction,'' *IEEE Internet Things J.*, vol. 7, no. 3, pp. 2137–2151, Mar. 2020.

[118] A. Azzouni, R. Boutaba, and G. Pujolle, ''NeuRoute: Predictive dynamic routing for software-defined networks,'' in *Proc. 13th Int. Conf. Netw. Service Manage.*, 2017, pp. 1–6.

[119] B. Pourpeighambar, M. Dehghan, and M. Sabaei, ''Multi-agent learning based routing for delay minimization in cognitive radio networks,'' *J. Netw. Comput. Appl.*, vol. 84, pp. 82–92, Apr. 2017, doi: 10.1016/j.jnca.2017.01.026.

[120] Y. Nakayama, D. Hisano, T. Kubo, T. Shimizu, H. Nakamura, J. Terada, and A. Otaka, ''Low-latency routing for fronthaul network: A Monte Carlo machine learning approach,'' in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[121] G. Stampa, M. Arias, D. Sánchez-Charles, V. Muntés-Mulero, and A. Cabellos, ''A deep-reinforcement learning approach for software-defined networking routing optimization,'' 2017, *arXiv:1709.07080*. [Online]. Available: http://arxiv.org/abs/1709.07080

[122] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, ''Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience,'' *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1046–1061, May 2017.

[123] M. Demirci, ''A survey of machine learning applications for energy-efficient resource management in cloud computing environments,'' in *Proc. IEEE 14th Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2015, pp. 1185–1190.

[124] S. Khan, H. A. Khattak, A. Almogren, M. A. Shah, I. U. Din, I. Alkhalifa, and M. Guizani, ''5G vehicular network resource management for improving radio access through machine learning,'' *IEEE Access*, vol. 8, pp. 6792–6800, 2020.

[125] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, ''LORM: Learning to optimize for resource management in wireless networks with few training samples,'' *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 665–679, Jan. 2020.

[126] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, ''Resource management with deep reinforcement learning,'' in *Proc. 15th ACM Workshop Hot Topics Netw. (HotNets)*, Nov. 2016, pp. 50–56, doi: 10.1145/3005745.3005750.

[127] R. Mijumbi, S. Hasija, S. Davy, A. Davy, B. Jennings, and R. Boutaba, ''A connectionist approach to dynamic resource management for virtualised network functions,'' in *Proc. 12th Int. Conf. Netw. Service Manage. (CNSM) Workshops, 3rd Int. Workshop Manage. SDN NFV ManSDN/NFV, Int. Workshop Green ICT Smart Netw. (GISN)*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Jan. 2017, pp. 1–9.

[128] S. K. Sharma and X. Wang, ''Toward massive machine type communications in ultra-dense cellular IoT networks: Current issues and machine learning-assisted solutions,'' *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 426–471, 1st Quart., 2020.

[129] M. E. Morocho-Cayamcela, H. Lee, and W. Lim, ''Machine learning for 5G/B5G mobile and wireless communications: Potential, limitations, and future directions,'' *IEEE Access*, vol. 7, pp. 137184–137206, 2019.

[130] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, ''Scheduling policies for federated learning in wireless networks,'' *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, Jan. 2020.

[131] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, ''A joint learning and communications framework for federated learning over wireless networks,'' *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021. [Online]. Available: http://arxiv.org/abs/1909.07972

[132] A. Taïk and S. Cherkaoui, "Federated edge learning: Design issues and challenges," 2020, *arXiv:2009.00081*. [Online]. Available: http://arxiv.org/abs/2009.00081

[133] Y. K. Cheng, R. Y. Chang, and L. J. Chen, "A comparative study of machine-learning indoor localization using FM and DVB-T signals in real testbed environments," in *Proc. IEEE Veh. Technol. Conf.*, Jun. 2017, pp. 1–7.

[134] P. B. Nithin, A. Francis, A. J. Chemmanam, B. A. Jose, and J. Mathew, "Face tracking robot testbed for performance assessment of machine learning techniques," in *Proc. 7th Int. Conf. Smart Comput. Commun. (ICSCC)*, Jun. 2019, pp. 3–7.

