IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 24, 2019, accepted July 6, 2019, date of publication July 15, 2019, date of current version August 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2928564

A Survey of Networking Applications Applying the Software Defined Networking Concept Based on Machine Learning

YANLING ZHAO[1], YE LI[1], XINCHANG ZHANG[1], (Member, IEEE), GUANGGANG GENG[2], WEI ZHANG[1], (Member, IEEE), AND YANJIE SUN[3]

[1]Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250101, China
[2]China Internet Network Information Center, Beijing 100190, China
[3]Branch Training Center in Shandong Province, China United Network Telecommunications Group, Jinan 250000, China

Corresponding author: Xinchang Zhang (zhangxc@sdas.org)

This work was supported in part by the National Natural Science Foundation of China under Grant 61802233, and in part by the Natural Science Foundation of Shandong Province under Grant ZR2018MF010, Grant ZR2016YL001, Grant ZR2016YL004, and Grant ZR2016YL008.

ABSTRACT The main task of future networks is to build, as much as possible, intelligent networking architectures for intellectualization, activation, and customization. Software-defined networking (SDN) technology breaks the tight coupling between the control plane and the data plane in the traditional network architecture, making the controllability, security, and economy of network resources into a reality. As one of the important actualization methods of artificial intelligence (AI), machine learning (ML), combined with SDN architecture will have great potential in areas, such as network resource management, route planning, traffic scheduling, fault diagnosis, and network security. This paper presents the network applications combined with SDN concepts based on ML from two perspectives, namely the perspective of ML algorithms and SDN network applications. From the perspective of ML algorithms, this paper focuses on the applications of classical ML algorithms in SDN-based networks, after a characteristic analysis of algorithms. From the other perspective, after classifying the existing network applications based on the SDN architecture, the related ML solutions are introduced. Finally, the future development of the ML algorithms and SDN concepts is discussed and analyzed. This paper occupies the intersection of the AI, big data, computer networking, and other disciplines; the AI itself is a new and complex interdisciplinary field, which causes the researchers in this field to often have different professional backgrounds and, sometimes, divergent research purposes. This paper is necessary and helpful for researchers from different fields to accurately master the key issues.

INDEX TERMS Artificial intelligence, machine learning, network management, software-defined networking.

I. INTRODUCTION

Today, networks are becoming more heterogeneous and complex. It is urgent for networks to optimize traffic distribution and manage a large number of devices. The main task of networks in the future is to build intelligent networking architecture for intellectualization, activation and customization as much as possible. Some scholars have proposed that the networks of the future are IBNs (Intent-Based Networks) or IDNs (Intent-Driven Networks). Their biggest features are the ability to automatically convert custom business requirements into network configuration strategies. This requires a network environment that can provide customized or specialized service networks that meet the needs of different users and can store and transmit large amounts of information data and traffic.

The currently acknowledged leading implementation method of IBN is AI + SDN, in which, AI (artificial intelligence) and ML (machine learning) are used to analyze the collected data, capture intentions, and convert intentions

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

to strategies; however, intelligent software (such as SDN controllers) determines how to translate intentions into a configuration for a particular infrastructure, allowing the network to act in the desired manner.

To ensure high quality computer network operation, maintenance and management, in addition to increasing the capacity of network equipment, we also need to use intelligent tools and technologies to improve the overall performance of the network. The introduction of more intelligent elements is required to meet the needs of different users, reduce operating costs and improve network performance. Thus, the introduction of AI enables the network to meet these challenges. With the superior learning ability of AI, systems can process massive amounts of data with data mining by training data, saving computing resources, and further realizing the intelligent management of networks and services. AI will play a role in network costs and operations and is also expected to enter the network security realm. Detection of complex attack patterns and network optimization are obviously important applications of AI.

As an important enablement method of AI, ML has the ability to actively predict and effectively schedule the network resources based on massive data inputs. ML can be used in each area of AI. It can analyze or simulate people's learning behavior so they can acquire new knowledge or technologies using a computer and rearrange the knowledge architecture to improve its performance. ML learns the rules from automatic data analysis and predicts unknown data. Applying AI and ML to network planning, design, and operations is still in the early stages, since the existing vertical chimney network architectures are not suited to the AI-enabled networks. The introduction of SDN (Software Defined Networking) is facilitating network management and enables programmatically efficient network configurations in order to improve network performance and monitoring [1]. SDN is capable of supporting the dynamic nature of future network functions and intelligent applications with lower operating costs through simplified hardware, software, and management [2]. By getting rid of hardware restrictions on the network architecture, the SDN control plane will have strong processes after decoupling, and it will provide greater speed and flexibility in routing instructions and the energy management of networking equipment such as routers and switches.

Innovations that combine SDN and AI have been involved in every aspect of the network. First, network resource management and operation refers to the unified control of multiple types of resources such as computing, storage, and networks under the integrated SDN network control capabilities. SDN applications provide centralized control of network policies and rules. In addition, they also provide a variety of functions that enable administrators to effectively solve network problems with ML methods. At the same time, network traffic control and management is implemented based on ML methods in the SDN architecture. The control of network traffic is effortless from a holistic network view, since the controller holds all information about the physical networks and their business requirements. In our findings, research on network traffic classification has been a hot topic for some time, which is important for network resource management, optimal route configurations, QoS (Quality of Service) requirements, etc. Network security is also one of the important applications that cannot be ignored. In simplifying network management and shortening the innovation cycle, SDN also introduces security threats that should not be underestimated, such as DDoS (Distributed Denial of Service) attacks and illegal accesses. However, based on the flexible and multidimensional features of the SDN architecture, combined with ML methods for extracting and analyzing network data, the SDN controller will allow detection of DDoS and other anomalies, thus enhancing and guaranteeing the security of networks. Moreover, the rapid development of emerging technologies and terminals has prompted technology upgrades, especially for the multimedia applications at all levels, and pushed them into a new stage of development.

Along the studies' progression, survey work began to appear. In our findings, [3] is a rare and recent research report in this area that is highly related to our work. Its introduction to ML basic algorithms and applications in SDN network is very detailed and provides very valuable reference and guidance. The study in [3] and ours both introduce relevant studies from the perspective of ML algorithms and SDN network applications, and the difference between them lies in the following: the study in [3] separates the two perspectives, while our work introduces one perspective combined with another. From the perspective of ML algorithms, we directly introduce different types of ML algorithms combined with the SDN-concept networks applications in Section II, which will be useful for researchers with professional ML backgrounds to understand applications in SDN. From the other perspective of network applications, SDN-concept network applications are reviewed in Section III (based on recent ML-based architecture solutions), which are more useful for researchers with professional computer network knowledge to study solutions to practical problems.

Another attractive and innovative work is a research study on adaptable and data-driven softwarized networks [4], in which the main focus is on the judicious use of combined SDN/NFV capabilities for data-driven decisions. The study in [4] differs from the surveys of the most current studies in that it considers the additional degrees of freedom introduced by SDN and NFV with a strong focus on the decision phase and proposes a conceptual adaptation framework around the three functional primitives of SDN/NFV networks (i.e., observation, composition, and control). It is more suitable for highly experienced professional researchers who can understand this innovation, while our work will provide guidance for a wider range of researchers.

As a whole, our work focuses on the multidisciplinary characteristics of the ML algorithms and SDN network applications, and introduces the current research from two perspectives. Finally, we prospect the future of this area from

the same two perspectives. This article will be helpful and valuable for researchers with different professional backgrounds and purposes to easily grasp the key issues in this area.

SDN-concept networks after a simple characteristic analysis of the algorithms. Section III surveys different SDN-concept network applications based on ML algorithms. Finally, in Section IV, we put forward future issues facing the SDN concept networks based on ML algorithms. We conclude this study in Section V with discussions.

II. MACHINE LEARNING METHODS IN SDN-CONCEPT NETWORKS

Some scholars believe that the "Train" and "Predict" process in ML correspond to the human "Induce" and "Predict" process, as seen in Fig. 1. Therefore, we conclude that the ideas underpinning ML are not complex, but they are merely a simulation of real life human learning and growth. That is, the general idea of the ML method is based on induction and synthesis, not deduction.

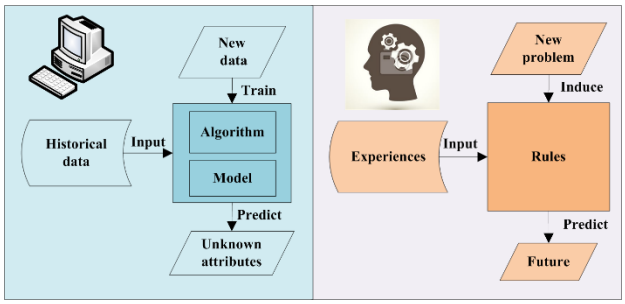

FIGURE 1. A contrast between ML and human thinking.

There are some different classification criteria for ML methods. According to task type, the ML models can be divided into regression models, classification models and structured learning models. The regression model is also called the prediction model, and its output is a numerical value that cannot be enumerated. The classification model is divided into binary classification models and multiple classification models. Spam filtering is one of the common binary classification problems, and classifying documents is a multiple classification problem. However, the output of a structured learning model is no longer a fixed length value; for example, the output is the text description in a semantic analysis of pictures.

According to the model's parameters, ML models can be divided into linear and nonlinear models. A linear model is relatively simple with an assignable role. In addition, it is the basis of a nonlinear model. The nonlinear model includes a traditional ML model and a DL (deep learning) model.

According to the training method, ML methods contain four classes of methods, namely, the supervised learning method, the unsupervised learning method, the semi-supervised learning and reinforcement learning. Supervised learning algorithms build a mathematical model with a labeled training sample. Unsupervised learning algorithms learn from unlabeled sample data. In addition, semi-supervised learning algorithms develop mathematical models from incomplete training data, where a portion of the sample input does not have labels. Compared to the traditional machine learning methods just mentioned, reinforcement learning is concerned with how the agents ought to take actions in an environment to maximize some notion of a cumulative reward. Due to its generality, this field is studied in many other disciplines, such as game theory, control theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, statistics and genetic algorithms.

ML algorithms have been widely used for a number of classification and prediction problems and have provided accurate results [5]. In this section, from the perspective of algorithms, we will do further work, that is, presenting different classical ML methods applied in SDN-concept networks according to the classification of the ML algorithms.

For more clarity, we list the applications and performance analysis of the algorithms in Table 1, and the first appeared abbreviations in Tables of our article are illustrated in alphabetical order in Table 2.

A. SUPERVISED LEARNING IN SDN-CONCEPT NETWORKS

Supervised learning methods make functional inferences by adjusting the parameters of the classifier from a set of samples labeled categories to achieve the required performance. Supervised learning is now used widely in many applications, such as speech recognition, spam detection and object recognition [6]. The goal is to predict the value of one or more output variables given the value of a vector of input variables.

All classification and regression algorithms are supervised learning methods. The difference between the classification and regression algorithms is the type of output variable. The methods with quantitative outputs are called regressions, or continuous variable predictions; the methods with qualitative output are called classifications, or discrete variable predictions. All the values of the input variables can be continuous or discrete and the classifier is a function obtained from the data.

In terms of regression algorithms, a regression is applied to predict the response time for the execution of a query caused by a traffic flow in the SDN architecture [7]. In addition, multiple linear regressions are used to derive a characteristic relationship between the application's key performance indicator (KPI) and the network metrics [8]. As a whole, the application of regression algorithms in SDN is not common at present. In this paper, we mainly introduce the classification algorithms in SDN-concept networks.

The task of classification algorithms is classifying data into the proper category. The most commonly used classification algorithms include KNN (K-Nearest Neighbors), Logistic regression, SVM (Support vector machine), Decision Trees, and Naive Bayesian.

TABLE 1. SDN-concept network applications and performance analysis of ML algorithms.

Algorithm category	Algorithm	Application	Ref.	Performance analysis
Supervised learning	KNN	Predis: detects several other types of attacks besides DDoS attacks	[9]	Easy to implement, highly accurate; Not sensitive to outliers; Calculate features easily; Suitable for multiclass classifications; Time consuming for large datasets
	SVM	Predict link failure;	[10]	Effectively reduces time for starting detection and classification recognition; A lower false alarm rate
		Detect DDoS attack;	[11],[12]	
	Decision Trees	Packet classification;	[13]-[17]	Easy to understand and implement; Data preparation is simple or unnecessary; High-speed; Errors may increase faster, when too many categories
		Inductive inference;	[18]	
		Flow classification;	[7],[19]	
		LCD: optimize the ASP;	[20]	
		Solution for flow table congestion problem	[21]	
	Ensemble learning	RF: indoor localization;	[23]	High accuracy
		RF: regression prediction to model the latency distribution of one VNF.	[24]	
Unsupervised learning	K-means (including variants)	Detect anomalies	[25]	A concrete application in real network; Less accurate and faster; High accuracy of inference; Very low false-positive rates
		Detection of DDoS attacks;	[27]	
		INSPIRE: discover the flow rules;	[28]	
		Solves controller placement problem	[29]-[32]	
Integration of supervised and unsupervised learning	SVM+ K-means	Traffic classification	[33]	Take advantages of various algorithms; Less accurate and faster; Alleviates complexity
	C4.5+K-means	Defining security policies	[7]	
	DT+MCP-PCM	Delivering satisfactory QoE	[20]	
Semi-supervised learning	Laplacian SVM	Traffic classification on the real Internet data	[34]	Used for the same type of applications as supervised learning; Process synthetic data and is only tested in the laboratory
Reinforcement learning	RL	Cognitive network management	[35]-[43]	Manage networks efficiently; Dynamic orchestration of networking, caching and computing resources; Promote resilience and scalability
	DRL	Adaptive multimedia traffic control mechanism leveraging	[44]	
	Deep Q-Learning	Approximate the Q value-action function	[45],[46]	

1) KNN (K-NEAREST NEIGHBORS) IN SDN-CONCEPT NETWORKS

KNN is classified by measuring the distance between different feature values. Its designing principle is that if most of the K nearest samples in the feature space of a sample (i.e., the closest in the feature space) belong to a certain category, then the sample also belongs to the same category. The classification results only depend on a very small number of adjacent samples. KNN is easy to implement, not sensitive to outliers, has high accuracy, and calculates the features easily; further, it is suitable for multiclass classifications.

