INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 6.078
(Volume 6, Issue 2)
Available online at: www.ijariit.com

Artificial Intelligence in Software Defined Networking

Arhaan Aggarwal
arhaanaggarwal2021@sfhs.in

ABSTRACT

This paper focuses on the challenges and opportunities of Software Defined Networking (SDN), as well as how to select the best possible SDN controller, which will help reduce the complexity of a network, price of implementation, and the maintenance of a network in any big organization. This is followed by a basic introduction to Artificial Intelligence and some of its important domains and their use in SDN.

Keywords— *Software Defined Networking, Artificial Intelligence, OpenFlow, Machine Learning*

1. INTRODUCTION

In the past decade or so, the hot topic in network administration has been Software Defined Networking (SDN) and SDN Controllers. SDN has been able to generate solutions to the problems of traditional networks. These problems include a separate control plane for each device in a network, including switches and routers, due to which both decision making and processing takes place on each device. In traditional networks, there is no centralized visibility of network devices, due to which a separate Network Monitoring System (NMS) needs to be installed for full visibility.

Customary systems administration structures (conventional approach) have many critical constraints due to which they are not able to fully support today's IT prerequisites. Today's systems need to scale to suit expanded workloads with better spryness, while also maintaining the costs at a minimum. In the past few years, SDN has been able to attract many industries due to its many advantages, which are mainly due to network virtualization. The main and the most important characteristic of SDN is that it combines the control and data plane, due to which a single software control program can and does manage and control the various data plane elements. SDN also applies server virtualization to increment asset effectiveness, improve the manual IT procedures, and tune applications and systems. SDN utilizes virtualization to incredibly growing, proficiently organize and accordingly answer the need to expand the network limit without burning up the available resources, and disentangle the administration of the united assets. SDN is thus able to provide solutions to the problems of traditional networking.

A major problem in traditional networking is the fact that there is a separate control plane for each device in a network, including switches and routers, due to which both decision making and processing takes place on each device. Also, there is no centralized visibility of network devices due to which a separate Network Monitoring System needs to be installed for full visibility. To solve these problems, SDN has come up with a centralized approach with a separate controller and data plane, network visibility, programmable interfaces, capable and speedy networks, OS compatibility as an open interface, and source code access. In SDN, both the data plane and the control plane exist inside the controller but are separated for their working. While the data plane is used for carrying traffic, the control plane is used for communication between the controller and network applications, decision making, and for forwarding the decisions and the actions.

Recently, many SDN Controllers like Beacon Open Flow controller, NOX, POX, Nettle, OpenDayLight, FloodLight, and Ryu have come into existence. Using these platforms, scientists and researchers have been able to create applications such as Load Balancing, Network Virtualisation, Energy Efficient Networking, Dynamic Access Control in Enterprise networks, Virtual Machine Mobility, etc.

2. SOFTWARE DEFINED NETWORKING (SDN)

Software-Defined Networking is one of the most discussed topics these days. It is also one of the most researched topics these days. It is a networking model that solves the problems of the old networking models. SDN's design is based on a separation of network intelligence and packet switching devices, and on merging the network intelligence in a centralized controller. This controller acts as the main brain or the main controller which is responsible for the decision making that takes place at the switches using protocols named OpenFlow protocols. It is being considered as a favorable technology due to the isolation of the control plane and the data plane, and the logical placement of the centralized control in the SDN controller. The data plane and control plane is decoupled to serve various purposes.

The implementation of SDN in existing networks is still a complex issue. Due to this, the implementation of SDN is still in a delicate stage despite its various achievements in the networking field. Most researchers agree that the delay caused in the implementation of SDN is mainly due to the complexities caused by it. Ongoing research and the deployment of industrial equipment could resolve some of its complications like performance, scalability, security, and interoperability issues. No matter how many challenges are faced by SDN during implementation, the advantages of scalability and reliability of SDN justifies the approach.

The SDN architecture has three main layers, namely the application layer, the control layer, and the underlying infrastructure layer. Unlike traditional architecture, where each device has a separate control plane, in SDN the control plane is centralized in an isolated process called the controller which runs in the control layer. Due to this, the applications running in the application layer appear to be running on a solo logical network switch.

