Application of Machine Learning in Software Defined Networks

Zeinab Sadeghian and Hessam Hashemizadeh
Isfahan University of Technology

Abstract— **In recent years, the rapid growth of internet and communication technologies, devices and resources in existing networks are getting more complicated and diverse. In order to manage all these devices and maintain and optimize them efficiently, networks need to be more intelligent. Due to the distributed attribute of old networks, Machine Learning algorithms had been hard to be applied in networks. Nowadays, Software Defined Networks (SDN) has eased the implementation of the algorithms in networks. In this survey, the first SDN is perused, then briefly some of Machine Learning algorithms are discussed. Finally, the application of these algorithms in different network issues is reviewed. Some of the investigated issues are Routing, Traffic Classification, Quality of Service and Quality of Experience.**

I. INTRODUCTION

By improvement and development of Smart Devices (like smartphones, smart houses) and Network Devices(like Cloud Computing), data and data traffic are growing exponentially and rapidly. On the other hand, different devices are used in networks and these devices use different protocols and communicate differently with each other. Some communication means are Wifi, WiMax, LTE, Bluetooth, etc. These differences enhance the network's complexity and divergence. Moreover, it creates many challenges for managing and optimizing the networks and data in them. Deploying more automation and perspicacity in networks is a way to overcome these challenges. In the past few years, there have existed some efforts to make the networks more intelligent. However none of them was that successful due to networks being distributed. Learning from Routers, which are like nodes in the Network Graph, limited our view of the network and can only act on small parts of it, also limiting our global intelligence.

In Software Defined Networks, Data Plane and Control Plane is separated. Consequently, in Software Defined Networks, a Logically Centralized Controller, which acts as the Networking Operating System (NOS), manages resources in the network. Furthermore, this logically centralized controller has a global view of the network; therefore, not only it can collect real-time data and state of the network, but it is also able to program the resources in the network dynamically. Using Machine Learning algorithms and techniques in Software Defined Networks is suitable due to the following reasons:

A) Logically Localized Software Defined Networks has a global view of the network, therefore it can collect different data on a big scale, which is the main requirement of Machine Learning algorithms.
B) In recent years advances in Tensor Processing Unit (TPU) and Graphics Processing Unit (GPU) have paved the way to implement and apply Machine Learning algorithms in the network.
C) Relying on real-time and historical data, which can be collected due to the global view of the Network provided from the Controller, Machine Learning algorithms can analyze these data to add more intelligence to the network and automate network services.

In this paper, the brief review of Software Defined Networks architecture followed by some Machine Learning algorithms, which can be applied in Software Defined Networks, are discussed. The rest of the article is partitioned in three parts: First, Software Defined Networks Architecture is presented in section II, then Machine Learning Techniques are briefly introduced in section III, And finally challenges of the networks and application of Machine Learning in each is discussed in section IV. At last, this survey is concluded in section V.

II. SOFTWARE DEFINED NETWORK'S ARCHITECTURE

A. Data plane

Data plane is the foundation of Software Defined Networks, similar to conventional routers data plane has forwarding table and forwarding devices which can be physical or virtual, although physical switches have the speed superiority but they lack the resilience, virtual switches can offer. Data plane is also known as infrastructure plane because it's the lowest layer in Software Defined Networks.

There are several acts that each switch can do; they can forward, drop or modify each packet. these actions were viable in conventional routers but what makes Software Defined Networks different is the ability to decide the action in a more flexible way with the policies that are implemented in Control Plane.

B. Control plane

If we assume that SDN is a human being, Control Plane would be the brain that also controls the Nervous System, which are the API. The Logically centralized controller is a middleman between the Data plane and The Application layer it collects the networks states and data from the Data plane and sends it to Application layer in an understandable manner. This Logically centralized controller also transfers the Application planes rules and policies to Data plane which consists of switches and forwarding tables. There are several architectures for controller and a total of Three interfaces for communicating between the layers which are northbound API, southbound API and eastbound/westbound API. Northbound connects control plane to application layer, southbound connects control plane to data plane and finally eastbound/westbound are used when we have more than one controller it establishes connection between multiple controllers.

C. Application Layer

As we mentioned Application layer is on top of control plane. we use northbound API to establish a connection between this layer and control plane to retrieve the data from lower layers such as forwarding table, network states and changes in the system. With using the provided data we can set new rules or analyze the incoming data to enhance the quality and performance of the network. For example, by monitoring the data through time we can predict congestion; therefore we can use traffic engineering techniques to ensure the quality of the network.