As a classifier, it has been widely used in many different areas. Zhu *et al.* [9] proposed Predis, a computationally simple and efficient KNN algorithm, as its detection algorithm. It is designed to achieve higher efficiency, which enables it to detect several other types of attacks, in addition to DDoS attacks, with high accuracy. However, KNN is usually implemented by linear scanning, which requires calculating each distance between the test data and the training data, and sorting and finding the nearest K instances. When the training dataset is large, the computation is very time-consuming.

As one of the simplest ML algorithms, KNN is easy to implement and calculates features with high accuracy and is suitable for multiclass classifications. However, it is a time-consuming algorithm when used for large datasets.

2) SVM (SUPPORT VECTOR MACHINE) IN SDN-CONCEPT NETWORKS

SVM is a kind of generalized linear classifier that performs binary classification in a supervised learning manner. Its decision boundary is the maximum margin hyperplane for the solution of learning samples. SVM is stable, since the

TABLE 2. Notes to abbreviations in the tables.

Abbreviations	Full name
ACO	Ant Colony Optimization
BML	Bayesian Machine Learning
CNN	Convolution Neural Network
DDoS	Distributed Denial of Service
DL	Deep Learning
DNN	Deep Neural Network
DRL	Deep Reinforcement Learning
DT	Decision Tree
LCD	Least Cost-Disruptive
MCP-PCM	Multi-Constrained Path-Possibilistic C-Means
NN	Neural Network
QoE	Quality of Experience
RBM	Restricted Boltzmann Machine
RF	Random Forest
RL	Reinforcement Learning
SVR	Support Vector Regression
VNF	Virtualized Network Function

optimization problem considers minimizations of both empirical risk and structural risk. It must be noted that SVM only applies to binary classification tasks. Therefore, multiple classification tasks will be reduced into several binary problems.

The trained SVM [10] model is used to predict link failures based on the input to SVM, namely, the most recent SNR (signal-to-noise ratio) data of the target node. To find the maximum margin hyperplane for separating the data points into two classes, the determination of the weight vector ω and bias b is derived as a quadratic programming (QP) problem, for which the Karush-Kuhn-Tucker (KKT) theorem is applied. In [11], [12], the platform embeds SVM in the controller and uses it to detect DDoS attacks. It can identify benign flow entries generated by normal traffic and malicious flow entries generated by DDoS attack traffic.

In general, SVM is stable and has a lower false-alarm rate for binary classification tasks. The detection scheme effectively reduces the time needed to begin attack detection and classification recognition. When SVM is designed at the SDN controller level, its complexity has very little impact on the efficiency of the SDN system.

3) DT (DECISION TREE) IN SDN-CONCEPT NETWORKS

DT is a predictive model, representing a mapping relationship between object properties and object values. It is a tree structure in which each internal node in the tree represents an object, each branch path represents a possible attribute value, and each leaf node represents a category. DT in data mining is often used to analyze data for prediction.

Its main application in networks is packet classification. These solutions, such as HiCuts [13], HyperCuts [14], EffiCuts [15] or CutSplit [16], are well-known approaches. Considering the dramatically increased dynamism and dimensionality in SDN, PartitionSort [17] is proposed, which combines the benefits of both TSS (Tuple Space Search) and DTs and achieves high-speed packet classification. DT is widely used for inductive inference [18].

For flow classification methods, C4.5 decision tree [19] was selected as the classification method, enabling the handling of a large number of packets. Therefore, it is widely used on high speed network switches. The proposed Real-time Detection Strategy module aims to select n-tuple features to build a robust classification to analyze whether a given n-tuple is an elephant flow or not. Comaneci and Dobre [7] applies C4.5 DT classifiers as pre-trained models for different types of traffic, with features per flow such as inter-packet arrival time, packet size, packet count, flow tuple. A least cost disruptive (LCD) decision tree was proposed (for client, network, or server side adaptations) as a classifier to optimize the ASP (Application Service Provider) data plane, and to handle trade-offs between satisfactory service delivery, cost of adaptations, and user disruption level factors [20]. The work in [21] used the DTs as a solution method for the Flow Table Congestion Problem (FTCP).

Compared with KNN and SVM, the most significant feature of DT is that it is easy to understand and quickly implement, and data preparation for DT is simple or even unnecessary. However, errors may increase more quickly in situations with too many categories.

4) ENSEMBLE LEARNING IN SDN-CONCEPT NETWORKS

In the supervised learning algorithm, the goal is to learn a stable model that performs well in all aspects, even where the facts are not very clear. Ensemble learning is a combination of multiple weak supervised models in order to get a better and more comprehensive strong supervision model. First, a set of individual learners are generated and then combined by a specific strategy. The main ensemble learning algorithms include bagging and boosting.

Though the use of these methods is less than for traditional methods, there are still reports that bagging and boosting approaches outperform other conventional ML methods with a confidence level of more than 99.5% [22]. Taking DT as base learner, RF (Random Forest) constructs bagging integration and applies it in many scenarios. The model in [23] uses RF-based cross validation to train itself and perform an indoor localization with a high accuracy of 98.3%, and its performance is best in other algorithms, such as KNN, SVM and NN (Neural Networks). Lei *et al.* [24] proposes a random-forest regression prediction method to accurately model the latency distribution of one VNF.

Compared with the traditional approaches described above, ensemble learning has superior accuracy, but at the cost of high complexity.

5) DISCUSSIONS OF SUPERVISED LEARNING IN SDN-CONCEPT NETWORKS

From our findings, we list the results of the commonly used supervised learning methods in Table 3. First, as for the number of practical applications, there is little research on logistic regression in SDN [24], while KNN, SVM, DTs

TABLE 3. Prediction accuracy for different supervised ML algorithms.

Ref.	Application	Dataset	KNN	SVM	Naive-Bayes	Decision tree	Else
[65]	Predict network attack patterns	Public dataset from the "Long Tail"	-	-	-	86.19%	Bayes Net: 91.68%, Decision Table: 88.52%
[74]	Detect DDoS attack	Real-time dataset	90%	-	94%	-	K-mediods: 88% K-means: 86%
[73]	Detect DDOS attack	Collected from the switches	97%	82%	83%	-	-
[34]	Predict load type	Generated in a real-time scenario	96.16%	97.72%	93.60%	-	-
[75]	Classify the network into malicious and benign	Created an SDN dataset using Mininet	-	99%	100%	-	Neural Network: 100%
[76]	Detect anomaly based intrusion	NSL-KDD benchmark dataset	98.14%	91.04%	64.16%	99.7%	RF: 99.7%, BaggingTrees: 99.33%, RUSBoost: 99.19%, AdaBoost: 99.03%
[102]	For indoor localization	UCI machine learning repository	-	92.68%	-	-	Random forest: 98.3% Highest NN: 95.16%

and Bayesian methods are used in more applications and attract more attention. Second, as for the detection accuracy of classification methods, although different extracted features and datasets yield different results, the most suitable methods always maintain a high average level of accuracy of over 90%. Third, regarding the selection of methods, the most suitable algorithm is not always the same, and it is better to consider different scenarios, requirements and extracted features. Additionally, ensemble learning approaches such as bagging, boosting and AdaBoost outperform other machine learning methods such as KNN, NN, SVM, which are defined as traditional supervised learning methods that use only one classifier. Lastly, regarding the training speed of the methods, they are not superior to traditional methods and are therefore more suitable to applications without high real-time requirements.

B. UNSUPERVISED LEARNING IN SDN-CONCEPT NETWORKS

In unsupervised learning methods, the label information of the training samples is unknown. The goal is to reveal the intrinsic properties and laws of the data through the study of unlabeled training samples, which provides a further basis for data analysis. The most commonly used method is "clustering", and the simplest and well-known algorithm is K-means.

The clustering method divides data samples into several disparate subsets, each of which is called a "cluster", that is, a category. It is noted that clustering does not know which data category it belonged to before. In [26], a concrete application using unsupervised machine learning in a real network is presented, which demonstrates how the application can detect anomalies at multiple network layers, anticipate anomalies before they become a problem, and identify the root cause of each anomaly, etc.

Standard K-means algorithms applied in SDN are not very common; variant K-means are increasingly being used [27], [28]. An algorithm based on a hierarchical K-means algorithm [29] is used to solve a controller placement problem in an SDN-based WAN architecture, and it is proven to be more balanced than the optimized K-means algorithm. Moreover, there are other variant K-means algorithms based mainly on SDN controller placement problems, such as a heuristic method based on the K-means algorithm and the Dijkstra algorithm [30], a K-means algorithm with cooperative game theory initialization [31], and an optimized K-means [32].

Algorithms comparing or integrating supervised learning and unsupervised learning are also appearing. The purpose of the comparison is to understand the advantages and disadvantages of each algorithm. Barki *et al.* [27] uses different supervised and unsupervised learning algorithms, such as Naive Bayes, KNN, K-means and K-medoids, to classify the traffic as normal or abnormal. The K-means and K-mediods have less accuracy and are faster than Bayes and KNN. The two ML algorithms, supervised SVM and unsupervised K-means clustering, are studied for traffic classification [33].

In practical applications, supervised and unsupervised learning methods are not separated but are instead merged on big data platforms in order to take advantages of various methods. One system [6] reduces complexity by using ML traffic-flow classification techniques and defining high-level SDN policies based on the derived flow classes. C4.5 decision-tree classifiers and K-means are used with features per flow such as interpacket arrival time, packet

size, packet count, and flow tuples. The results showed that the classifiers perform fairly well (the F-score is higher than 80%) with normal traffic, while their performance with abnormal traffic still remains high enough (the F-scores dropped approximately 10%∼15%) for the system to be useful. In [20], a PCM-based clustering scheme based on K-means and the LCD (least cost-disruptive) decision-tree scheme are proposed for delivering satisfactory user QoE by synergistically optimizing both ASP management and data planes. Additionally, the MCP-PCM ensures satisfactory user SLOs (service level objectives that meet user QoE expectations) during cloud service placement, and the LCD decision tree is adaptable to different scenarios, gaining up to 50% in (profiled) user QoE over related solutions. Finally, GENI Cloud testbed experiments were conducted to examine how the proposed algorithms improved overall QoS and enhanced user QoE.

It is certain that the unsupervised learning methods are also based on the learning and training of a large amount of data, and what is learned is not the data source, but the judgment rules from the data dynamics. In addition, they work while establishing rules, adjust at any time, and are more intelligent. Unsupervised learning is considered to be a relatively focused technology field of artificial intelligence.

C. SEMI-SUPERVISED LEARNING IN SDN-CONCEPT NETWORKS

Traditional ML technology is divided into two categories, supervised and unsupervised learning as stated above. Supervised learning uses only labeled sample sets for learning, while unsupervised learning uses only unlabeled sample sets. However, in many practical problems, there is only a small amount of labeled data, because of the very high cost of labeling data; while a large amount of unlabeled data is easily available. This led to a rapid development of semi-supervised learning techniques that can use both labeled and unlabeled samples. It is a learning method that combines supervised learning with unsupervised learning. It mainly considers how to use a small number of labeled samples and a large number of unlabeled samples for training and classifying. Semi-supervised learning is used for the same type of applications as supervised learning [6].

Since its inception, semi-supervised learning has been mainly used to process synthetic data and has been only tested in the laboratory, while [34] conducted experiments to realize accurate traffic classification of real Internet data. The QoS parameters may be used to efficiently reroute "elephant" flows to meet the resource utilization goals. In addition, semi-supervised ML is employed in the QoS classifier to handle the traffic from unknown applications. Relatively speaking, its practical significance has not been reflected. In addition, the practical value of semi-supervised learning is worth more research.

D. REINFORCEMENT LEARNING IN SDN-CONCEPT NETWORKS

Reinforcement learning (RL) is the reward guidance behavior that an agent learns in a trial-and-error manner and obtains reward through interactions with the environment. The goal of a reinforcement learning system (RLS) is to dynamically adjust parameters to achieve the maximum reinforcement signal. The reinforcement signal provided by the environment is a good or bad evaluation of the resulting action, rather than telling the system how to produce the right action.

From our research, RL is usually used for promoting resilience and scalability [35], [36], and it provides path selection or route optimization in SDN-concept networks in [37]–[40]. DROM [37], when it considers delay minimization and throughput maximization as the operation and maintenance strategy, has good convergence and effectiveness, and improved network performance with stable and superior routing services. SDCoR [38] is the first study that can provide an optimal routing policy adaptively through sensing and learning from the IoV (the Internet of Vehicles) environment, and achieves better performance than several typical IoV protocols. To deal with the key challenge of a high level of jitter, the configuration must aim to minimize the usage of different paths for contiguous data frames [39], which is solved by having larger packet bucket sizes, and by minimizing the number of contiguous packets following different paths. Additionally, the jitter level in [40] mostly remained below 40 *milliseconds* by avoiding the high-loss-rate routes, which was significantly better than what can be achieved using traditional routing.