[135] G. Liu, K. Zhang, X. Chen, H. Lu, J. Guo, J. Yin, R. Proietti, Z. Zhu, and S. J. B. Yoo, "The first testbed demonstration of cognitive end-to-end optical service provisioning with hierarchical learning across multiple autonomous systems," in *Proc. Opt. Fiber Commun. Conf. Postdeadline Papers*, 2018, pp. 1–3.

[136] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. De Silva, F. Tufvesson, A. Benjebbour, and G. Wunder, "5G: A tutorial overview of standards, trials, challenges, deployment, and practice," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 6, pp. 1201–1221, Jun. 2017.

[137] J. Joung, "Machine learning-based antenna selection in wireless communications," *IEEE Commun. Lett.*, vol. 20, no. 11, pp. 2241–2244, Nov. 2016.

[138] A. Martin, J. Egaña, J. Flórez, J. Montalbán, I. G. Olaizola, M. Quartulli, R. Viola, and M. Zorrilla, "Network resource allocation system for QoE-aware delivery of media services in 5G networks," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 561–574, Jun. 2018.

[139] W. S. H. M. W. Ahmad, N. A. M. Radzi, F. S. Samidi, A. Ismail, F. Abdullah, M. Z. Jamaludin, and M. N. Zakaria, "5G technology: Towards dynamic spectrum sharing using cognitive radio networks," *IEEE Access*, vol. 8, pp. 14460–14488, 2020.

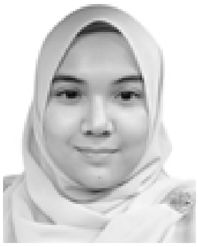
[140] S. J. Nawaz, S. K. Sharma, S. Wyne, M. N. Patwary, and M. Asaduzzaman, "Quantum machine learning for 6G communication networks: State-of-the-art and vision for the future," *IEEE Access*, vol. 7, pp. 46317–46350, 2019.

[141] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, "Future intelligent and secure vehicular network toward 6G: Machine-learning approaches," *Proc. IEEE*, vol. 108, no. 2, pp. 292–307, Feb. 2020.

M. A. RIDWAN received the B.E.E.E. (Hons.) and M.E.E. degrees from Universiti Tenaga Nasional (UNITEN), Malaysia, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree in engineering. His research interest includes machine learning, especially in communication network development.

N. A. M. RADZI (Senior Member, IEEE) received the B.E.E.E. (Hons.) and M.E.E. degrees and the Ph.D. degree in engineering from Universiti Tenaga Nasional, in 2008, 2010, and 2013, respectively. She is currently working as a Senior Lecturer with the Department of Electrical Electronics Engineering, Universiti Tenaga Nasional. She has contributed 50 technical papers in various journals and conferences. Her research interests include optical communication and quality of service. She is also a Professional Engineer and a Chartered Engineer of IET. She is also the Treasurer of the IEEE Photonics Society, Malaysia Chapter.

F. ABDULLAH (Senior Member, IEEE) received the B.Eng. degree in electronics from Universiti Tenaga Nasional, Malaysia, in 2001, the M.Sc. degree in network and communication engineering from Universiti Putra Malaysia, in 2004, and the Ph.D. degree in fibre laser sensor from Universiti Tenaga Nasional, in 2012. He is currently an Associate Professor with the Department of Electronics and Electronics Engineering, College of Engineering, Universiti Tenaga Nasional. To date, he has published more than 80 journal and conference papers. His research interests include optical communications, fibre lasers, optical amplifiers, and fibre optic sensors. He has also been an active member of the IEEE Photonic Society.



Y. E. JALIL received the B.Eng. degree in electrical-telecommunications engineering from Universiti Teknologi Malaysia, in 2002, the M.Sc. degree in digital communication systems from Loughborough University, U.K., in 2005, and the Ph.D. degree in engineering from Universiti Tenaga Nasional, Malaysia, in 2019. She is currently working as a Senior Lecturer with the Department of Electrical Electronics Engineering, Universiti Tenaga Nasional. She has published more than 30 technical papers in various journals and conferences. Her main research interests include antenna design and wireless technology.