3. OpenFlow

OpenFlow system was initially developed at Stanford and is now undergoing dynamic gauges of improvement by the Open Networking Foundation (ONF). The OpenFlow protocol is based on SDN layered architecture; it lies between the control plane and the forwarding plane and acts as a communication protocol. It is usually implemented between SDN Controllers and OpenFlow enabled switches, and uses flow tables for matching the traffic (data flow) through the network. There are two types of OpenFlow switches OpenFlow Only switches and OpenFlow Hybrid switches. The OpenFlow Only switches use only the OpenFlow operations, while the OpenFlow Hybrid switches utilize both OpenFlow operations and the normal Ethernet operations.

OpenFlow was originally made for analysts to rapidly and adaptably try different things with new SDN views a d device in a wide variety of environments. It helps cut physical systems through an intermediate layer. The controller works as a straightforward transition between OpenFlow switches and other standard OpenFlow controllers, and manages the data transfer capacity, CPU use and fill tables.

OpenFlow characterizes a variation through which a constantly incorporated controller can control OpenFlow switches. Each OpenFlow enabled switch keeps up one or more accumulation tables that are used to perform module queries.

Traditional systems are not able to offer a dynamic approach to express the full range of a client's requirements like accessibility. A few systems try to implement the clients' requirements separately through transfer analysis, which can increase costs and at times generate prompts of misclassification. SDN can offer a client the capacity to completely indicate its need for a trusted relation that can be adapted. Control choices are made on a method with a global perspective of the system state, instead of distribution in separate modules at every system call. With SDN, the control plane functions as a private, sensibly unified framework regarding both planning and determining asset clashes and also in a selection of low-level device components.

Networks can also be controlled by programming interfaces called Application Programming Interfaces (APIs). Most probable, hardware-independent technology will be achieved by using APIs. In software-defined networks, devices like switches and routers are only used for forwarding the packets(information), and all decisions are made by the controller which uses network applications (located inside the controller) to decide what action should be taken for a certain flow. In SDN, network devices are configured centrally and not individually. OpenFlow is an ascending protocol, and due to its use in SDN as an API, it can run without being dependent on the vendor for compatibility to any service/application. The overall expense of a network can be drastically reduced by using SDN with the OpenFlow Protocol. SDN OpenFlow design enables the accessibility of all network devices using APIs. These APIs can get all the information on network working services like routing, QoS, etc. In traditional networks, all the network devices have to be provisioned by accessing them individually, but here all the network devices are provisioned centrally by the controller.

There are two different types of interfaces in SDN controllers, both of which have separate functions. First, the North Bound Interface, also known as the application layer connected interface, communicates with the upper layer to obtain updates about the rules which have to be followed for communication. This is a high-level communication. Second, the South Bound Interface is usually used for downward communication with the network and for obtaining status updates about the forwarding policies being pushed to downward devices, which are generally switches.

4. SELECTION OF SDN CONTROLLERS

The selection of an SDN controller in an SDN is a key challenge faced by the network administrator. The controller provides a universal of the entire network and supports applications and services. In SDN, it is necessary to have a good device for the efficient processing of all requests made by the switch and for the overall good behavior of the network.

SDN controllers have several different properties that affect the overall efficiency of the network. Selecting controllers based on a single property is like looking for a needle in a haystack. Therefore, multiple properties of a controller need to be selected and analyzed to find out their impact on the efficiency and effectiveness in the network. For the analysis of SDN controllers, an Analytical Hierarchy Process (AHP) is used as it is used for solving and getting results from multiple properties being analyzed. Two of the best SDN controllers have been analyzed in brief right after this.

SDN has developed from many types of research, particularly SANE and Ethane. NOX (an SDN controller) was created at Nicira Networks. It used to be the main OpenFlow Controller. Nicira handed over NOX to the inspection group in 2008. NOX can provide asynchronous and incredible fast Input and Output(I/0). It provides a C++ OpenFlow 1.0 API and targets the latest Linux technologies. It includes many trial components for the discovery of new topologies, switch learning, and network-wide switches.

FloodLight controller is an SDN controller that characterizes the open interchanges convention in an SDN domain, which permits the SDN controller (the brain of the system) to address the sending plane (switches and routers).

4.1 Modernization using SDN based Networks
Network Managing Devices and Administrations can easily be integrated into SDN system applications, and numerous systems applications have been proposed by the SDN research group. They are SDN Network Management, Load Balancing, SDN Security, Virtualization, etc. Applications that are added to an SDN based system only require a reasonable number of control components to be updated.