For better performance, some novel research on RL is proposed to combine with other technologies. For example, Random Neural Networks with RL are developed to find the optimal overlay paths with minimal monitoring overhead [41]. SRSA [42], an RL-based auto-scaling decision mechanism, was studied for auto-scaling policy decisions. Furthermore, RL with architecture changes are discussed, because of the complicated and dynamic network environment. Daher *et al.* [43] proposed a scalable approach based on distributed RL in order to manage SON (Self-Organizing Networks) enabled networks efficiently.

Deep Reinforcement Learning (DRL) takes advantage of both DL and RL, and improves the learning speed and the performance of RL algorithms. DRL has achieved remarkable results in both theory and application. In particular, the DRL-based AlphaGo, produced by the Google DeepMind team, is considered to be a new milestone in the history of AI. Our findings confirm that DRL has made some progress in SDN-concept networks.

Huang *et al.* [44] studied DRL for adaptive multimedia traffic control mechanism leveraging. It is able to control multimedia traffic directly without a mathematical model. In particular, Deep Q-Learning (DQL) is mostly used for the DRL related works [45]. And different DQL techniques

can be used to solve different problems in different network scenarios. He *et al.* [46] proposed an integrated DQL framework consisting of SDN architecture in which deep Q network is used to approximate the Q value-action function. The proposed framework improves the performance of green heterogeneous wireless networks. To improve the throughput of a blockchain system, Qiu *et al*. [47] proposed a new, dueling DQL approach to consider the trust features of blockchain nodes and controllers, and computational capability as a joint optimization problem. TDRL-RP [48] was proposed based on the DQL framework in a logically centralized controller of SDN for VANET (vehicular ad hoc networks), in which a trust model is introduced to decide the immediate trust path for the long-term reward (Q-value) in path learning; DQL is used to determine the best routing policy.

Overall, RL is an important ML method and is used widely in network related issues. Note that it only characterizes the interaction procedures instead of providing another learning method. In addition, each learning algorithm can be transformed into a RL [49], and it will be widely used for analysis and prediction.

E. DISCUSSIONS OF MACHINE LEARNING METHODS IN SDN-CONCEPT NETWORKS

In Section II, we introduced the applications of four classical ML algorithms in SDN-concept networks, including supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. The above findings indicate that supervised learning algorithms in SDN-concept networks (in which an optimal model is obtained by training existing training samples) exhibit a relatively more mature state of development, and perform fairly well compared to unsupervised learning algorithms and semi-supervised learning. Unsupervised learning differs from supervised learning in that we need no training data in advance, and can directly model the data. The purpose of clustering (a typical unsupervised learning algorithm) is to bring together similar things without caring about their types. Therefore, a clustering algorithm will work as long as it is known how to calculate similarity. Compared to supervised and unsupervised learning methods, the study of semi-supervised learning started relatively late, and its practical application has not yet been fully demonstrated. For more resilience and scalability, RL is the prevalent approach to dynamically adjusting parameters in intelligent networks, which is more in line with the future network development trends.

III. SDN-CONCEPT NETWORK APPLICATIONS WITH ML METHODS

Section II introduces the ML methods applied in SDN-concept networks based on the classification of ML methods, which is the focus of our work. It is helpful for researchers to make clear the characteristics of ML algorithms. In this section, from the practical application perspective of SDN-concept architecture, we will introduce solutions based on ML methods.

SDN aims to create an ecosystem of opening switches and controlling software to achieve rapid innovation and a fresh environment that is easy to integrate. Several cases benefit from the application of ML and data analytics techniques. In this paper, according to the different application scenarios, we divide the existing application cases into five categories, namely, resource management and allocation, flow and traffic processing, system security guarantees, theoretic architecture approaches and parameter modeling and promotion in multimedia content services. We will introduce these application cases in SDN-concept networks with ML methods in detail as provided in Table 4.

A. RESOURCE MANAGEMENT AND ALLOCATION

Resource management problems in systems and networks often mainly focus on difficult online decision making, while appropriate solutions depend on understanding the workload and environment [50]. Therefore, more research is urgent in the area of resource management. SDN extracts the control plane from the distributed network device and controls the entire network from a centralized controller. The controller can detect the resource capacity and network requirements from a global perspective. Thus, the introduction of SDN improves the control of network resources and enables the automation of management. Its resource management achieves unified control over multiple types of resources such as computing, storage, and networks, and meets the needs of resource delivery in business scenarios.

It is necessary to build a system that learns to interact with the environment to manage resources directly from experience. Inspired by recent advances in DRL for AI problems, DRL is assumed to be a promising solution. The general setting [51] is shown in Fig. 2, where an agent interacts with an environment. RL deals with agents that learn to make better decisions directly from the experience of interacting with the environment. The agent begins knowing nothing about the task at hand. At each time t, there exists a state St, and the agent is asked to choose an action At; then the state of the environment transitions to $S(t + 1)$, and the reward Rt is received. The state transitions and rewards are assumed to have the Markov property. In addition, the state transition probabilities and rewards depend only on the state of the environment St and the action At chosen by the agent.

DeepRM [52] was presented as a solution that translates the problem of packing tasks with multiple resource demands into a learning problem. The initial results showed that DeepRM adapted to different conditions, converged quickly, and learned strategies that were sensible in hindsight. It is feasible to apply state-of-the-art DRL techniques to large-scale systems. In [53], a multi-agent learning algorithm was proposed to implement the substrate network resource management in a coordinated and decentralized way. The task of these agents is to learn an optimal policy by evaluative feedback and then dynamically allocate network resources to virtual nodes and links.

TABLE 4. SDN-concept network applications and performance analysis ML algorithms.

Scenarios	Applications	Ref.	ML algorithms	Performance analysis
Resource management and allocation	Allocate network resources	[48]-[50] [53]-[57]	DRL SVR	Learn to interact with the environment, to manage resources directly from experience; Adapts to different conditions, converges quickly.
	Network slicing	[51][52]		
Network flow and traffic management	Network traffic control	[58]-[65]	RL, DT, BML, SVM, K-means, DL	Extract knowledge from network traffic; Master types of applications accurately; Provide a real-time identification for data.
	Network traffic classification	[25], [33]-[34], [66]-[79]		
Network security protection and guarantee	Network intrusion detection	[5]-[6],[36], [59],[82]-[90]	SVM, NN, RL, RBM	Strengthen network security; Realize timely detection and responses; against network intrusion; Detect DDoS attacks with high effectiveness.
	Network attacks detection	[94]-[105]		
	Network fault diagnosis	[106]-[111]		
Theoretical framework approaches and indicator modeling	Theoretical framework approaches	[115]-[123]	RL, multiple MLs, ACO, NN, CNN, DRL	Actuate automatic optimization and reconfiguration strategies; Improves the proactive fault response time; Capable of achieving or balancing objectives; The performance of modeling work depends on chosen features and situations, sometimes presents poor results.
	Routing optimization	[46],[58],[115] [124]-[129]		
	Indicator modeling	[132]-[134]		
Promotion in multimedia content services	Estimate available resources	[137]	RF, DL, DNN, Similarity learning	Improve network efficiency and alleviate the high demand for the network resources.
	Network edge caching	[138]		
	Content popularity prediction	[139]		

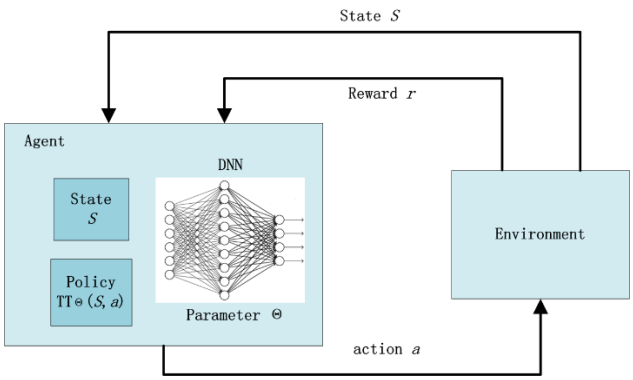

FIGURE 2. RL with policy presented via DNN.

Furthermore, intelligent innovations make the resource management consistent with users' activities per slice. Authors in [54] and [55] have applied DRL to network slicing, and the application of DRL performs well in solving some typical resource management for network slicing scenarios, such as radio resource slicing and priority-based core network slicing. Martin *et al.* [56] provided a network resource allocator system that enables autonomous network management aware of QoE. In addition, [57] focuses on a simple and practical experience-driven approach based on DRL, which is easily used in solving complicated control and resource allocation problems in communication networks.

Research on network resource management becomes a critical requirement with new tools in different application scenarios. For 5G network service providers, a new framework that adds a smart node with MEC (Mobile Edge Computing) and new tools such as ML [58], [59], enables the hosting of applications close to end users with reduced latency and improved performance, facilitates the management and efficient allocation of network resources, and improves the services of network providers [58]. A deeper study by Abderrahim *et al*. [59] indicated that the optimal placement of the smart node can be problematic, and can be determined using game theory.

As for another important VNF scenarios, its most challenging task is to meet the continuously varying demands of dynamic algorithm calls, in order to efficiently scale the allocated resources and meet fluctuating needs. After studying the behavior of a VNF as a function of its environment, an SVR (Support Vector Regression) approach was proposed [60] that helped model its resource requirements in order to allocate them dynamically, with greater efficiency and superiority than the state-of-the-art methods.

It is easy to understand that network resource management and allocation have become critical to enhancing network performance due to the increase in various network applications. Currently, more and more new tools such as ML and MEC are being used to make decisions adaptively and intelligently based on information from a global perspective. Although there are many problems that have not yet been solved, some exploratory, theoretical work has been discussed, such as the modeling of the optimal placement of

the intelligent entity in [59], and estimating the CPU needs of VNFs as a function in [60].

B. NETWORK FLOW AND TRAFFIC MANAGEMENT

1) NETWORK TRAFFIC CONTROL

A network traffic flow is defined to be a sequence of data packets. In the case of SDNs, the flow information is useful for programming routers, mitigating wireless interferences, scheduling congested data traffic, and so on. Network traffic control is a kind of control of computer network traffic that uses software or hardware. Its most popular method is to determine the priority of packet traffic by marking different types of network packets.

Flow prediction is an important area of network traffic control that has witnessed a growing number of deep learning applications recently. Fadlullah [61] provided an overview of the state-of-the-art DL architectures and algorithms relevant to the network traffic control systems. To manage the limited network resources, flow characteristics such as the burst size (i.e., packet number and packet-size) and the inter-burst gap are often used. Wang *et al.* [62] proposed an ID3 decision tree theory to outrank raw features, and it determines the most qualified features for flow management approaches with SDN architecture.

The related research also includes the problem of predicting throughput for reactive flows defined formally with source constraints [63]. Combining RL and mixed integer linear programming, a traffic prediction is used to dynamically provide resources in advance [64]. Alawe *et al.* [65] anticipates traffic load changes in 5G, and Zhang *et al.* [66] estimates upcoming traffic rates effectively for flow service quality assurance and resource cost minimization. For reducing latency and overheads, Bayesian Machine Learning (BML) is used to allow the controller to classify packets into flows. In addition, a switch assigns those packets whose flows are not given previously by the controller to the most appropriate flow [67].

As for traffic matrix (TM) measurement and inference, OpenTM and Open-NetMon measure the traffic matrix by keeping track of statistics for each flow. These per-flow-based solutions do not scale well with an increase of traffic size and impose heavy overheads on the network. OpenMeasure [68], a network-wide adaptive flow measurement and inference framework with continuous learning capability, takes advantages of online learning and the global optimization enabled in SDN. It can continuously track and measure the most informative flows based on the dynamic adjustment scheme.

We introduce the principal research on network traffic control, including flow prediction, throughput prediction, and TM measurement. Though this work started early on, its implementation is not satisfactory because of difficulties with the actual environments.

2) NETWORK TRAFFIC CLASSIFICATION (TC)

The task of network traffic classification is to match all the traffic in the networks with the applications that generate them in real time, so that the running applications can be mastered accurately in the networks. Network TC is an important prerequisite for network management, QoS, and security monitoring of various networks [69], [70].

In recent years, research on TC has been well developed. Several well-known techniques have been proposed, as well as sharing of some key limitations [71]–[73]. For example, the effectiveness of port-based approaches has diminished even when port numbers carry valuable information about the application or protocol. Deep packet inspection (DPI), relies on the availability of a training set, cannot provide a real-time identification for encrypted data traffic, and needs an expensive retraining phase. Big data analysis extracts information from raw data, but it often requires ML algorithms [72]. To resolve these issues, self-tuning, simple tools are proposed to extract knowledge from network traffic, including different data analytics techniques. Unlike state-of-the-art classifiers, the biggest advantage of Self Learning Network Traffic Classification is its ability to discover new protocols and applications in an almost automated way. The algorithms, based on different ML technologies [69], [74], are proposed to classify the network traffic.

With the increasing role of SDN, the research for SDN architecture [75] develops encrypted data classifiers (DataNets) based on three DL schemes, i.e., multilayer perceptron, stacked auto encoder, and convolutional neural networks. Wang *et al.* [34] presented a traffic classification engine at the SDN edge switches, and then performed a "global" traffic classification via the network controller, which was responsible for training, constructing, and refining QoS policies based on the learned traffic information. The study in [76]–[79] shows that the ML model outperforms the existing algorithm, and the SDN controller assigns more appropriate route paths for different types of traffic and highly improves the network's QoS. ML algorithms, such as SVM and K-means clustering, are studied for TC with a high accuracy of over 95% [25], [33].