III. Machine Learning algorithms

In this section basic concepts of most researched Machine Learning algorithms are discussed. Before starting, Let's define what machine learning is. Machine learning is the study of statistical models that computers use to perform a task without explicit instructions and explicit programming computers to do that task. Machine learning algorithms build a mathematical model, based on a sample data called the Training data-set in order to make decisions or predictions. In fact, In order to create a model, Machine learning algorithms are applied to training data-sets to learn the system model. Afterward, this trained model accepts input and obtains an estimated output for that input. Depending on how creating the model and learning is performed, These algorithms distinguished into four categories:

A) Supervised Learning
B) Unsupervised learning
C) Semi-supervised learning
D) reinforcement learning

Following this section, some of these algorithms are briefly explained. For more information on each of these algortihms, please refer to [1]-[4].

A. Supervised learning

Supervised Learning is a Machine Learning task, which is basically a labeling technique. As the name implies, these algorithms require a supervisor to learn their parameters. These algorithms are given a labeled training data-set (i.e., inputs and known outputs) to build a model that has learned the relation between inputs and outputs. After training, When a new input is given to that model, the trained model is used to get the expected output. Supervised Learning is a very broad domain and has several algorithms. In the following some of the most used algorithms such as k-nearest neighbor, decision tree, random forest, neural network, support vector machine, Bayes' theory, and hidden Markov models are presented:

1) k-nearest neighbor (k-NN): One of the popular methods of supervised learning is K-NN. It is used when the distribution of the observation and the result is not known. The process of this algorithms is very straightforward; It tries to classify a new data sample based on the k nearest neighbors of a certain class that unclassified class has Since the main metric in this algorithm is the distance between the unlabeled sample and classified samples [5]. Some of the common and most used are: Euclidean, Euclidean squared, City-block and Chebyshev. For more information on this algorithm, please refer to [6].

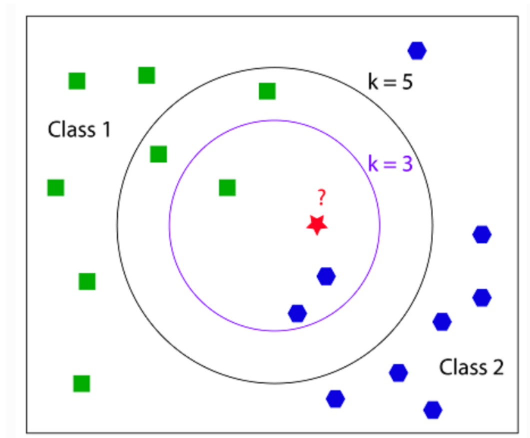

Fig. 1. In this example, the main goal is to classify the red star to be either a green square or a blue octagon. For k=3, it is shown that there are two blue octagons and a green square. Therefore, the red star is classified into the blue octagon class, while for k=5 it is shown that the number of green squares around the star is greater than the number of blue octagons. So the red star is classified into the green squares class. Source: Adapted from [25]

2) Bayes' Theory: Bayes' theorem is an important rule in statistical models and probabilities. This theory uses conditional probability to calculate the probability of an event by knowing the conditions that might be related to that event. Bayes' theorem is stated mathematically as follows:

$$P(\theta|\mathbf{D}) = P(\theta)\frac{P(\mathbf{D}|\theta)}{P(\mathbf{D})} \tag{1}$$

where P(**D**) is not equals to zero, $P(\theta|\mathbf{D})$ is the probability of event θ occurring given that **D** is true, $P(\theta)$ and $P(\mathbf{D})$ are the probabilities of observing θ and **D** respectively, **D** is a new evidence and θ is a hypothesis[43]//

In order to use Bayes' theorem in a classification problem it is assumed that the features of the data samples in training data-set is independent of each other [110]. In these kind of problems the Bayes' theory learns a model by using that training data-set. The evidence **D** is a data sample, and the hypothesis θ is the class which that sample is assigning to. $P(\theta|\mathbf{D})$ indicates the probability of data sample **D** belonging to class θ. To calculate this probability, $P(\mathbf{D}|\theta)$, $P(\mathbf{D})$, $P(\theta)$ need to be known. These probabilities each calculated based on training data-set, which is the process of creating the system model. Finally, the data sample will be assigned to a class with the highest $P(\theta|\mathbf{D})$ probability.