4.2 Load Balancing for Application Servers
Different SDN based applications have been recommended for big commercial systems. A typical application is a load balancing for application servers. An OpenFlow switch disseminates the activity of various servers. It is done by a centralized device (load balancer) that receives the packets that are specifically sent towards support counterfeits.

4.3 Security and Network Access Control:
SDN can control the flow of virtualized resources in a network. Service Function Changing (SFC) (firewalls, load balancers, etc.) is responsible for system security and access control. SDN can accomplish useful work for system security by managing the activity to the resources dependent upon system strategies. For the requirement of system-wide management, Open Management design can be used to 4.4 checks and uphold control strategies and invariants continuously. Such invariants can incorporate checks for sending loops, flawed instructions, and managing flaws, depending on the self-arranging maps to group activity design.

4.4 Challenges to SDN
The major problems of SDN are:
- Identifying the properties of the SDN controller and their impact on the decision-making techniques of complex problems.
- Identification of the best SDN controller by using the AHP technique.
- A backup SDN controller should always be on active standby, and if one controller is down, then all the flow tables should go to the backup controller for smooth traffic operations.
- Implementing dynamic multipath load balancing i.e. sharing of load in case of congestion.
- The huge amount of data that has to be handles makes it necessary for the network to have a high CPU and memory at the controller side so that all the requests can be handled without delay.
- OpenFlow hardware switches send flow for decision making to the SDN controller, due to which thousands of flow requests can come, which causes latency.
- Controller placement is very important as delays can be caused if the controller is not properly placed.
- Interoperability is a major issue in SDN. There needs to be an interface for the communication between an SDN and a non SDN control plane.
- From the security point of view, there should be some sort of Access Control List (ACL).

4.5 Artificial Intelligence
Artificial Intelligence (AI) is a rapidly growing field that includes a wide range of subfields, including knowledge representation, reasoning, planning, decision making, optimization, machine learning (ML) and metaheuristic algorithms. Turing Test provides a fulfilling definition of intelligence, where a computer has to answer some questions written by a human interrogator. Accordingly, the computer passes the test if the interrogator cannot tell whether the answer was written by a human or a machine. To pass the Turing Test, the computer needs to include advanced capabilities such as natural language processing, knowledge representation, automated reasoning, machine learning, and computer vision. AI research started later in the mid-1950s, where a summer workshop organized by Martin Minsky and Claude Shannon at Dartmouth College resulted in the birth of the field of AI. The first contribution, however, was made in 1943 by McCulloch and Pitts, when they proposed the first model for artificial neural networks, in which each neuron has a binary output (1,+1) with a sign activation function. The adoption of AI approaches increased, thanks to key contributions that led to the emergence of new subfields such as expert systems, fuzzy logic, and evolutionary computation. Further efforts have fueled the research in AI field by refining the existing methods and proposing brand new hybrid intelligent approaches. Machine learning, metaheuristic algorithms, and fuzzy inference systems are widely used in the SDN paradigm therefore we provide an introduction regarding these approaches.

4.6 Machine Learning
An intelligent machine learns from experience (i.e., learns from the data available in its environment) and uses it to improve the overall performance. The technique under which a machine can learn from its experience is known as Machine Learning.

4.7 Supervised Learning
Supervised learning methods are provided with pre defined knowledge. For instance, a training dataset that consists of input-output pairs, where the system learns a function that maps a given input to an appropriate output. This approach requires having a dataset that represents the system under consideration and can be used to estimate the performance of the selected method

4.8 Artificial Neural Networks
Neural networks are mainly inspired by biological learning systems such as biological neurons in the human brain. Artificial neural networks have many advantages. First, they can adjust themselves to the data without explicitly specifying a functional or

distribution for representing the underlying model. Second, neural networks form a universal functional approximator, which can approximate any function. Third, neural networks are non-linear models, which gives them the flexibility to represent and model complex relationships. The feed-forward multilayer networks or multilayer perceptrons (MLPs) are the most commonly used neural network classifiers. MLPs are mainly trained with supervised training algorithms. Neural networks are subject to over fitting when we use too many parameters in our model. We also need to find the best network structure to achieve good performance.