With the great progress in networking, it is imperative to extend the SDN framework to develop TC tools in a scalable, efficient and flexible way with ML techniques. These challenges will remain for some time to come. For example, the existing TC approaches using ML in SDN are mainly based on supervised or unsupervised learning in wireless sensor networks (WSNs). TC faces several challenges, including energy efficiency, shareable testing data and design [80]. Overall, increasing network bandwidth requires a large-scale, fine-grained, and adaptive TC method to process gigabits or even more data per second in future applications [70], [81].

C. NETWORK SECURITY PROTECTION AND GUARANTEE

When applications or businesses are running on the networks, security is the first issue to be addressed. Providing security measures is a critical step to fully unleash the new model's capabilities. In addition to the security provided by traditional anti-threat applications (such as firewalls, anti-virus

software and spyware detection software), network behavior analysis is an important component of security protection. In this section, we list three important and related applications, namely, network intrusion detection, attack detection and fault diagnosis.

1) NETWORK INTRUSION DETECTION

Intrusion detection is an option for enhancing networking security [82]. As a proactive security protection technology, intrusion detection provides real-time protection against internal attacks, external attacks, and accidental operations. In addition, it is considered to be a second safety gate after firewalls. It detects attacks and mishandlings in a timely manner without any performance influence on networks. To provide more thorough understanding of this research, Sultana *et al.* [83] recently reviewed various recent studies on SDN-based ML methods to implement Network Intrusion Detection Systems (NIDS). These authors evaluated the features of various ML algorithms, including supervised learning, unsupervised learning, and semi-supervised learning. In addition, DL methods are considered as a modern update to artificial neural networks. Lastly, this paper identifies challenges and provides conclusions for future studies in this area.

Specifically, NIDS methods are divided into misuse detection and anomaly detection. The advantage of the former is that it is highly accurate, but it does not handle new attacks; the latter can effectively detect new attacks [6]. At present, the latter has not been very mature, and it remains a hot topic in the development of intrusion detection systems (IDS) [84]. The diagram of intrusion detection is shown in Fig. 3.

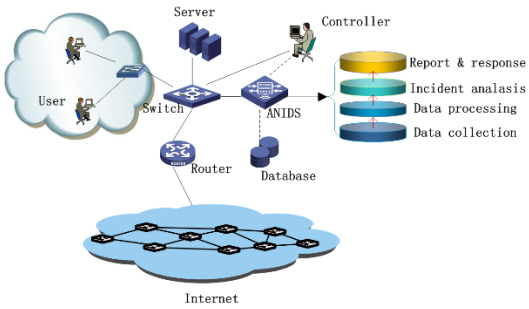

FIGURE 3. The diagram of network intrusion detection system.

Network anomaly is based on rules that assume that there are some regular patterns in normal behavior, which can be concluded by analysis of log information. An anomaly happens when there are serious aberrations from the norm, which can be detected by the different behavior. In conclusion, anomaly detection is based on the assumptions that the pattern of normal behavior is regular and described efficiently by the data and has obvious difference from normal behavior. Therefore, the first stage of network anomaly detection is a learning stage, in which transcendental knowledge is obtained for normal patterns. Then, the second stage is testing, in which the preprocessed data is compared with

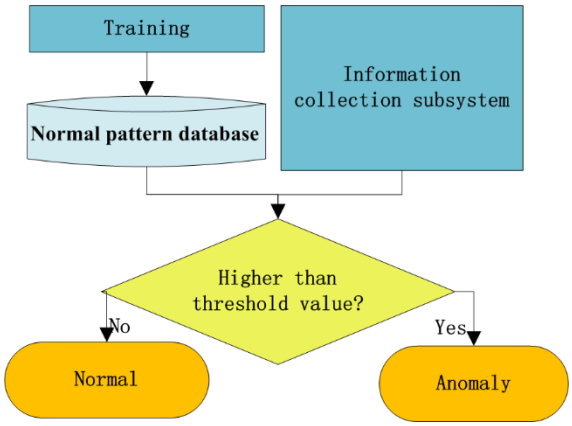

FIGURE 4. The flow of network anomaly detection.

the normal data. When the difference is not higher than the threshold value, the case is normal. Otherwise, it indicates an anomaly. This can be seen in Fig. 4.

Though SDN brings another security burden with more open network vulnerabilities [85], it provides a chance to strengthen network security by the decoupling of its control plane and data plane. The proposed system in [6], [14] enables timely detection and responses against network intrusion in SDN. Except for a small amount of research on streaming data anomaly detection [86], most research focuses on flow-based approaches using familiar ML methods with minimal extra overhead [87]. Based on SVM, Study in [62] is to categorize network threats, and [85] overcomes the limitations of signature-based IDS, which has a positive improvement for the detection of almost all the possible attacks in SDN with an over 97% accuracy. Meti *et al.* [88] uses the SVM classifier and the NN classifier to detect suspicious and harmful connections. Further, [89] uses a feature selection method, which has been proven to have a strong potential for detecting anomalies in the OpenFlow controller.

To promote resilience in SDN, policies for dealing with anomalies are defined by using RL based on rewards for each action [36]. In addition, the results show that it obtains mostly positive rewards. Garg *et al.* [90] proposed a hybrid DL- based anomaly detection module by leveraging the improved Restricted Boltzmann Machine and SVM. In addition, a large-scale analysis is conducted to identify its performance in detecting malicious events such as identity theft, profile cloning, and confidential data collection.

To detect threats from malicious SDN applications, Boero *et al.* [91] uses SVM as a core system for detecting malware by using only traffic features. Indago [92], statically analyzes SDN applications to model their behavioral profiles, and automatically detects most known SDN malware applications with a high detection rate and low error rates.

Furthermore, newly emerging technologies make the applications of intrusion detection increase rapidly. A fog-assisted SDN driven intrusion detection system for IoT networks identifies various attack models in near real time for effective

neutralization of threats [93]. In addition, it has proven to be more effective than traditional techniques. Although there is still related research with poor results, there is proven potentially in using DL for flow-based anomaly detection systems [94].

2) NETWORK ATTACKS DETECTION

In addition to the network anomalies described above, malicious attacks that can eventually halt network services are unavoidable [95]. Among the most predominant attacks on the SDN controller layer, Link Discovery Attacks and ARP (Address Resolution Protocol) Spoofing Attacks are fundamental since they are the gateways to many other SDN threats and attacks [96], [97].

Moreover, the distributed denial of service (DDoS) flooding attack continues to be one of the major security concerns as these types of attacks are increasing year by year. A DDoS attack concentrates on making resources unavailable to legitimate users via overloading systems with superfluous traffic from distributed sources [98]. Many studies have shown Slowloris, ACK (acknowledgement) and SYN (synchronization) flooding attacks to be notorious among the several other forms of DDoS attacks [99]. The effectiveness of using ML to detect DDoS attacks in SDN is discussed [100]–[102]. Though the classifier in [87] is trained only for the DNS amplification attack, both DNS and NTP amplification attacks are blocked with great accuracy. Koning *et al.* [103] concludes that in the SDN architecture it is possible to achieve high effectiveness of response by carefully choosing a relatively minor number of actions.

For more applications, JESS [98] is the first model that utilizes joint entropy for DDoS detection and mitigation in SDN. Since, by reliance on a statistical model, it mitigates not only known attacks but also unfamiliar attacks in an efficient manner. Siddharth and Sterbenz [104] extends the functionality of the SDN controller to include a resilience framework, ReSDN incorporates ML to distinguish DoS attacks. And the advanced SVM (ASVM) [105] was proposed based on SDN for DDoS attacks. It significantly reduces the testing time as well as training time compared with the traditional SVM algorithm. Studies on [79], [106] have found that the DDoS attack defender outperforms the existing mechanisms in an SDN-based cloud environment. The framework in [107] is capable of detecting DDoS on IoT with an approximate 98% accuracy. In addition, in [108], a protocol for multi-SDN controllers is designed. The main task of the protocol is to build and maintain an independent network and exchange attack information among the controllers of different SDNs, then find attackers and mitigate the DDoS attack.

In DDoS attacks, the attacker sends large amount of malicious packets to the victim server, and the legitimate users fail to access resources. It is worth noting that, Bakker *et al.* [109] suggests that the technologies for DDoS detection in SDN with ML methods may inadvertently lead to degraded performance for legitimate network traffic. Their main task in the future is to prevent malicious traffic from affecting the network's performance and allow legitimate traffic to circulate.

3) NETWORK FAULT DIAGNOSIS

With the increasing scale of networks, their complexity is increasing, so networks face severe challenges in putting in place effective management. Fault diagnosis is one of the most important and difficult tasks in network management. If the fault in a network cannot be diagnosed and repaired quickly, it not only increases the operating cost for operators but also reduces the service quality for users.

The introduction of SDN creates costs for a potential increase in failures, since each modification of the controller will produce a new possibility for failures and decrease the quality of service; this can be resolved by classical approaches with dramatically increased cost. Failure prediction has become a reality thanks to the introduction of ML techniques. Benayas *et al.* [110] presented an architecture for a self-diagnosis service with ML and data analysis. In addition, it is encouraging that a prototype with different diagnosis models for SDN has been developed, which will be explored in future.

Rafique *et al.* [111] introduces the concept of cognitive fault management, elaborates on its integration into a transport SDN controller, and demonstrates its operation based on real-world fault examples. It detects and identifies significant faults and outperforms conventional fixed threshold-triggered operations. Rafique *et al.* [112] proposed an SDN-integrated framework for distributed cognitive fault discovery and diagnosis across an end-to-end network infrastructure. Jagadeesan and Mendiratta [113] shows how ML-based detection is used to identify SDN software faults and helps with real-time network responses.

In terms of single link failure scenarios, this kind of approach focuses on reducing the update operation costs [114]. Natalino *et al.* [115] studies an orchestration strategy for optical cloud networks that are able to reconfigure vulnerable cloud services before an actual failure takes place. It is proven that proactive restoration leads to up to 97% fewer cloud services having to be relocated, which is a benefit for cloud service availability, especially in low load conditions.

Based on our findings, not much research has been done on fault diagnosis. It is promising that some further directions are suggested to achieve future advances in this research area based on the existing results.

It is important to note that there is a gap between academic research on ML-based solutions for SDNs and their operational deployment. Previous surveys have been done on adversarial machine learning or on the general vulnerabilities of SDNs, but not both. For the first time, Nguyen [116] aim to provide a complete picture regarding ML-based security solutions for SDN. In addition, ML models deployed in network detection/prevention systems are recognized to be imperfect. Hence, there is always a possibility for attackers to manipulate and/or bypass the models. Attackers are also equipped

with ML capabilities and will build systems to predict the behaviors of the defending models. At the beginning of their projects, solution designers should pay special attentions to the threat model, the secure development processes, and so on. This study will make a case for more secure development processes, and its recommendations will improve the practical properties of ML-based solutions for SDNs.

D. THEORETICAL FRAMEWORK APPROACHES AND INDICATOR MODELING

In addition to the specific solutions for a particular problem described in the above subsections A to C, there are also studies on the theoretical frameworks and parameter modeling focusing on universality. These research ambitions are for future network models and development directions based on new technologies and will accelerate network innovation.

1) THEORETICAL FRAMEWORK APPROACHES

The use of artificial intelligence and big data technology in SDN and NFV (network function virtualization) to achieve intelligent network traffic management and optimization is of great significance for telecom operators [117]. The cognitive network, described in [118], [119], is a network that is capable of perceiving current network conditions and then planning, learning, and acting according to an end-to-end goal. In addition, SDN uses software to define the network, separates the software and hardware for the network, and centrally controls the network through software. It is not difficult to understand that SDN provides a feasible solution for the realization of cognitive networks. An increasing amount of research tends to use ML and other artificial intelligent methods to build an SDN-concept framework closely related to a cognitive network, which is able to incorporate the self-learning and self-management functions.

Guided by the concept of cognitive network, research on frameworks or mechanisms with cognitive functions continues to emerge. Network-wide load balancing, Lee [120] proposed a cross-layer mechanism in which learning agents in the middleware layer can monitor the queue sizes of the MAC layer. Rafique *et al*. [111] introduced the concept of cognitive fault management and builds a cognitive network assurance architecture for next-generation network management and operation. The framework not only allows for simpler network management, getting rid of the definition and maintenance of multiple fixed set points; it also significantly improves the proactive fault response time. A framework of the autonomic self-managing network [121] is capable of achieving or balancing objectives such as high QoS, low energy usage and operational efficiency. The main novelty of the architecture is the Cognitive Smart Engine introduced to enable ML, particularly (near) real-time learning, in order to dynamically adapt resources to the immediate requirements. This architecture is built within the CogNet European Horizon 2020 project. COBANETS [122], which combine this learning architecture with the emerging network virtualization paradigms, makes it possible to actuate automatic optimization and reconfiguration strategies at the system level, thus fully unleashing the potential of the learning approach.

ML is highly suitable for complex system representations. Danish and Velasco [123] reviews several ML concepts tailored to the optical networking industry and discusses algorithm choices, data and model management strategies, and integration into existing network control and management tools. Aside from the specified one or two ML algorithms, newly proposed theoretic frameworks tend to be proposed with more flexibility. Liu *et al*. [68] points that more advanced ML approaches could be applied in the proposed framework to improve prediction accuracy. The proposed framework ATLANTIC [124] combines the use of information theory to calculate deviations in the entropy of flow tables and a range of ML algorithms to classify traffic flows. ATLANTIC is a flexible framework capable of categorizing traffic anomalies and using the information collected to handle each traffic profile in a specific manner.