3) Hidden Markov Models (HMM): Statistical Markov model called Hidden Markov Model is a modeling technique that the system being modeled is regarded to be a Markov process with hidden states. It is based on augmenting the Markov chain. A Markov chain is a model which gives of some information on the probabilities of sequence of random variables or states, each of which can take on values from some set. There is strong assumption in Markov chain and that is if we want to predict the future in the sequence, all that matters is the current state. The main difference between HMM and other similar algorithms is that HMM is often applied in environments where system states either partially visible or not visible. [7], [8].

4) Support Vector Machine (SVM): SVM is a supervised ML algorithm which can be used in classification or regression challenges. Given a training set with data samples, each marked as belonging to one or the other of two categories, an SVM algorithms builds a model which assigns a sample to one category or another. This manner makes SVM a binary linear classifier. If samples are assumed as points in space, SVM model maps these samples on points such that samples belonging to different categories are divided by a clear gap. In addition, SVM can perform non-linear classification by using what is called kernels. There are different kernel functions such as Radial Based Function (RBF), linear and polynomial functions. Studies has shown that RBF conducts better than other two kernel functions [9]. Basically SVM maps the input vectors into a high-dimensional feature space. As mentioned above, the SVM algorithm labels samples into categories by a clear gap. Actually it finds a separating hyperplane in the feature space to maximize the gap between different classes.

5) Neural Networks: Neural networks are a set of algorithms modeled after the human brain and are designed to recognized patterns. Patterns recognized by neural networks are numerically contained in vectors. Artificial neural networks are made up of processing units called neurons to perform complex and parallel computations, which is connected by variable link weights and operate in parallel to learn from historical data given to them. Neural networks have many layers. First and the last layers are called input and output layer, respectively. Layers between them are called hidden layers. A simple neural network only has three layers, by adding more layers, nodes more complex models are trained. Overall, neural networks can be categorized into two classes of unsupervised and supervised; each includes different types of these networks. Some of the supervised neural networks are discussed in the following part.

A *Deep neural network:* Neural networks with multiple hidden layers are considered as deep neural networks. For complex and high-dimensional data neural networks with more layers and more nodes are required; Therefore, hardware with higher computational ability is needed. The recent improvement in graphics processing units (GPUs) and tensor processing units (TPUs) made working with these complex neural networks easier.

B *Convolutional neural network:* CNNs are a type of deep neural networks, which were inspired by the human brain and processes in that. They are the feedforward neural networks and consist of multiple hidden layers Since they are a kind of deep neural networks. These hidden layers are the convolutional layers, The activation function is commonly Relu layer and it is followed by additional convolutions such as pooling layers, fully connected layers and normalization layers. These layers reduce the training difficulties of convolutional NNs greatly.

C *Recurrent neural network:* Unlike feedforward neural networks recurrent neural network or RNN can use its the internal state as a memory to process sequence of data. Furthermore, the parameters in RNNs do not change and it shares the same parameters during all time steps. This means that at each time step, the recurrent NN performs the same task, just with different inputs. So the total number of parameters reduces greatly. Long Short-Term Memory (LSTM) is an artificial RNN architecture which has a good ability to capture long-term dependencies. Each unit is composed of a cell, an input gate, an output gate and a forget gate to compute the hidden state [10], [11].

6) Decision Tree: A decision tree is one of the classification methods. It uses a model that is like a tree and is made up of decisions and the possible consequences of making those decisions. In the tree, each node represents a feature of data and each leaf node is a class label, and all branches indicate the sequences of features that lead to classifications. Features of an unlabeled data sample will be compared to the features of nodes of the decision tree.

7) Random Forest: random Forest consists of many decision trees. It is used for classification and regression tasks. This method randomly chooses a subset of feature space to construct each decision tree. An unlabeled sample will be given to each of the decision trees and that tree outputs a label which can be considered a vote of that tree, finally the sample will be classified into the class that has the most votes between the outputs of the trees.

B. Unsupervised Learning

Unlike Supervised learning, Unsupervised learning algorithms use unlabeled data to create a model. The main task of these kind of algorithms is to find the patterns and knowledge hidden in samples of unlabeled data-set. Using similarities between different samples, these algorithms cluster them into different groups. Therefore, One of the main tasks in unsupervised learning is clustering data intro different groups and clustering analysis. Some of the most used unsupervised learning algorithms are discussed here, such as K-Means and self-organizing maps.