4.9 Supervised Deep Learning

Deep learning provides a general-purpose multi-level representation learning approach. In representation learning a machine can learn to automatically discover the representations required for classification or detection tasks based merely on raw data, whereas conventional machine learning techniques cannot deal natural data in their raw form. Multiple levels of representation allows transforming the representation from the low level into a higher abstract one. Enough number of these transformations allows learning more complex functions. Deep learning techniques have achieved better performance compared to the traditional algorithms used for many machine learning tasks such as speech recognition, intrusion detection, objection detection, and natural language understanding. Deep learning models are categorized into three groups, namely: 1) generative, 2) discriminative, and 3) hybrid models. Discriminative models mainly use supervised learning approaches, whereas generative models employ unsupervised learning approaches. Hybrid models, on the other hand, make use of both discriminative and generative models. In this paper, the term deep neural networks (DNN) refers to deep feedforward multilayer networks or multilayer perceptrons (MLPs). Other important two supervised models used in deep learning are recurrent neural networks and convolutional neural network

4.10 Unsupervised Learning

Unsupervised learning methods are provided without a predefined knowledge (i.e., having an unlabeled data). Therefore, the system mainly focuses on finding specific patterns in the input. An example of the unsupervised learning approach is clustering, which is used for detecting useful clusters in the input data based on similar properties defined by a proper distance metric such as Euclidian, Jaccard, and cosine distance metrics.

4.11 Semi-Supervised Learning

In semi-supervised learning, the system learns from both labeled and unlabeled data, where the lack of labels, as well as the labeled part, may contain a random noise forms a situation between supervised and unsupervised learning. It is more realistic as in many real-world applications, it is often difficult to collect many labeled data because the data are manually labeled by the experts, whereas it was easier to collect a lot of unlabeled data. As it includes some small labeled data, the performance of semi supervise learning approaches is superior to unsupervised learning

Table 1: Comparison of different ML Approaches

ML Approach	Advantages	Disadvantages
Supervised Learning	Learns from labeled data.	Requires a dataset that represents the system. The data is manually labeled by human experts, which is not appropriate for many real-world applications.
Unsupervised Learning	Finds hidden patterns without relying on labeled data. Performs better for unseen data in comparison with the supervised approach.	It may not provide a useful insight into the hidden patterns and what they mean
Semi-Supervised Learning	Learns from both labeled and unlabelled data. Certain assumptions about the underlying data distribution must be met.	It may lead to worse performance when we choose wrong assumptions.

4.12 AI In SDN

AI and ML approaches have been widely used for solving various problems such as routing, traffic classification, flow clustering intrusion detection, load balancing, fault detection, quality of service (QoS) and quality of experience (QoE) optimiSation, admission control and resource allocation. However, in the era of SDN the role of AI has significantly increased due to the huge efforts made by industry and research community. Recent studies have shown by a strong tendency of research community towards adoption of AI approaches in SDNs. It is worth mentioning that ML, meta heuristics and fuzzy systems were the most common approaches for solving various networking related problems.

4.13 Neural Networks in SDN

Neural networks (NN) approach was used mainly for intrusion detection and prevention, solving controller placement problem, load balancing, performance prediction, service level agreements (SLA) enforcement, routing and optimal virtual machine (VM) placement.

Chen and Yu proposed, CIPA, a collaborative intrusion prevention architecture, which represents a distributed intrusion prevention system based on NN approach. They used the following features: number of all packets monitored, proportion of ICMP packets to all packets, the proportion of short packets, the proportion of long packets, the proportion of UDP packets to all packets and the base 10 logarithm of the proportion of packets with syn flag set to packets with ack flag set. The experimental results showed that CIPA outperforms in detecting DDoS flooding attacks. CIPA also achieved good results in detecting Witty, Slammer and Conficker worm outbreak. The system achieved low computational and communication overhead due to its parallel and simple computational capabilities.

He et al. proposed a multi label classification approach to predict global network allocations (i.e. weighted controller placement problem). Compared to decision tree and logistic regression, neural network approach showed better results and saved up to two thirds of the algorithm runtime. Alvizu et al. used a neural network approach for off line prediction of traffic demands in a mobile network operator, which resulted in reducing the optimality gap below 0.2% (virtual wavelength path hourly NN) and 0.45% (wavelength path hourly NN). In addition, NN approach was used for off line prediction of the next configuration time point. Abubakar et al. proposed an intrusion detection system for SDN based on NN approach, which achieved a high accuracy reached 97.3% using NSL KDD dataset.