The ultimate purpose of all of the studies is to make better use of them. To improve a certain aspect of performance in the system, frameworks for specific application scenarios have emerged. Budhraja *et al*. [124] proposes a risk-based swarm routing protocol for SDN for efficiently guaranteeing the data forwarding performance of the SDN controller. The Risk-Based Packet Routing combined with Ant Colony Optimization (ACO) provides a holistic solution to privacy and risk compliance associated with the whole SDN network. The framework [125] accepts the users' QoS (Quality of Service) demands, their pricing plans and the network constraints to manage the network flow such that revenue collected from the users can be maximized. Xu *et al*. [57] presents an architecture model and builds a model to find the optimal placement of Smart Nodes in the network. He *et al*. [126] considers a well-studied fc-median problem arising in SDN and aims to imitate and speedup existing heuristics as well as to predict good initial solutions for local search algorithms; it is found that NN can provide the best abstraction. Abderrahim *et al*. [59] investigates the use of ML techniques to estimate VNF's needs in terms of a CPU, as a function of the traffic they will process. Compared with the previous offline works, Sieber *et al*. [127] proposes an online approach to determine the mapping of hypervisor resources to the control workload at runtime. In addition, an online ML pipeline [128] is proposed to synthesize a performance model of a running hypervisor instance in the face of varying resources.

As a whole, research on theoretical frameworks has not reached its peak yet, in terms of its scale and versatility. From the results, approaches with ML techniques have been proven to be very useful tools in SDN-concept networks and more efforts need to be taken into this area.

2) ROUTING OPTIMIZATION

Based on the idea of SDN, the central controller is responsible for network link discovery, topology management, policy

making, table issuance, etc., from a holistic point of view. For better theoretical research, routing optimization strategy is one of the key issues of SDN-concept networks used to achieve network load balancing and autonomous control that occupies a large proportion of the relevant theoretical research.

The proposed approach is capable of prioritizing each of the flows and assigns a path based on its classified priority [129]. A ML-based framework in SDN is studied in [130] for traffic-aware and energy-efficient routing. A cross-layer mechanism [120] was implemented, in which learning agents in the middleware layer can monitor the queue sizes of the MAC layer, thereby allowing for the discovery of optimal routes.

Among ML methods, RL techniques have already been widely used for routing optimization, which were pioneered in [131]. Fadlullah [61] focuses on a DL application for intelligent routing operations of a backbone network and shows how the DL-based intelligent routing technique outperforms the conventional routing strategies, such as the OSPF (Open Shortest Path First) routing strategy. Moreover, this approach provides operationally important advantages. Recent attempts also use RL techniques to achieve QoS routing [49]. A DL-based strategy is used to solve the problem by intelligent routing learning and prediction [132], which obtains high network performance in SFCR (Service Function Chain Request) acceptance rates and end-to-end delays. Mao *et al*. [133] utilizes the CNNs as DL architecture, and the controller runs the CNNs to choose the best path combination for packet forwarding in switches. Moreover, there is an attempt to use DRL techniques for routing optimization [134]. It is clear that a fully automated DRL agent can provide routing configurations by minimizing the network's delay.

An intelligent path is a natural requirement, and better performance could be achieved only if the network is capable of prioritizing the flows and assigning resources based on their application specific requirements. By training with the optimal routing solutions of historical traffic traces, the related approaches will certainly provide real-time routing decisions.

3) NETWORK INDICATOR MODELING

A very important part of theoretical research is network indicator modeling, which is an expression of network characteristics. It is helpful to understand the related network scenario and to provide an important aided tool for subsequent work.

Inspired by the existing knowledge of network modeling, [135] refers to the resulting architecture as KDN (Knowledge-Defined Networking [136]) and focuses on model CPU consumption as a function of the input traffic by choosing the ANN (artificial neural network) method.

In [137], the delays of a network are estimated, and an M/M/1-inspired ML regressor characterized the delays in a network given the traffic load. For comparison, the M/M/1-inspired estimator provides more information of the network, rather than a black-box in neural network. The method in [138] uses NN-based techniques to learn a generative input model of proprietary network protocols, and generates new messages used as test cases to fuzz the implementations of the protocols. A prediction of quality of experience is proposed in SDN networks [139].

These modeling studies provide learned models for fundamental problems-based in ML in SDN-concept networks and can be used as an important tool for future analysis and experiments. The experiments prove that the performance of these works depends on the various features chosen for different network situations and sometimes present poor results [137]. At present, this work is not mature and extensive. Future work includes how to represent the configuration in the learning process to enable the use of the same model for different configurations. It is still considered to be a useful attempt and deserves more attention.

E. PROMOTION IN MULTIMEDIA CONTENT SERVICES

For the next generation networks, the user experience will dramatically promote the development and integration of mobile multimedia industrial chains. Multimedia content services such as mobile video, AR/VR, and mobile games will surely be closely related to people's daily lives. In this section, we present the technical works on multimedia content services.

GENI (the Global Environment for Networking Innovation) [140], is a distributed virtual laboratory for transformative, at-scale experiments in network science, services, and security. In addition, it is able to implement a wide variety of experiments in a range of areas, such as protocol design and evaluation, distributed service offerings, content management, and in-network service deployment. Patman *et al*. [141] discusses the design and implementation considerations when deploying image processing services on SDN-operated hardware in GENI. In addition, an exploratory test-bed uses the proposed DL-based image processing service for representing a compute-intensive fog service. It is considered a viable option for bringing responsive and state-of-the-art visual processing services to the network-edge. Reference [142] introduces an additional level of complexity for measuring perceived video quality, as it varies the video bitrates. This work optimizes the QoE for video streaming in SDN networks considering the variety of devices, video parameters and the network requirements. The optimization problem of QoE is modeled based on several parameters that effect the user perception such as stall number and bitrates.

Faced with a vast amount of emerging multimedia services, it is suggested that exploring big data analytics to advance edge caching capabilities is a promising approach to improve network efficiency and alleviate the high demand for the network resources and a hierarchical collaborative edge caching structure was discussed in [143]. In addition, a DL-based Content Popularity Prediction (DLCPP) [144] is proposed for steady improvement in caching performance over other dominant cache management frameworks.

From our research, we found there are relatively fewer studies on multimedia content services, most of them are mainly focused on network structure and performance optimization. However, the existing research has been shown effective and has good results. With the continuous development and innovation of network technology, the research directions are promising.

F. DISCUSSIONS OF SDN-CONCEPT NETWORK APPLICATIONS WITH ML METHODS

In Section III, we list the five main cases in SDN-concept network applications with ML methods. As we know, the greatest advantages of SDN are to realize network virtualization and improve the efficiency of resource use. The central controller can obtain, manage and allocate full-network resources from a global perspective. ML methods, as an important implementation method, makes it possible to automatically manage resources and control traffic flow on demand, as stated in Part A and B of this section. Additionally, network traffic recognition and control have been hot topics in recent years, and provide an adaptive and intelligent method for processing gigabits (or even more) of data per second for future applications, especially in situations with high network bandwidth. In Part C, we list three important and prevalent types of applications for guaranteeing network security that can protect network information dynamically. The difference between ML and traditional methods for NIDS is that it turns network abnormal behavior recognition into a pattern recognition problem, and distinguishes normal and abnormal behavior by analyzing network traffic characteristics and host record data. Compared with fault diagnosis, attack detection has attracted more research attention. In Part D, we discuss related developments at the system architecture level, or for the overall performance improvement of systems, and in this sense, the content in Part A can also be included in Part D. Finally, we present the work in Part A separately to highlight the related research, considering its rapid development. Lastly, our emphasis changes to multimedia services in Part E (which seems to depend on the aforementioned content), because of its important role in the future networks. In addition, there is relatively little research in Part E, although the research on the above four parts is in full swing.

Our findings indicate that, although studies on each part vary in terms of speed and performance, rapid development has already appeared overall. We believe that there will be more research on high-level applications with the continuous development of infrastructure, resource management, and other basic applications.

IV. FUTURE ML IN FUTURE SDN

To date, ML has gone through three stages. In the 1980s, connectionism was popular, representing work with Perceptron and NN. In the 1990s, statistical learning methods began to occupy the mainstream stage, and representative methods included SVM. In the 21st century, DNN was proposed, and connectionism returned to the forefront. With the rapid improvements in data volumes and computing power, many AI applications based on DL are gradually maturing.

A. FUTURE ML IN SDN-CONCEPT NETWORKS

There are many commonly used methods in ML. The selection of methods depends to a large extent on the data you have and its characteristics, your training goals, and especially the specific use scenario. Therefore, for better performance in an SDN-concept field, the particularity of the scenarios is an essential consideration factor.

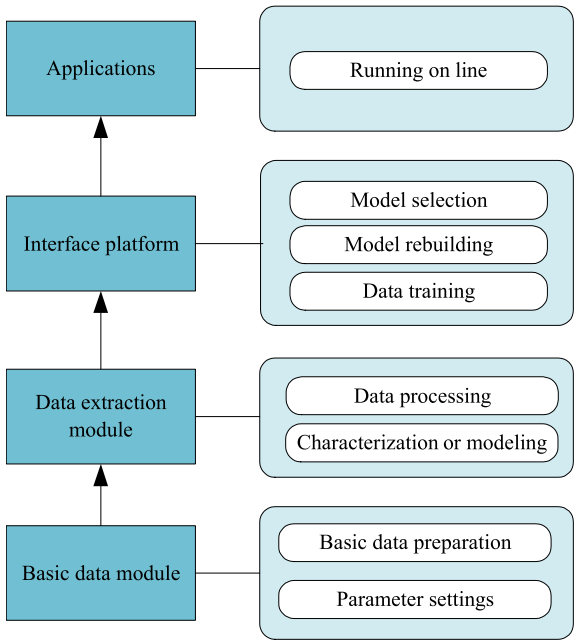

FIGURE 5. The abstract hierarchical structure of ML.

We list an abstract hierarchical structure of ML in Fig. 5. From the bottom to the top layer, the first is a basic data module, including parameter settings and basic data preparation; the second is a data collection module containing data processing and characterization or modeling, and then there is an interface tool and directly used services for the consumer. It has been shown that no matter which layer is being used, it is necessary to rely on methods to operationalize the above intelligent applications above.

1) ISSUE ON DATASETS

In Fig. 5, we place the basic data in the bottom layer. The availability of high quality datasets is considered an important challenge for ML [3], [109], and the concerns further include data sources, labels, classes balance, etc. [11], [145]. If the characteristics of the data sources are neglected during the data training period, the performance or results achieved during testing will seldom reflect the real systems.

In fact, the data obtained can rarely be used directly, because there may be some issues in the sample data, such as missing attributes, unlabeled data, too many attributes, not

enough attributes, test and validation data not separated, and imbalances in different categories.

A dataset determines the upper limit of the ML results. In general, the datasets should be representative of the corresponding network architecture, and extraction of high-quality data reflecting those characteristics will be an important research direction in the future.

2) ISSUE ON PROBLEM REPRESENTING

Representing or defining a problem, means finding out what you want to solve. Without sufficient analysis of the problem, it is possible to be overwhelmed when looking for a method or model. The importance of this step, for the entire process, is obvious. If a directional error has been made at the beginning of the solution, the results will be invalid. For example, the use of a classification algorithm to solve a clustering problem will make obtaining correct results impossible.

Another puzzle is overconfidence in applying powerful algorithms. For example, it is possible that a logistic regression can achieve even better results in some cases, though any classification problem is solved by SVM. So, first of all, understanding the nature of the problem is of paramount importance in some cases.

Representing a specified problem with mathematical language is more intuitive and convenient to presentation and finding solutions. Representing a practical problem as a mathematical problem is a key turning point for ML. Understanding problems deeply will avoid many detours, since feature engineering and model training in ML are very time-consuming.

In this paper, we present multiple ML applications in SDN-concept network scenarios in Section II and III. SDN, a new generation of smart, dynamic, open, customized, and fast innovation networks, covers a wide range of applications. Accurate and abstract representations of problems will still be an indispensable issue.

3) ISSUE ON MODEL BUILDING AND OPTIMIZING

After a problem is represented, a model is referred to as a mathematical model for describing the objective world, and it is abstracted from the data. In data analysis, we usually have only data at hand and then try to derive rules from the data. Here, the rule represents the model we want.

The results may also be very different with different models, even with the same algorithms. Taking the polynomial regression algorithm as an example, for a sample set, we can list multi-order hypothetical functions and then find the most suitable one. Furthermore, building a model does not mean successful results, since some factors will affect the final results of the model, such as the number of features, the number of samples, and the regularization parameter.

There needs to be a series of standards to prove that one model is better than other models. This is the strategy. Different strategies correspond to the compare and selection criteria of different models. The optimized results are also different with different strategies from different people, which led to the fact that there are a variety of models and solutions to the same problem.

4) ISSUE ON ALGORITHM SELECTION

In this paper, we have listed representative ML algorithms, which can be used in many aspects of SDN-concept networks. It is known that ML is conducted for a model, which is implemented through algorithms.

In this part, we will introduce other issues for algorithms. For example, what if the foundation is not strong, and people do not understand the most commonly used model; what if there is a lack of understanding of the practical problems applicable to the model and lack of experience in applying the model to practical problems? For any method, what we should know clearly is not only its nature but also the application scenarios, applicable conditions and limitations of the method.

Which ML algorithm should be used? The answer to this question always depends on the situation. It is certain that complex algorithms should not be selected except under special circumstances. The ML algorithm cheat sheet helps you to choose from a variety of ML algorithms to find the appropriate algorithm for your specific problem [146]. However, in many cases, even an experienced data scientist could not be sure which algorithm will produce the best results before trying the various algorithms. A suitable method selection depends on many factors, such as data size, quality and characteristics, available calculation time, task urgency, and data processing. However, with the limitations of some factors, we can usually choose an appropriate algorithm as our first attempt.