1) K-means: One of the most popular unsupervised algorithms is K-means. Basically, It is used for clustering the data-set. The algorithms steps are as followed:

A Randomly choose k nodes and call them centroids.

B By using a distance function label other nodes with the closest centroid node.

C choose new centroids based on the current state of the classes of the nodes.

D In case of reaching to the convergence condition stop, Otherwise go to *B*.

For more information on how these algorithms works please refer to [12], [13].

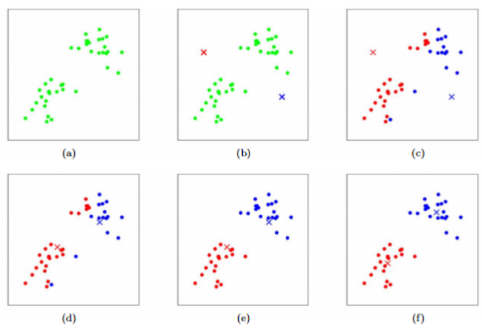

Fig. 2. K-means algorithm. The dots are Training examples, and the crosses are the cluster centroids.
(a) Original dataset.
(b) Random initial cluster centroids.
(c-f) shows the two iterations of running the algorithm. Each point has been assigned to the closest centroid. (training examples assigned to the corresponding centroid has been colored the same as the centroid); Afterward each centroid has been moved to the mean of the points which are assigned to.
Source: Adapted from [24]

2) Self-Organizing Maps (SOM): Another popular unsupervised learning algorithms is SOM also called SelfOrganizing Feature Map (SOFM). Commonly, two tasks of clustering and dimension reduction are done by this algorithm. It is consists of two layers: an input layer and a map layer. In the case of using SOM for a clustering task, the neurons of the map layer are the same size as the number of clusters and each neurons has a vector of weights. SOM algorithm works as followed for a clustering task:

A First start by initializing the weight vectors of the neurons.

B Second choose a sample from the training data-set.

C Third by using a similarity function and calculating the distance between the sample and weight vectors of all neurons, choose the most similar neuron to the sample data and call it the Best Matching Unit (BMU). There is only one BMU each time due to competitive property of this algorithm.

D Calculate the neighborhood of the BMU.

E The neighborhood of the BMU and the BMU itself are adjusted towards the sample input data.

F In case of reaching to the convergence condition stop, Otherwise go to *B*.

C. Semi-Supervised Learning

Semi-supervised algorithms use both labeled and unlabeled data. One advantages of this kind of learning, which should be noted here are that First, using unlabeled data efficiently during the learning process improves the trained model. Second, providing labeled data in real-world is difficult and expensive while acquiring unlabeled data is relatively easier. One of the semi-supervised techniques is called Pseudo Labeling [14], [15]. In this simple technique, first labeled data is used to train a model. Afterward, this model is used to predict the labels of the unlabeled data. Finally, it combines both labeled data and newly labeled data by the model to train a new model. There are so many other semi-supervised methods such as graph-based methods, co-training, Expectation Maximization (EM).

D. Reinforcement Learning

1) Reinforcement Learning (RL): Another popular ML technique is RL. RL involves a learning entity called agent which interact with the environment to choose the best action in order to maximize its long-term reward.The long-terms reward relates to both immediate reward and future reward. Applying RL in software defined networks, the agent is the controller of the network and the environment is the network itself. Therefore, The controller monitors the network status carefully then make decision to control data forwarding based on that. The function which is used to calculate the long-term reward of an action based on the current input state is called value function. The most common used value function is Q-function. It uses Q-learning to learn a table consist of pairs of action-status and their corresponding rewards.

2) Deep Reinforcement Learning: RL works well without knowing the exact mathematical model of the environment. However, It has some disadvantages such as its low rate to converge to optimal behavior. Also it is not able to work with high dimensional data, high dimensional states, high dimensional actions. These problems can be solves by using Deep Reinforcement Learning (DRL). Using the powerful property of deep NNs is function approximation, DRL approximates its value function. After the deep NN is trained, DRL is able to approximate the long-term reward for the pair of state-action as an input, This approximated result helps the agent to choose the best action.