Chen Xiao and Ya Bin proposed an NN approach for load balancing. Compared to and static Round Robin strategy, the experimental study showed that this method can achieve better performance and resulted in 19.3% decreasing of network latency. Sabbeh et al. proposed an NN approach based on Levenberg Marquardt (LM) algorithm to predict the performance of SDN according to round trip time (RTT) and throughput which are the two parameters used for training. The experimental results showed that one hidden layer achieves low mean squared error (MSE).

Bendriss et al. proposed a new approach to enforce service level agreements (SLA) in SDN and virtualized network functions (NFV). Their research focused on prediction of service level objectives (SLOs) breaches for streaming services running on NFV and SDN. The experimental results showed that long short term memory (LSTM) is more robust than feed forward neural networks.

Mestres et al. presented a new paradigm that employs AI in SDN, termed as knowledge defined networking (KDN). The paradigm included a proof of concept concerning routing in an overlay network based on NN approach where the mean squared error (MSE) reached 1%. In addition to solving the problem of optimal virtual machine (VM) placement in NFV paradigm. Mihai Gabriel and Victor Valeriu proposed a method for mitigating DDoS attacks in SDNs by risk assessing based on neural networks and biological danger theory. The risk of a DDoS attack is calculated on every host and then the reported to a VM that is responsible for monitoring the network traffic. When the risk of the observed traffic exceeds a predefined value, instructions will be sent to the controller to install the appropriate controls that allow the SDN to enter in a proactive mode to reduce the burden on the controller caused by sending these flows to the controller for analysis.

4.14 Supervised Deep Learning In SDN

Deep learning techniques have shown promising results in SDN compared to traditional ML approaches. Tang et al. proposed an intrusion detection system based on a simple deep neural network, where deep learning approach achieved the best results compared to other supervised ML algorithms such as naive bayes, SVM and DT, with an accuracy of 75.75%. In, the same authors improved their results by using GRU RNN instead of the simple deep learning model. Their new proposed model outperformed their previous simple DNN based model, SVM and NB Tree with an accuracy of 89%. Lazaris and Prasanna proposed, DeepFlow, an intelligent traffic measurement framework for SDNs that uses the available TCAM memory to install measurement rules for important flows, and employs LSTM RNN to predict the size of rest of the flows when flow counters cannot be placed at a switch due to its limited resources, on the basis of historical data from previous measurement periods. The experimental results showed an average mean absolute percentage error (MAPE) of 12% for approximating real flow sizes on CAIDA traces and MAPE of 3.9% on simulating the network topology of Google's B4.

Azzouni et al. employed a (LSTM RNN) approach for predicting traffic matrix. GÉANT traffic matrices and related network states were the input of the LSTM network. The authors implemented a GÉANT network topology and generated 10000 samples using Mininet. Experimental studies showed that this approach outperformed an efficient dynamic routing heuristic by finding the near optimal path in shorter time. In the same context, Azzouni and Pujolle used (LSTM RNN) for the same purpose. However, in this work they validate their model based on real world data from GÉANT backbone net works. The experimental results showed that RRN LSTM can outperform linear forecasting models (ARMA, ARAR, HW) and feed forward neural network (FFNN).

Huang et al. studied adversarial attacks on SDN based deep learning port scan detection system. They applied three different adversarial attack mechanisms namely: Fast Gradient Sign Method (FGSM), Jacobian based Saliency Map Attack (JSMA), JSMA reverse (JSMA RE) to three different deep learning algorithms: MLP (which is previously refereed as DNN in this paper), CNN, LSTM. The deep learning models detected the port scan attacks based on Packet In messages and STATs reports. They generated samples with three different adversarial test sets and one normal test set. JSMA showed a significant effect on the deep learning models ranges from 14 to 42%. JSMA RE did not reduce the accuracy of CNN and LSTM, however, it reduced the accuracy of MLP to 35%. FGSM caused a significant reduction in the accuracy of LSTM (more than 50%).

4.15 Unsupervised Learning in SDN

K means clustering, self-organizing map hidden Markov model, Restricted Boltzmann machines, and unsupervised deep learning approaches were the most used unsupervised learning techniques in SDN paradigm.