5) ISSUE ON ML FRAMEWORKS INNOVATION

To conveniently develop and implement AI applications, many ML frameworks have emerged. From Section III-D, there is some research focusing on system frameworks. These frameworks provide developers with a good shortcut; some focus on their own availability, and others focus on deployment or parameter optimization.

With the popularization and development of SDN-concept architecture, the algorithms and frameworks based on ML methods have been continuously improved. Researchers and engineers around the world are encouraged to create, share, and even synthesize large-scale new algorithms in this area. At this time, algorithms will also become like a container, capable of arbitrary combinations and extensions, and they will be used to build a general framework suitable for different applications. In other words, multiple ML algorithms may perhaps be combined into a more powerful framework to better analyze the data and fully exploit the value of the data.

6) DISCUSSION OF FUTURE ML ISSUES

In this subsection, we summarize the related issues requiring more attention in the future, with respect to the ML flow. Though we state what appears to be five simple problems,

there are actually many practical aspects to this. Taking the issue of datasets as an example, all dataset-related processes, such as data collection, feature extraction, sample data balance, and abnormal data processing must be considered. Additionally, a correct understanding of such issues will encourage researchers with ML or AI backgrounds to conduct in-depth research on models or algorithms to improve the practicality and effectiveness of ML algorithms in the future.

B. FUTURE SDN-CONCEPT NETWORKS WITH ML

The various, efficient and massive application requirements have necessitated higher requirements for the future network architecture in terms of performance, flexibility, and controllability. However, network innovation is relatively slow and inefficient. It was found that traditional networks have not been able to support the applications. Failure to solve the network problems will not lead to an increase in efficiency and may lead to a decrease.

SDN has become a hot research topic in the global networking field. The early and narrow sense of SDN specifically referred to networks based on the OpenFlow South Bound Interface (SBI). Now SDN tends to refer to a generalized network with SDN concepts; that is, SDN will support more SBIs (such as, NETConf, OVSDB, BGPLS, PCEP, etc.) in addition to OpenFlow, implementing flexible programming, and deploying intelligent analysis and scheduling beyond the traditional routing protocols.

In 2001, IBM's senior vice president of research, Paul Horn, introduced the idea of autonomic computing with self-management, self-configuration, self-optimization, self-protection and self-healing [147], [148]. It is well known that SDN and learning-based Network Analytics (NA) will facilitate the adoption of AI techniques in the context of network operations and controls. Ayoubi *et al*. [84] make the transitions IBM's autonomic element MAPE (Monitor-Analyze-Plan-Execute) to an improved cognitive control loop named C-MAPE. A new notable paradigm is Knowledge-Defined Networking (KDN), which relies on ML and cognitive techniques to operate the network [136]. The KDN paradigm brings significant advantages to networking, and their core ideas are encouraging for computer network research.

It is foreseeable that SDN will reach a new stage in making the system intelligent as far as being self-aware, self-adaptive, and proactive with big data analysis and AI in the future. The SDN-concept impacts both wired and wireless networks, such as optical networks, mobile networks, vehicle networks, Internet of Things (IoT) and wireless sensor networks [149]–[154]. A set of network design and optimization schemes has also been presented, which will not be repeated here. Generally speaking, SDN-concept networks in the future will be characterized by automatic and intelligent functions including: (1) knowledge extraction from network logs, (2) intelligent routing, (3) resource management in SDN scenarios, (4) short and long-term network scheduling, (5) system security and protection.

V. CONCLUSION

Driven by data availability and the theoretical development of ML frameworks, ML is considered to be one of the most promising AI tools for autonomic network operations and management because of its ability to extract knowledge from the data. SDN, a critically important cornerstone of the modern networking architecture, has strength and vitality, especially with the rapid development of networks and big data for future high-level service requirements. Although there have been some surveys of the issues and challenges for ML in various networks based on SDN [3], [26], [70], [84], [123], [155], there has been little evidence of a failure of the applications to achieve practical management solutions for autonomic networks.

In this paper, we discuss the ML applications in SDN-concept networks from two perspectives, namely, the perspective of ML algorithms and the perspective of SDN network applications. In terms of the ML algorithm perspective, we present applications of ML methods in SDN-concept networks, followed by the classifications of ML methods; the common ML algorithms are separately introduced. For the other aspect, we focus on SDN network applications with ML algorithms. In addition, we discuss the future research directions in this area. The main challenges for ML methods are identified. Although some progress has been made in ML fields, effective ML is difficult because of difficult patterns and insufficiently available training data. As a result, many ML programs often fail to show the expected performance. We hope that our discussions may provide a simple guide for the development of SDN and the implementation of a more intelligent network. This work will be helpful for researchers with different objectives to master the key issues in the field.

SDN-concept networks with ML methods will play an important role in all aspects of future network construction and management, including intelligent routing management, resource management, flow control, network security, etc. In the future, we will conduct in-depth studies on the key challenges outlined in the paper.

REFERENCES

[1] K. Benzekki, A. El Fergougui, and A. E. Elalaoui, "Software-defined networking (SDN): A survey," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 5803–5833, 2016.

[2] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, "Are we ready for SDN? Implementation challenges for software-defined networks," *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 36–43, Jul. 2013.

[3] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, 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.

[4] W. Kellerer, P. Kalmbach, A. Blenk, A. Basta, M. Reisslein, and S. Schmid, "Adaptable and data-driven softwarized networks: Review, opportunities, and challenges," *Proc. IEEE*, vol. 107, no. 4, pp. 711–731, Apr. 2019.

[5] S. Nanda, F. Zafari, C. DeCusatis, E. Wedaa, and B. Yang, "Predicting network attack patterns in SDN using machine learning approach," in *Proc. IEEE NFV-SDN*, Nov. 2016, pp. 167–172.

[6] C. Song, Y. Park, K. Golani, Y. Kim, K. Bhatt, and K. Goswami, "Machine-learning based threat-aware system in software defined networks," in *Proc. IEEE ICCCN*, Jul./Aug. 2017, pp. 1–9.

[7] D. Comaneci and C. Dobre, "Securing networks using SDN and machine learning," in *Proc. IEEE CSE*, Oct. 2018, pp. 194–200.
[8] H. Z. Jahromi, A. Hines, and D. T. Delanev, "Towards application-aware networking: ML-based end-to-end application KPI/QoE metrics characterization in SDN," in *Proc. ICUFN*, Jul. 2018, pp. 126–131.
[9] L. Zhu, X. Tang, M. Shen, X. Du, and M. Guizani, "Privacy-preserving DDoS attack detection using cross-domain traffic in software defined networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 628–643, Mar. 2018.
[10] K. Bao, J. D. Matyjas, F. Hu, and S. Kumar, "Intelligent software-defined mesh networks with link-failure adaptive traffic balancing," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 2, pp. 266–276, Jun. 2018.
[11] Y. Yu, L. Guo, Y. Liu, J. Zheng, and Y. Zong, "An efficient SDN-based DDoS attack detection and rapid response platform in vehicular networks," *IEEE Access*, vol. 6, pp. 44570–44579, Sep. 2018.
[12] D. Hu, P. Hong, and Y. Chen, "FADM: DDOS flooding attack detection and mitigation system in software-defined networking," in *Proc. IEEE GLOBECOM*, Dec. 2017, pp. 1–7.
[13] P. Gupta and N. McKeown, "Classifying packets with hierarchical intelligent cuttings," *IEEE Micro*, vol. 20, no. 1, pp. 34–41, Jan. 2000.
[14] S. Singh, F. Baboescu, G. Varghese, and J. Wang, "Packet classification using multidimensional cutting," in *Proc. ACM SIGCOMM*, Aug. 2003, pp. 213–224.
[15] B. Vamanan, G. Voskuilen, and T. N. Vijaykumar, "EffiCuts: Optimizing packet classification for memory and throughput," in *Proc. ACM SIGCOMM*, Sep. 2010, pp. 207–218.
[16] W. Li, X. Li, H. Li, and G. Xie, "CutSplit: A decision-tree combining cutting and splitting for scalable packet classification," in *Proc. IEEE INFOCOM*, Apr. 2018, pp. 2645–2653.
[17] S. Yingchareonthawornchai, J. Daly, A. X. Liu, and E. Torng, "A sorted-partitioning approach to fast and scalable dynamic packet classification," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1907–1920, Aug. 2018.
[18] K. Polat and S. Güneş, "A novel hybrid intelligent method based on C4.5 decision tree classifier and one-against-all approach for multi-class classification problems," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 1587–1592, Mar. 2009.
[19] P. Xiao, W. Qu, H. Qi, Y. Xu, and Z. Li, "An efficient elephant flow detection with cost-sensitive in SDN," in *Proc. IEEE INISCom*, Mar. 2015, pp. 24–28.
[20] D. Chemodanov, P. Calyam, S. Valluripally, H. Trinh, J. Patman, and K. Palaniappan, "On QoE-oriented cloud service orchestration for application providers," *IEEE Trans. Services Comput.*, to be published.
[21] B. Leng, L. Huang, C. Qiao, and H. Xu, "A decision-tree-based on-line flow table compressing method in Software Defined Networks," in *Proc. IEEE/ACM IWQoS*, Jun. 2016, pp. 1–2.
[22] M. Latah and L. Toker, "Towards an efficient anomaly-based intrusion detection for software-defined networks," *IET Netw.*, vol. 7, no. 6, pp. 453–459, Nov. 2018.
[23] R. Gomes, M. Ahsan, and A. Denton, "Random forest classifier in SDN framework for user-based indoor localization," in *Proc. IEEE EIT*, May 2018, pp. 537–542.
[24] T.-H. Lei, Y.-T. Hsu, I-C. Wang, and C. H.-P. Wen, "Deploying QoS-assured service function chains with stochastic prediction models on VNF latency," in *Proc. IEEE NFV-SDN*, Nov. 2017, pp. 1–6.
[25] Q. Cheng, C. Wu, H. Zhou, Y. Zhang, R. Wang, and W. Ruan, "Guarding the perimeter of cloud-based enterprise networks: An intelligent SDN firewall," in *Proc. IEEE HPCC/SmartCity/DSS*, Jun. 2018, pp. 897–902.
[26] D. Côté, "Using machine learning in communication networks," *J. Opt. Commun. Netw.*, vol. 10, no. 10, pp. D100–D109, Oct. 018.
[27] L. Barki, A. Shidling, N. Meti, D. G. Narayan, and M. M. Mulla, "Detection of distributed denial of service attacks in software defined networks," in *Proc. IEEE ICACCI*, Sep. 2016, pp. 2576–2581.
[28] P.-C. Lin, P.-C. Li, and V. L. Nguyen, "Inferring openflow rules by active probing in software-defined networks," in *Proc. IEEE ICACT*, Feb. 2017, pp. 415–420.
[29] H. Kuang, Y. Qiu, R. Li, and X. Liu, "A hierarchical K-means algorithm for controller placement in SDN-Based WAN architecture," in *Proc. ICMTMA*, Feb. 2018, pp. 263–267.
[30] L. Zhu, R. Chai, and Q. Chen, "Control plane delay minimization based SDN controller placement scheme," in *Proc. WCSP*, Oct. 2017, pp. 1–6.
[31] B. P. R. Killi, E. A. Reddy, and S. V. Rao, "Cooperative game theory based network partitioning for controller placement in SDN," in *Proc. COMSNETS*, Jan. 2018, pp. 105–112.
[32] G. Wang, Y. Zhao, J. Huang, Q. Duan, and J. Li, "A K-means-based network partition algorithm for controller placement in software defined network," in *Proc. IEEE ICC*, May 2016, pp. 1–6.
[33] Z. Fan and R. Liu, "Investigation of machine learning based network traffic classification," in *Proc. ISWCS*, Aug. 2017, pp. 1–6.
[34] 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 SCC*, San Francisco, CA, USA, Jun./Jul. 2016, pp. 760–765.
[35] Z. Zhang, L. Ma, K. K. Leung, L. Tassiulas, and J. Tucker, "Q-placement: Reinforcement-learning-based service placement in software-defined networks," in *Proc. IEEE ICDCS*, Jul. 2018, pp. 1527–1532.
[36] L. S. R. Sampaio, P. H. A. Faustini, A. S. Silva, L. Z. Granville, and A. Schaeffer-Filho, "Using NFV and reinforcement learning for anomalies detection and mitigation in SDN," in *Proc. IEEE ISCC*, Jun. 2018, pp. 432–437.
[37] C. Yu, J. Lan, Z. Guo, and Y. Hu, "DROM: Optimizing the routing in software-defined networks with deep reinforcement learning," *IEEE Access*, vol. 6, pp. 64533–64539, Oct. 2018.
[38] C. Wang, L. Zhang, Z. Li, and C. Jiang, "SDCoR: Software defined cognitive routing for Internet of vehicles," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3513–3520, Oct. 2018.
[39] J. Chavula, M. Densmore, and H. Suleman, "Using SDN and reinforcement learning for traffic engineering in UbuntuNet alliance," in *Proc. ICACCE*, Nov. 2016, pp. 349–355.
[40] S. Sendra, A. Rego, J. Lloret, J. M. Jimenez, and O. Romero, "Including artificial intelligence in a routing protocol using software defined networks," in *Proc. IEEE ICC*, May 2017, pp. 670–674.
[41] F. Francois and E. Gelenbe, "Optimizing secure SDN-enabled inter-data centre overlay networks through cognitive routing," in *Proc. IEEE MASCOTS*, Sep. 2016, pp. 283–288.
[42] P. Tang, F. Li, W. Zhou, W. Hu, and L. Yang, "Efficient auto-scaling approach in the telco cloud using self-learning algorithm," in *Proc. IEEE GLOBECOM*, Dec. 2015, pp. 1–6.
[43] T. Daher, S. B. Jemaa, and L. Decreusefond, "Softwarized and distributed learning for SON management systems," in *Proc. IEEE/IFIP NOMS*, Apr. 2018, pp. 1–7.
[44] X. Huang, T. Yuan, G. Qiao, and Y. Ren, "Deep reinforcement learning for multimedia traffic control in software defined networking," *IEEE Netw.*, vol. 32, no. 6, pp. 35–41, Nov./Dec. 2018.
[45] 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," 2018, *arXiv:1810.07862*. [Online]. Available: https://arxiv.org/abs/1810.07862
[46] Y. He, Z. Zhang, and Y. Zhang, "A big data deep reinforcement learning approach to next generation green wireless networks," in *Proc. IEEE GLOBECOM*, Dec. 2017, pp. 1–6.
[47] C. Qiu, F. R. Yu, H. Yao, C. Jiang, F. Xu, and C. Zhao, "Blockchain-based software-defined industrial Internet of Things: A dueling deep Q-learning approach," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4627–4639, Jun. 2019.
[48] D. Zhang, F. R. Yu, and R. Yang, "A machine learning approach for software-defined vehicular ad hoc networks with trust management," in *Proc. IEEE GLOBECOM*, Dec. 2018, pp. 1–6.
[49] S.-C. Lin, I. F. Akyildiz, P. Wang, and M. Luo, "QoS-aware adaptive routing in multi-layer hierarchical software defined networks: A reinforcement learning approach," in *Proc. IEEE SCC*, Jun./Jul. 2016, pp. 25–33.
[50] H. Z. Jahromi and D. T. Delaney, "An application awareness framework based on SDN and machine learning: Defining the roadmap and challenges," in *Proc. IEEE CONTROLO*, Jul. 2018, pp. 411–416.
[51] R. Li, Z. Zhao, Q. Sun, C.-L. I, C. Yang, X. Chen, M. Zhao, and H. Zhang, "Deep reinforcement learning for resource management in network slicing," *IEEE Access*, vol. 6, pp. 74429–74441, Nov. 2018.
[52] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in *Proc. ACM HotNets*, Nov. 2016, pp. 50–56.
[53] R. Mijumbi, J.-L. Gorricho, J. Serrat, M. Claeys, F. De Turck, and S. Latré, "Design and evaluation of learning algorithms for dynamic resource management in virtual networks," in *Proc. IEEE NOMS*, May 2014, pp. 1–9.
[54] R. Li, Z. Zhao, Q. Sun, C.-L. I, C. Yang, X. Chen, M. Zhao, and H. Zhang, "Deep reinforcement learning for resource management in network slicing," 2018, *arXiv:1805.06591*. [Online]. Available: https://arxiv.org/abs/1805.06591