In summary, It is important to note that supervised learning algorithms are applied to the classification and regression task while for clustering and decision making tasks unsupervised and reinforcement learning techniques are used.

IV. Challenges in Networks and Application of Machine Learning

In a software defined network, The controller has a global view, It makes managing and controlling the network easier, While machine learning algorithms can help SDN to automate the services and bring intelligence to the network. In conclusion, SDN controller can learn the environment of the network and automatically make optimal decisions. In this section, some of the challenges are Networks and how the ML algorithms change and help them are discussed. Challenges such as Routing optimization, traffic classification and QoS/QoE predication are mentioned here.

A. Routing Optimization

Routing is the main and fundamental task in networks. In software defined networks, The controller controls the routing of flows by changing and modifying the flow tables of the routers in the network. Inefficient decisions in order to choose an appropriate route for a specific flow can cause problems such as increasing the end-to-end transmission delay or over-loading of the links of the network. Therefore, optimizing the routing of traffic flows is a critical task.

Two types of routing optimization algorithms which is greatly applied in networks are heuristic algorithms and Shortest Path First (SPF) algorithms. SPF algorithms simply choose the shortest path between the source and the destination router. This shortest path can be based on hop-count or delay, Despite its simplicity, this algorithm is best effort hence it does not use all of the resources of the network.As mentioned earlier in this section, heuristic algorithms such as ant colony, are another approach for routing optimization but they have some disadvantages such as high computational complexity [16]. Moreover, These algorithms add to the computational burden of the controller. Remembering the work flow of SDN architecture, when a router does not have an output for a new data flow, the controller is responsible for the calculation of the appropriate policy. Hence, the computational burden of the controller increases. In order to solve these problems, machine learning techniques are used. On one hand these algorithms outputs the near-optimal solution on the other hand they do not require the exact mathematical model of the network. Choose a route for a specific traffic can be costarred as a decision making task. Therefore, reinforcement learning techniques are great approaches for this problem. However, Some supervised algorithms can be useful too. Following of this section briefly ML algorithms used in routing optimization is mentioned.

1) Supervised-based Routing Optimization: When applying Supervised algorithms to SDN for optimize routing, the network and traffic status are the input of the training dataset and the corresponding heuristic algorithm results are the outputs.

Ref. [16] represents a framework called NeuRoute. This framework is actually for dynamic routing and uses LSTM to approximate and estimate future network traffic. Afterward the estimated network traffic and the status of the network is used as an input and the corresponding heuristic algorithm result is used as an output and finally a deep neural network is trained based on this input and output. This trained the neural network can be applied in order to obtain the real-time heuristic-like results.
The advantage of supervised learning algorithms is that they are useful to obtain heuristic-like solutions. However, gathering of labeled training datasets has a high computational complexity.

2) Reinforcement Learning-based Routing optimization: RL algorithms basically is used for decision-making problems. In RL algorithm applied in the network, the agent is the controller, the network is the environment, the state space is network and traffic states and the action is the routing solution. The reward can be defined differently and it is based on optimization metrics such as network delay.

In [17], the authors study the routing optimization in multi-layer hierarchical SDN. The method proposed in this paper is a QoS-aware Adaptive Routing (QAR). By applying RL algorithms, it enables time-efficient adaptive packet forwarding. Based on the traffic types and users' applications the path with the maximum QoS-aware reward is selected.

Ref.[18] utilizes a DRL model for optimization routing. To minimize the network delay, the DRL model selects the optimal routing path for all source-destination pairs which is given to the traffic matrix.

Compared to supervised learning algorithms, Rl algorithms have some advantages such as not needing a labeled data-set and its flexibility to optimize routing based on different metrics (e.g., delay, throughput and energy efficiency) and change the reward function.

3) Traffic classification in Routing Optimization: Traffic classification is considered as an important issue in routing optimization. By analyzing the historical traffic information trend of the traffic volume can be predicted [19]. Based on the prediction result, the SDN controller can make traffic decisions.

Ref. [21] a LSTM-based, framework NeuTm, is proposed to predict the traffic matrix of the network. The data used for training the network is GEANT, which is the real-world traffic data [22].

Ref. [20] proposes a load balance strategy to optimize path load. In order to predict the load of the each path four features are selected for training the neural network, transmission hop, packet loss rate, bandwidth utilization ratio and transmission latency. The SDN controller uses these four features ti predict the load of each path and the lead loaded path will be selected for the new traffic flow.