4.16 Semi Supervised Learning In SDN

Semi supervised learning was also used in SDN, but much less common compared with other learning approaches. The research was focused on traffic classification, routing and intrusion detection. Wang et al. proposed a new framework for QoS aware traffic classification based on semi supervised learning. This approach can classify the network traffic according to the QoS requirements. The system allows achieving deep packet inspection (DPI) and semi supervised learning based on Laplacian SVM. The Laplacian SVM approach outperformed a previous semi supervised approach based on k means classifier.

Chen and Zheng introduced an efficient routing predesign solution based on semi supervised approach. The study suggested using an appropriate clustering algorithm such as Gaussian mixture model and k means clustering for feature extraction. Thereafter, a supervised classification approach such as extreme learning machine (ELM) can be used for flow demand forecasting. The authors also suggested using an adaptive multipath routing approach based on analytic hierarchy process (AHP) for handling to elephant flows according different constraint factor weights.

Li et al. proposed a new method for fine grained traffic classification based on semi supervised approach called nearest application-based cluster classifier (NACC). Unlike traditional methods, which use one feature vectors, this algorithm constructs a matrix with several cluster centroids based on k means clustering to represent the application.

The algorithm uses a small number of labelled flows to build a supervised dataset. Then it collects the unlabeled flows to be merged with previously collected dataset, based on investigating the correlated flows, which is used to map an application to different clusters. The experimental results showed a good identification accuracy reaching 90%.

Wang et al. introduced an intrusion detection method based on semi supervised approach for wireless SDN based e health monitoring systems. The proposed system employed the concept of off line training and on line testing to allow running localized intrusion detection on wireless massive machine type communications (mMTC) devices. Their system adopts semi supervised learning on the basis of modified contrastive pessimistic likelihood estimation (CPLE), in which they replace the maximization calculation by a relaxation function. CPLE performs semi supervised parameter estimation for likelihood based classifiers. The proposed modified CPLE outperformed Naive Bayes, SVM, DNN, self-training (semi supervised approach) and the original CPLE based on the experiments conducted on NSL KDD dataset.

5. CONCLUSION

In this paper, a state-of-the-art overview into SDN was provided with its various features, limitations, and selection of the best possible SDN Controller. This was followed by an analysis of AI and its domains and their use and role in SDN and its applications. Overall, the main focus of this paper was on how AI can be implemented in SDN. It can be noted that AI tools have been very useful tools in SDN. However, both of these are relatively new and evolving fields, and many further advancements are being done in them which is likely to change the scope of both of these fields.