[55] X. Chen, Z. Zhao, C. Wu, M. Bennis, H. Liu, Y. Ji, and H. Zhang, "Multi-tenant cross-slice resource orchestration: A deep reinforcement learning approach," 2018, *arXiv:1807.09350*. [Online]. Available: https://arxiv.org/abs/1807.09350

[56] 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.

[57] Z. Xu, J. Tang, J. Meng, W. Zhang, Y. Wang, C. H. Liu, and D. Yang, "Experience-driven networking: A deep reinforcement learning based approach," in *Proc. IEEE INFOCOM*, Apr. 2018, pp. 1871–1879.

[58] M. Abderrahim, A. B. Letaifa, A. Haji, and S. Tabbane, "How to use MEC and ML to improve resources allocation in SDN networks?" in *Proc. WAINA*, May 2018, pp. 442–447.

[59] M. Abderrahim, A. B. Letaifa, A. Haji, and S. Tabbane, "A game theory-based effective network management in SDN networks," in *Proc. WAINA*, May 2018, pp. 390–395.

[60] H. Jmila, M. I. Khedher, and M. A. El Yacoubi, "Estimating VNF resource requirements using machine learning techniques," in *Proc. ICONIP*, May 2017, pp. 883–892.

[61] 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, 4th Quart., May 2017.

[62] P. Wang, K.-M. Chao, H.-C. Lin, W.-H. Lin, and C.-C. Lo, "An efficient flow control approach for SDN-based network threat detection and migration using Support Vector Machine," in *Proc. IEEE ICEBE*, Nov. 2016, pp. 56–63.

[63] K. Gao, J. Zhang, Y. R. Yang, and J. Bi, "Prophet: Fast accurate model-based throughput prediction for reactive flow in DC networks," in *Proc. IEEE INFOCOM*, Apr. 2018, pp. 720–728.

[64] R. Alvizu, S. Troia, G. Maier, and Pattavina, "Machine-learning-based prediction and optimization of mobile metro-core networks," in *Proc. IEEE SUM*, Jul. 2018, pp. 155–156.

[65] I. Alawe, A. Ksentini, Y. Hadjadj-Aoul, and P. Bertin, "Improving traffic forecasting for 5G core network scalability: A machine learning approach," *IEEE/ACM Trans. Netw.*, vol. 32, no. 6, pp. 42–49, Nov./Dec. 2018.

[66] X. Zhang, C. Wu, Z. Li, and F. C. M. Lau, "Proactive VNF provisioning with multi-timescale cloud resources: Fusing online learning and online optimization," in *Proc. IEEE INFOCOM*, May 2017, pp. 1–9.

[67] A. Baz, "Bayesian machine learning algorithm for flow prediction in SDN switches," in *Proc. ICCAIS*, Apr. 2018, pp. 1–7.

[68] C. Liu, A. Malboubi, and C.-N. Chuah, "OpenMeasure: Adaptive flow measurement & inference with online learning in SDN," in *Proc. IEEE INFOCOM WKSHPS*, Apr. 2016, pp. 47–52.

[69] H. Singh, "Performance analysis of unsupervised machine learning techniques for network traffic classification," in *Proc. ACCT*, Feb. 2015, pp. 401–404.

[70] J. Yan and J. Yuan, "A survey of traffic classification in software defined networks," in *Proc. IEEE HotICN*, Aug. 2018, pp. 200–206.

[71] L. Grimaudo, M. Mellia, E. Baralis, and R. Keralapura, "SeLeCT: Self-learning classifier for Internet traffic," *IEEE Trans. Netw. Service Manag.*, vol. 11, no. 2, pp. 144–157, Jun. 2014.

[72] D. Apiletti, "SeLINA: A self-learning insightful network analyzer," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 3, pp. 696–710, Sep. 2016.

[73] L. Grimaudo, M. Mellia, E. Baralis, and R. Keralapura, "Self-learning classifier for Internet traffic," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 3381–3386.

[74] Y. Hong, C. Huang, B. Nandy, and N. Seddigh, "Iterative-tuning support vector machine for network traffic classification," in *Proc. IFIP/IEEE IM*, May 2015, pp. 458–466.

[75] P. Wang, F. Ye, X. Chen, and Y. Qian, "DataNet: Deep learning based encrypted network traffic classification in SDN home gateway," *IEEE Access*, vol. 6, pp. 55380–55391, Sep. 2018.

[76] J. Xu, J. Wang, Q. Qi, H. Sun, and B. He, "Deep neural networks for application awareness in SDN-based network," in *Proc. IEEE MLSP*, Sep. 2018, pp. 1–6.

[77] P. Amaral, J. Dinis, P. Pinto, L. Bernardo, J. Tavares, and H. S. Mamede, "Machine learning in software defined networks: Data collection and traffic classification," in *Proc. IEEE ICNP*, Nov. 2016, pp. 1–5.

[78] N. F. Huang, C.-C. Li, C.-H. Li, C.-C. Chen, C.-H. Chen, and I-H. Hsu, "Application identification system for SDN QoS based on machine learning and DNS responses," in *Proc. IEEE APNOMS*, Sep. 2017, pp. 407–410.

[79] T. V. Phan and M. Park, "Efficient distributed denial-of-service attack defense in SDN-based cloud," *IEEE Access*, vol. 7, pp. 18701–18714, Jan. 2019.

[80] R. Thupae, B. Isong, N. Gasela, and A. M. Abu-Mahfouz, "Machine learning techniques for traffic identification and classifiacation in SDWSN: A survey," in *Proc. IEEE IECON*, Oct. 2018, pp. 4645–4650.

[81] M. Uddin and T. Nadeem, "TrafficVision: A case for pushing software defined networks to wireless edges," in *Proc. IEEE MASS*, Oct. 2016, pp. 37–46.

[82] A. Dawoud, S. Shahristani, and C. Raun, "A deep learning framework to enhance software defined networks security," in *Proc. IEEE WAINA*, May 2018, pp. 709–714.

[83] 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, Jan. 2019.

[84] S. Ayoubi, N. Limam, M. A. Salahuddin, N. Shahriar, R. Boutaba, F. Estrada-Solano, and O. M. Caicedo, "Machine learning for cognitive network management," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 158–165, Jan. 2018.

[85] A. Abubakar and B. Pranggono, "Machine learning based intrusion detection system for software defined networks," in *Proc. IEEE EST*, Sep. 2017, pp. 138–143.

[86] M. Wang and S. Handurukande, "A streaming data anomaly detection analytic engine for mobile network management," in *Proc. IEEE ICCDC*, Jul. 2016, pp. 722–729.

[87] Q. Schueller, K. Basu, M. Younas, M. Patel, and F. Ball, "A hierarchical intrusion detection system using support vector machine for SDN network in cloud data center," in *Proc. ITNAC*, Nov. 2018, pp. 1–6.

[88] N. Meti, D. G. Narayan, and V. P. Baligar, "Detection of distributed denial of service attacks using machine learning algorithms in software defined networks," in *Proc. IEEE ICACCI*, Sep. 2017, pp. 1366–1371.

[89] S. K. Dey, M. M. Rahman, and M. R. Uddin, "Detection of flow based anomaly in openflow controller: Machine learning approach in software defined networking," in *Proc. IEEE iCEEiCT*, Sep. 2018, pp. 416–421.

[90] S. Garg, K. Kaur, N. Kumar, and J. J. P. C. Rodrigues, "Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in SDN: A social multimedia perspective," *IEEE Trans. Multimedia*, vol. 21, no. 3, pp. 566–578, Mar. 2019.

[91] L. Boero, M. Marchese, and S. Zappatore, "Support vector machine meets software defined networking in ids domain," in *Proc. IEEE ITC*, Sep. 2017, pp. 25–30.

[92] C. Lee, C. Yoon, S. Shin, and S. K. Cha, "INDAGO: A new framework for detecting malicious SDN applications," in *Proc. IEEE ICNP*, Sep. 2018, pp. 220–230.

[93] Q. Shafi, A. Basit, S. Qaisar, A. Koay, and I. Welch, "Fog-assisted SDN controlled framework for enduring anomaly detection in an IoT network," *IEEE Access*, vol. 6, pp. 73713–73723, Nov. 2018.

[94] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," in *Proc. IEEE WINCOM*, Oct. 2016, pp. 258–263.

[95] E. Unal, S. Sen-Baidya, and R. Hewett, "Towards prediction of security attacks on software defined networks: A big data analytic approach," in *Proc. IEEE Big Data*, Dec. 2018, pp. 4582–4588.

[96] T.-H. Nguyen and M. Yoo, "Analysis of link discovery service attacks in SDN controller," in *Proc. IEEE ICOIN*, Jan. 2017, pp. 259–261.

[97] M. Z. Masoud, Y. Jaradat, and I. Jannoud, "On preventing ARP poisoning attack utilizing software defined network (SDN) paradigm," in *Proc. IEEE AEECT*, Nov. 2015, pp. 1–5.

[98] K. Kalkan, L. Altay, G. Gür, and F. Alagöz, "JESS: Joint entropy-based DDoS defense scheme in SDN," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2358–2372, Oct. 2018.

[99] A. Sangodoyin, B. Modu, I. Awan, and J. P. Disso, "An approach to detecting distributed denial of service attacks in software defined networks," in *Proc. IEEE FiCloud*, Aug. 2018, pp. 436–443.

[100] L. Yang and H. Zhao, "DDoS attack identification and defense using SDN based on machine learning method," in *Proc. IEEE I-SPAN*, Oct. 2018, pp. 174–178.

[101] J. Ashraf and S. Latif, "Handling intrusion and DDoS attacks in software defined networks using machine learning techniques," in *Proc. IEEE NSEC*, Nov. 2014, pp. 55–60.

[102] S. S. Mohammed, R. Hussain, O. Senko, B. Bimaganbetov, J. Lee, F. Hussain, C. A. Kerrache, E. Barka, and M. Z. A. Bhuiyan, "A new machine learning-based collaborative DDoS mitigation mechanism in software-defined network," in *Proc. IEEE WiMob*, Oct. 2018, pp. 1–8.

[103] R. Koning, B. de Graaff, C. de Laat, R. Meijer, and P. Grosso, "Interactive analysis of SDN-driven defence against distributed denial of service attacks," in *Proc. IEEE NetSoft*, Jun. 2016, pp. 483–488.

[104] S. Gangadhar and J. P. G. Sterbenz, "Machine learning aided traffic tolerance to improve resilience for software defined networks," in *Proc. RNDM*, Sep. 2017, pp. 1–7.

[105] M. M. Oo, S. Kamolphiwong, and T. Kamolphiwong, "The design of SDN based detection for distributed denial of service (DDoS) attack," in *Proc. IEEE ICSEC*, Nov. 2017, pp. 1–5.

[106] Z. Chen, F. Jiang, Y. Cheng, X. Gu, W. Liu, and J. Peng, "XGBoost classifier for DDoS attack detection and analysis in SDN-based cloud," in *Proc. IEEE BigComp*, Jan. 2018, pp. 251–256.