4) Others: In [23], an efficient routing protocol is proposed that considers risks. This protocol first uses K-means offline to cluster traffic data into classes of risk ratios. Then, Ant Colony Optimization is used to select minimized compliance risks for a transmission session online.

B. Traffic Classification

Traffic classification is used to provide differentiate services and provide a more efficient network. There are a few techniques to do classify the traffic, such as port-based approach, Deep packet inspection and ML based techniques [26].

1) Port-based Approach: Port-based approach uses port number to classify the connections. FTP, TCP, UDP and other commonly used protocols have specific ports. Modern applications nowadays use multiple and dynamic ports which makes this method not really useful in modern era.

2) Deep Packet Inspection: DPI identifies each flow by mapping it to a regular expression.As the network grows, flows grow exponentially. Although this mapping is precise, it has a high cost and it can't grow with the same speed as the network grows.

3) Machine Learning Approach: Machine Learning Approach has a few way to classify the flows. Elephant flow traffic classification is one of them. In this method we have two type of flows, Elephant flows and Mice flows. Elephant flows are making up most of the bandwidth of the network unlike Mice flows which are small packets but they need much less delay. Most of the flows in a network are Mice flows [27]. Ref. [28] Machine Learning Algorithms are used to classify the flows at the edge of the network in hybrid data centers.

C. Quality of Service and Quality of Experience Prediction

1) Quality of Service: QoS prediction can help us predict the jitter, throughput, loss rate or any network metric. These QoS perimeters are tightly coupled with Key Performance Indicators (KPIs). Since QoS data is continuous we can use regression to solve this problem. As we mentioned earlier we use supervised learning for regression problems.

For example, in Video-on-Demand problem two important factors are response time and frame rate which can be used to predict the QoS metrics [29].

Ref. [30] We use a supervised learning algorithm after making a desicion tree based on realation of QoS parameters and KPIs.

2) Quality of Experience: Due to the widespread use of the social media, it's important to keep the customers happy, because happy customers bring more customers from sharing and giving positive feedback. Unlike QoS, data in QoE is not continuous, Customers are either happy or not. Mean Opinion Score (MOS) [31] is one of the ways that we quantify customer satisfaction into bad, poor, fair, good and excellent. Goal is to find a relation between QoE scores and QoS parameters, for example how much packet loss can affect user satisfaction.

Ref. [31] four method has been used to predict the MOS on video QoS parameters including neural networks and k-NN.

V. CONCLUSIONS

Effective ML algorithms has been applied to SDN and it can significantly improve the quality of the network. We have provided a brief overview on software defined networks, then we moved some Machine learning techniques and the end we mentioned how these techniques can be applied to Routing, QoS/QoE prediction and Traffic classification.

In summary, there are rooms for improvement in using ML in SDN, due to the nature of SDN we have all we need to use machine learning, we have data on large-scale, powerful network computers and skilled engineers. There are many challenges in our way but the we are sure that we can pave the road for having an enhanced network.

REFERENCES

[1] M. Mohammed, M. B. Khan, and E. B. M. Bashier, Machine Learing: Algorithms and Applications. CRC Press, 2016.

[2] S. Marsland, Machine Learing: An Algorithmic Perpective. CRC Press, 2015.

[3] M. Kubat, An Introduction to Machine Learning. Springer, 2016.

[4] E. Alpaydin, Introduction to Machine Learning. MIT Press, 2014.

[5] J. Friedman, T. Hastie, and R. Tibshirani, The elements of statistical learning, vol. 1. Springer series in statistics Springer, Berlin, 2001.

[6] T. Cover and P. Hart, "Nearest neighbor pattern classification," IEEE transactions on information theory, vol. 13, no. 1, pp. 21–27, 1967.

[7] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," Proceedings of the IEEE, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[8] P. Holgado, V. A. VILLAGRA, and L. Vazquez, "Real-time multistep attack prediction based on hidden markov models," IEEE Trans. Dependable and Secure Computing, vol. PP, no. 99, pp. 1–1, 2017.