6. REFERENCES

[1] Russell, S., Norvig, P.: ‘Artificial Intelligence (A Modern Approach)’. 3rd ed. New Jersey: Prentice Hall, 1995. 1152 p.
[2] Negnevitsky, M.: ‘Artificial Intelligence A Guide to Intelligent Systems’. 2nd ed. Essex: Addison Wesley, 2005. 415 p.
[3] Zhang, G.P.: ‘Neural networks for classification: a survey’, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2000,30,(4), pp. 451 462
[4] Nguyen, T.T., Armitage, G.: ‘A survey of techniques for internet traffic classification using machine learning’. IEEE Communications Surveys & Tutorials. 2008,10,(4), pp. 56 76.
[5] LeCun, Y., Bengio, Y., Hinton, G.: ‘Deep learning’, Nature, 2016, 521,(7553), pp.436 444.
[6] Deng, L.: ‘A tutorial survey of architectures, algorithms, and applications for deep learning’, APSIPA Transactions on Signal and Information Processing, 2014, 3,(e2), pp. 1 19.
[7] Mohammadi, M., Al Fuqaha, A., Sorour, S. Guizani, M.: ‘Deep Learning for IoT Big Data and Streaming Analytics: A Survey’, IEEE Communications Surveys & Tutorials.
[8] Tang, T., Zaidi, S.A.R., McLernon, D., Mhamdi, L. Ghogho, M.: ‘Deep Recurrent Neural Network for Intrusion Detection in SDN based Networks’. In 2018 IEEE International Conference on Network Softwarization (NetSoft 2018), Montreal, Canada, Jun 2018.
[9] Fan, X., Guo, Z.: ‘A semi supervised Text Classification Method based on Incremental EM Algorithm’. WASE International Conference on Information Engineering, Beidaihe, China, August, 2010, pp. 211 214.
[10] Gui, T., Ma, C., Wang, F., Wilkins, D.E.: ‘Survey on swarm intelligence based routing protocols for wireless sensor networks: An extensive study. In: Proc. of IEEE International Conference on Industrial Technology (ICIT), Taipei, Taiwan, March 2016, pp. 1944 1949.
[11] Soysal, M., Schmidt, E.G.: ‘Machine learning algorithms for accurate flow based network traffic classification: Evaluation and comparison’, Performance Evaluation, 2010,67,(6), pp. 451 467.
[12] McGregor, A., Hall, M., Lorier, P., Brunskill, J.: ‘Flow clustering using machine learning techniques. In: Proc. of International Workshop on Passive and Active Network Measurement (PAM), Antibes Juan les Pins, France, April 2014, pp. 205 214.
[13] Xu, X., Wang, X.: ‘An adaptive network intrusion detection method based on PCA and support vector machines’. In: Proc. of Advanced Data Mining and Applications, Wuhan, China, July 2005, pp. 696 703.
[14] Kim, H.Y., Kim, J.M.: ‘A load balancing scheme based on deep learning in IoT’, Cluster Computing, 2017,20,(1), pp. 873 878
[15] Moustapha, A.I., Selmic, R.R.: ‘Wireless sensor network modeling using modified recurrent neural networks: Application to fault detection, IEEE Transactions on Instrumentation and Measurement, 2008,57,(5), pp. 981 988
[16] Mushtaq, M.S., Augustin, B., Mellouk, A.: ‘Empirical study based on machine learning approach to assess the QoS/QoE correlation. In: Proc. of 17th European Conference on Networks and Optical Communications (NOC), Vilanova i la Geltru, Spain, June 2012, pp. 1 7
[17] Testolin, A., Zanforlin, M., De Grazia, M.D.F., et al.: ‘A machine learning approach to QoE based video admission control and resource allocation in wireless systems. In: Proc. of 13th Annual Mediterranean Ad Hoc Networking Workshop (MED HOC NET), Piran, Slovenia, June 2014, pp. 31 38