[107] S. S. Bhunia and M. Gurusamy, "Dynamic attack detection and mitigation in IoT using SDN," in *Proc. IEEE ITNAC*, Nov. 2017, pp. 1–6.

[108] B. He, F. Zou, and Y. Wu, "Multi-SDN based cooperation scheme for DDoS attack defense," in *Proc. SSIC*, Oct. 2018, pp. 1–7.

[109] J. N. Bakker, B. Ng, and W. K. G. Seah, "Can machine learning techniques be effectively used in real networks against DDoS attacks?" in *Proc. IEEE ICCCN*, Jul./Aug. 2018, pp. 1–6.

[110] F. Benayas, A. Carrera, and C. A. Iglesias, "Towards an autonomic Bayesian fault diagnosis service for SDN environments based on a big data infrastructure," in *Proc. IEEE SDS*, Apr. 2018, pp. 7–13.

[111] D. Rafique, T. Szyrkowiec, H. Grießer, A. Autenrieth, and J.-P. Elbers, "Cognitive assurance architecture for optical network fault management," *J. Lightw. Technol.*, vol. 36, no. 7, pp. 1443–1450, Apr. 1, 2018.

[112] D. Rafique, T. Szyrkowiec, A. Autenrieth, and J.-P. Elbers, "Analytics-driven fault discovery and diagnosis for cognitive root cause analysis," in *Proc. OFC*, Mar. 2018, pp. 1–3.

[113] L. J. Jagadeesan and V. Mendiratta, "Programming the network: Application software faults in software-defined networks," in *Proc. IEEE ISSREW*, Oct. 2016, pp. 125–131.

[114] A. Malik, B. Aziz, C.-H. Ke, H. Liu, and M. Adda, "Virtual topology partitioning towards an efficient failure recovery of software defined networks," in *Proc. ICMLC*, Jul. 2017, pp. 646–651.

[115] C. Natalino, F. Coelho, G. Lacerda, A. Braga, L. Wosinska, and P. Monti, "A proactive restoration strategy for optical cloud networks based on failure predictions," in *Proc. IEEE ICTON*, Jul. 2018, pp. 1–5.

[116] T. N. Nguyen, "The challenges in SDN/ML based network security: A survey," 2018, *arXiv:1804.03539*. [Online]. Available: https://arxiv.org/abs/1804.03539

[117] A. Guo, C. Yuan, G. He, and L. Xu, "Research on SDN/NFV network traffic management and optimization based on big data and artificial intelligence," in *Proc. IEEE ISCIT*, Sep. 2018, pp. 377–382.

[118] R. W. Thomas, D. H. Friend, L. A. DaSilva, and A. B. MacKenzie, "Cognitive networks: Adaptation and learning to achieve end-to-end performance objectives," *IEEE Commun. Mag.*, vol. 44, no. 12, pp. 51–57, Dec. 2006.

[119] R. W. Thomas, L. A. DaSilva, and A. B. Mackenzie, "Cognitive networks," in *Proc. IEEE DySPAN*, Nov. 2005, pp. 352–360.

[120] M. Lee, X. Ye, D. Marconett, S. Johnson, R. Vemuri, and S. J. B. Yoo, "Autonomous network management using cooperative learning for network-wide load balancing in heterogeneous networks," in *Proc. IEEE GLOBECOM*, Nov./Dec. 2008, pp. 1–5.

[121] L. Xu, H. Assem, I. G. B. Yahia, T. S. Buda, A. Martin, D. Gallico, M. Biancani, A. Pastor, P. A. Aranda, M. Smirnov, D. Raz, O. Uryupina, A. Mozo, B. Ordozgoiti, M.-I. Corici, P. O'Sullivan, and R. Mullins, "CogNet: A network management architecture featuring cognitive capabilities," in *Proc. IEEE EuCNC*, Jun. 2016, pp. 325–329.

[122] M. Zorzi, A. Zanella, A. Testolin, M. De F. De Grazia, and M. Zorzi, "Cognition-based networks: A new perspective on network optimization using learning and distributed intelligence," *IEEE Access*, vol. 3, pp. 1512–1530, Aug. 2015.

[123] D. Rafique and L. Velasco, "Machine learning for network automation: Overview, architecture, and applications [invited tutorial]," *J. Opt. Commun. Netw.*, vol. 10, no. 10, pp. D126–D143, Oct. 2018.

[124] K. K. Budhraja, A. Malvankar, M. Bahrami, C. Kundu, A. Kundu, and M. Singhal, "Risk-based packet routing for privacy and compliance-preserving SDN," in *Proc. IEEE CLOUD*, Jun. 2017, pp. 761–765.

[125] V. Agarwal, S. Saraswat, H. P. Gupta, and T. Dutta, "A traffic engineering framework for maximizing network revenue in software defined network," in *Proc. COMSNETS*, Jan. 2018, pp. 531–533.

[126] M. He, P. Kalmbach, A. Blenk, W. Kellerer, and S. Schmid, "Algorithm-data driven optimization of adaptive communication networks," in *Proc. IEEE ICNP*, Oct. 2017, pp. 1–6.

[127] C. Sieber, A. Basta, A. Blenk, and W. Kellerer, "Online resource mapping for SDN network hypervisors using machine learning," in *Proc. IEEE NetSoft*, Jun. 2016, pp. 78–82.

[128] C. Sieber, A. Obermair, and W. Kellerer, "Online learning and adaptation of network hypervisor performance models," in *Proc. IFIP/IEEE IM*, May 2017, pp. 1204–1212.

[129] S. T. V. Pasca, S. S. P. Kodali, and K. Kataoka, "AMPS: Application aware multipath flow routing using machine learning in SDN," in *Proc. IEEE NCC*, Mar. 2017, pp. 1–6.

[130] B. G. Assefa and O. Ozkasap, "MER-SDN: Machine learning framework for traffic aware energy efficient routing in SDN," in *Proc. IEEE DASC/PiCom/DataCom/CyberSciTech*, Aug. 2018, pp. 974–980.

[131] J. A. Boyan and M. L. Littman, "Packet routing in dynamically changing networks: A reinforcement learning approach," in *Proc. NIPS*, Dec. 1994, pp. 671–678.

[132] J. Pei, P. Hong, and D. Li, "Virtual network function selection and chaining based on deep learning in SDN and NFV-enabled networks," in *Proc. IEEE ICC Workshops*, May 2018, pp. 1–6.

[133] B. Mao, F. Tang, Z. M. Fadlullah, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "A novel non-supervised deep-learning-based network traffic control method for software defined wireless networks," *IEEE Wireless Commun.*, vol. 25, no. 4, pp. 74–81, Aug. 2018.

[134] G. Stampa, M. Arias, D. Sanchez-Charles, V. Muntes-Mulero, and A. Cabellos, "A deep-reinforcement learning approach for software-defined networking routing optimization," 2017, *arXiv:1709.07080*. [Online]. Available: https://arxiv.org/abs/1709.07080

[135] A. Mestres, E. Alarcón, and A. Cabellos, "A machine learning-based approach for virtual network function modeling," in *Proc. IEEE WCNCW*, Apr. 2018, pp. 237–242.

[136] A. Mestres, A. Rodriguez-Natal, J. Carner, P. Barlet-Ros, E. Alarcón, M. Solé, V. Muntés-Mulero, D. Meyer, S. Barkai, M. J. Hibbett, G. Estrada, K. Ma'ruf, F. Coras, V. Ermagan, H. Latapie, C. Cassar, J. Evans, F. Maino, J. Walrand, and A. Cabellos, "Knowledge-defined networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 47, no. 3, pp. 2–10, Sep. 2017.

[137] J. Carner, A. Mestres, E. Alarcón, and A. Cabellos, "Machine learning-based network modeling: An artificial neural network model vs a theoretical inspired model," in *Proc. IEEE ICUFN*, Jul. 2017, pp. 522–524.

[138] R. Fan and Y. Chang, "Machine learning for black-box fuzzing of network protocols," in *Proc. ICICS*, Dec. 2017, pp. 621–632.

[139] T. Abar, A. B. Letaifa, and S. El Asmi, "Machine learning based QoE prediction in SDN networks," in *Proc. IEEE IWCMC*, Jun. 2017, pp. 1395–1400.

[140] M. Berman, J. S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, and I. Seskar, "GENI: A federated testbed for innovative network experiments," *Comput. Netw.*, vol. 61, pp. 5–23, Mar. 2014.

[141] J. Patman, M. Alfarhood, S. Islam, M. Lemus, P. Calyam, and K. Palaniappan, "Predictive analytics for fog computing using machine learning and GENI," in *Proc. IEEE INFOCOM WKSHPS*, Apr. 2018, pp. 790–795.

[142] T. Abar, A. B. Letaifa, and S. Elasmi, "Enhancing QoE based on machine learning and DASH in SDN networks," in *Proc. IEEE WAINA*, May 2018, pp. 258–263.

[143] Z. Chang, L. Lei, Z. Zhou, S. Mao, and T. Ristaniemi, "Learn to cache: Machine learning for network edge caching in the big data era," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 28–35, Jun. 2018.

[144] W.-X. Liu, J. Zhang, Z.-W. Liang, L.-X. Peng, and J. Cai, "Content popularity prediction and caching for ICN: A deep learning approach with SDN," *IEEE Access*, vol. 6, pp. 5075–5089, Dec. 2018.

[145] C. A. Hammerschmidt, S. Garcia, S. Verwer, and R. State, "Reliable machine learning for networking: Key issues and approaches," in *Proc. IEEE LCN*, Oct. 2017, pp. 167–170.

[146] H. Li. (Apr. 2017). *Which Machine Learning Algorithm Should I Use?* [Online]. Available: https://blogs.sas.com/content/subconsciousmusings/2017/04/12/machine-learning-algorithm-use/

[147] H. Paul. (Oct. 2001). *Autonomic Computing: IBM's Perspective on the State of Information Technology*. [Online]. Available: http://www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf

[148] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, Jan. 2003.

[149] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, Apr. 2017.

[150] R. Casellas, R. Martínez, L. Velasco, R. Vilalta, P. Pavón, D. King, and R. Muñoz, "Enabling data analytics and machine learning for 5G services within disaggregated multi-layer transport networks," in *Proc. IEEE ICTON*, Jul. 2018, pp. 1–4.

[151] F. Tang, B. Mao, Z. M. Fadlullah, and N. Kato, "On a novel deep-learning-based intelligent partially overlapping channel assignment in SDN-IoT," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 80–86, Sep. 2018.

[152] Z. Zhao, E. Schiller, E. Kalogeiton, T. Braun, B. Stiller, M. T. Garip, J. Joy, M. Gerla, N. Akhtar, and I. Matta, "Autonomic communications in software-driven networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2431–2445, Nov. 2017.

[153] O. G. Matlou and A. M. Abu-Mahfouz, "Utilising artificial intelligence in software defined wireless sensor network," in *Proc. IEEE IECON*, Oct./Nov. 2017, pp. 6131–6136.

[154] 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," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 10, no. 10, pp. D52–D62, Oct. 2018.

[155] T. S. Buda, H. Assem, L. Xu, D. Raz, U. Margolin, E. Rosensweig, D. R. Lopez, M.-I. Corici, M. Smirnov, R. Mullins, O. Uryupina, A. Mozo, B. Ordozgoiti, A. Martin, A. Alloush, P. O'Sullivan, and I. G. B. Yahia, "Can machine learning aid in delivering new use cases and scenarios in 5G?" in *Proc. IEEE/IFIP NOMS*, Apr. 2016, pp. 1279–1284.

YANLING ZHAO received the B.S. degree from the School of Information Science and Engineering, Shandong University, Jinan, China, in 2005. Since 2005, she has been involved in communication networks at the Shandong Computer Science Center, Jinan. She became an Associate Research Fellow, in 2012. Her research interests include communication networks, machine learning, and multimedia communication.

YE LI received the B.S. and Ph.D. degrees from Tsinghua University, China, in 2004 and 2009, respectively. He is currently a Professor with the National Supercomputer Center, Jinan, where he is also the Director of the Shandong Provincial Key Laboratory of Computer Networks. His current research interests include multimedia communications, speech compression, and network protocol. He was a recipient of the Shandong Scientific and Technological Progress Award, in 2013.

XINCHANG ZHANG (M'14) received the Ph.D. degree from the Computer Network Information Center, Chinese Academy of Sciences, Beijing, China, in 2010. He is currently an Associate Professor with the Shandong Computer Science Center (National Supercomputer Center in Jinan), Jinan, China, where he is also the Director of the Internet Laboratory. His research interests include network protocols and architectures, and cloud computing.

GUANGGANG GENG received the Ph.D. degree from the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. He joined the Computer Network Information Center, Chinese Academy of Sciences, in 2008. He is currently a Professor with the China Internet Network Information Center. His current research interests include machine learning, adversarial information retrieval on the web, and web search.

WEI ZHANG (M'17) received the B.E. degree from Zhejiang University, in 2004, the M.S. degree from Liaoning University, in 2008, and the Ph.D. degree from the Shandong University of Science and Technology, in 2018. Since 2008, he has been a Researcher with the Shandong Computer Science Center (National Supercomputer Center in Jinan). His research interest includes network architectures and protocols. He has over 20 research papers in these research areas.

YANJIE SUN received the B.S. degree from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2007. He has passed the CCIE Certification and the AACTP Certification. He joined the Branch Training Center in Shandong Province, China United Network Telecommunications Group, in 2010. He is currently a Senior Engineer. His current research include software-defined networking (SDN) networks, 5G networks, and transmission network IPRAN.