[9] B. Yekkehkhany, A. Safari, S. Homayouni, and M. Hasanlou, "A comparision study of different functions for SVM-based classification of multi-temporal polarimetry SAR data," The International Archives of Photogrammetry, Remote Sending and Spatial Information Sciences, vol. 40, no. 2, p. 281, 2014.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] X. Li and X. Wu, "Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition," in Proc. IEEE ICASSP'15, Brisbane, QLD, Australia, April 2015, pp. 4520–4524.

[12] J. Friedman, T. Hastie, and R. Tibshirani, The elemets of Static learning. Springer Series in Statistics New York, 2001, vol. 1.

[13] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient K-means clustering algorithm: Analysis and implementation," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 881–892, Jul. 2002.

[14] D.-H. Lee, "Pseudo-lable: The simple and efficient semi-supervised learning method for deep neural networks," in Workshop on Challenges in Representation Learning, ICML, vol. 3, 2013, p. 2.

[15] H. Wu and S. Prasad, "Semi-supervised deep learning using pseudo labels for hyperspectral image classification," IEEE Trans. Image Processing, vol. 27, no. 3, pp. 1259–1270, March 2018.

[16] L. Yanjun, L. Xiaobo, and Y. Osamu, "Traffic engineering framework with machine learning based meta-layer in software-defined networks," in Proc. IEEE ICNIDC'14, Beijing, China, Sept. 2014, pp. 121–125.

[17] S. C. Lin, I. F. Akyildiz, P. Wang, and M. Luo, "QoS-aware adaptive routing in multi-layer hierarchical software defined networks: A reinforcemnet learning approach," in Proc. IEEE SCC'16, San Francisco, CA, USA, June. 2016, pp. 25–33.

[18] G. Stampa, M. Arias, D. Sanchez-Charles, V. Muntes-Mulero, and A. Cabellos, "AA deep-reinforcemnet learning approach for spftware defined-networking routing optimization," arXiv preprint arXiv:1709,0v080, 2017.

[19] A'.L'opez-Ravent'os,F. Wilhelmi,S. Barrachina-Mu~noz,and B.Bellalta, "Machine Learining and software defined networks for high0density WLANs," arXiv preprint arXiv:1804,05534, 2018.

[20] C. Chen-Xiao and X. Ya-Bin, "Research on load balance method in SDN," International Journal of Grid and Distributed Computing, vol. 9, no. 1, pp. 25–36, 2016.

[21] A. Azzouni and G. Pujolle, "NeuTM: A neural network-based framework for traffic matrix prediction in SDN," arXiv preprint arXiv:1710,06799, 2017.

[22] "GEANT Network," May. 2018. [Online]. Available: https://www.geant.org/Projects/GEANT Project GN4

[23] 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'17, Honolulu, CA, USA, June. 2017, pp. 761–765.

[24] Chris piech, "k-means based on Andrew N.g handout", Stanford, 2013. [online] Available: https://stanford.edu/ cpiech/cs221/handouts/kmeans.html

[25] Amirsina Torfi, "k-Nearest Neighbors", 2019. [online] Available: https://machine-learning-course.readthedocs.io/en/latest/content/supervised/knn.html

[26] P. Amaral, J. Dinis, P. Pinto, L. Bernardo, J. Tavares, and H. S. Mamede, "Machine learning in software dened networks: Data collection and trafc classication," in Proc. IEEE ICNP'16, Singapore, Singapore, Nov. 2016, pp. 1–5

[27] T. Benson, A. Akella, and D. A. Maltz, "Network trafc characteristics of data centers in the wild," in Proc. ACM IMC'10, Melbourne, Australia, 2010, pp. 267–280.
[28] M. Glick and H. Rastegarfar, "Scheduling and control in hybrid data centers,"in Proc. IEEE PHOSST'17, San Juan, Puerto Rico,July.2017, pp. 115–116.
[29] R. Pasquini and R. Stadler, "Learning end-to-end application QoS from OpenFlow switch statistics," in Proc. IEEE NETSOFT'17, Bologna, Italy, 2017, pp. 1–9.
[30] S. Jain, M. Khandelwal, A. Katkar, and J. Nygate, "Applying big data technologies to manage QoS in an SDN," in Proc. IEEE CNSM'16, Montreal, QC, Canada, Oct. 2016, pp. 302–306.
[31] T. Abar, A. B. Letaifa, and S. E. Asmi, "Machine learning based QoE prediction in SDN networks," in Proc. IEEE IWCMC'17, Valencia, Spain, 2017, pp. 1395–1400.