[18] Chen, X.F., Yu, S.Z.: 'CIPA: A collaborative intrusion prevention architecture for programmable network and SDN', Computers and Security, 2016,58, pp. 1 19
[19] Gamer, T.: 'Collaborative anomaly based detection of large scale internet attacks', Computer Networks, 2012,56,(1), pp. 169 185.
[20] He, M., Kalmbach, P., Blenk, A., Kellerer, W., Schmid, S.: 'Algorithm Data Driven Optimization of Adaptive Communication Networks'. In Proc. of IEEE 25th International Conference on Network Protocols (ICNP), Toronto, Canada, Oct. 2017, pp.1 6.
[21] He, M., Kalmbach, P., Blenk, A., Kellerer, W., Schmid, S.: 'Algorithm Data Driven Optimization of Adaptive Communication Networks'. In Proc. of IEEE 25th International Conference on Network Protocols (ICNP), Toronto, Canada, Oct. 2017, pp.1 6
[22] Abubakar, A., Pranggono, B.: 'Machine learning based intrusion detection system for software defined networks'. In Proc. of Seventh International Conference on Emerging Security Technologies (EST), Canterbury, UK, Sept. 2017, pp. 138 143
[23] Chen Xiao, C., Ya Bin, X.: 'Research on load balance method in SDN', International Journal of Grid and Distributed Computing, 2016,9,(1), pp. 25 36
[24] Li, Y., Pan, D.: 'OpenFlow based load balancing for Fat Tree networks with multipath support'. In Proc. 12th IEEE International Conference on Communications(ICC'13), Budapest, Hungary, June. 2013, pp. 1 5
[25] Sabbeh, A., Al Dunainawi, Y., Al Raweshidy, H.S., Abbod, M.F.: 'Performance prediction of software defined network using an artificial neural network'. In Proc. of SAI Computing Conference (SAI), London, UK, July 2016, pp. 80 84
[26] Bendriss, J., Yahia, I.G.B., Chemouil, P., Zeghlache, D.: 'AI for SLA Management in Programmable Networks'. In Proc. of 13th International Conference on Design of Reliable Communication Networks, Munich, Germany, March 2017, pp. 1-8.
[27] Bendriss, J., Yahia, I.G.B., Zeghlache, D.: 'Forecasting and anticipating SLO breaches in programmable networks'. In Proc. of 20th Conference on Innovations in Clouds, Internet and Networks (ICIN), Paris, France, March 2017, pp. 127 134
[28] Mestres, A., Rodriguez Natal, A., Carner, J., et al.: 'Knowledge defined networking', ACM SIGCOMM Computer Communication Review, 2017,47,(3), pp. 2 10
[29] Mihai Gabriel, I., Victor Valeriu, P: 'Achieving DDoS resiliency in a software defined network by intelligent risk assessment based on neural networks and danger theory'. In Proc. of IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI), Budapest, Hungary, Nov. 2014, pp. 319 324
[30] Tang, T.A., Mhamdi, L., McLernon, D., Zaidi, S.A.R., Ghogho, M.: 'Deep Learning Approach for Network Intrusion Detection in Software Defined Networking'. In Proc. of IEEE International Conference on Wireless Networks and Mobile Communications (WINCOM), Fez, Morocco, Oct. 2016, pp. 258 263
[31] Lazaris, A., Prasanna, V. K.: 'DeepFlow: a deep learning framework for software defined measurement'. In Proceedings of the 2nd Workshop on Cloud Assisted Networking, Incheon, Republic of Korea, Dec. 2017, pp. 43 48
[32] Azzouni, A., Boutaba, R. and Pujolle, G.: 'NeuRoute: Predictive dynamic routing for software defined networks'. 2017 13th International Conference on Network and Service Management (CNSM), Tokyo, Japan, Nov., 2017, pp. 1 6
[33] Azzouni, A., Pujolle, G.: 'NeuTM: A neural network-based framework for traffic matrix prediction in SDN'. In NOMS 2018 2018 IEEE/IFIP Network Operations and Management Symposium, April 2018, Taipei, Taiwan
[34] Huang, C.H., Lee, T.H., Chang, L., Lin J.R., Horng G.: 'Adversarial Attacks on SDN Based Deep Learning IDS System'. In Proc. of International Conference on Mobile and Wireless Technology (ICMWT 2018), Hong Kong, China, June 2018, pp. 181-191.
[35] Wang, P., Lin, S.C., Luo, M. 'A framework for QoS aware traffic classification using semi supervised machine learning in SDNs'. IEEE International Conference on Services Computing (SCC), San Francisco, USA, June July 2016, pp.760 765.
[36] Chen, F., Zheng, X.: 'Machine learning based routing pre plan for sdn'. In International Workshop on Multi disciplinary Trends in Artificial Intelligence, Fuzhou, China, Nov 2015, pp. 149 159.
[37] Li, W., Li, G., Yu, X.: 'A fast traffic classification method based on SDN network'. In: Proc. of the 4th International Conference on Electronics, Communications and Networks (CECNET IV), Beijing, China, Dec. 2014, pp. 223 229
[38] Wang, B., Sun, Y., Yuan, C., Xu, X.: 'LESLA: A Smart Solution for SDN enabled mMTC E health Monitoring System'. In Proceedings of the 8th ACM MobiHoc 2018 Workshop on Pervasive Wireless Healthcare Workshop, June 2018, LA, USA
[39] Loog, M.: 'Contrastive pessimistic likelihood estimation for semi supervised classification', IEEE transactions on pattern analysis and machine intelligence, 2016,38,(3), pp. 462 475
[40] Feamster N. The Road to SDN An intellectual history of pro grammable networks. Publisher ACM. 2013; 11(12):1–21
[41] What is SDN. Available from: www.sdxcentral.com/sdn/ definitions/what the definition of software defined net working sdn
[42] Patouni E, Merentitis A, Panagiotopoulos P, Glentis A. Network Virtualization Trends: Virtually Anything Is Possible by Connecting the Unconnected. IEEE SDN for Future Networks and Services; 2013. p. 1–7.
[43] Traditional Vs Software Defined Networking. Available from: www.ipknowledge.net/wp content/uploads/2014/12/ SDN.pdf.
[44] Traditional Network Infrastructure Model and Problems Associated with it. Available from: http://www. pluribusnetworks.com/traditional network infrastructure model and problems associated with it.
[45] Astuto ABB. Nunes, Mendonca M, Nguyen XN, Obraczka K, Turletti T. A Survey of Software Defined Networking Past Present and Future of Programmable Networks. IEEE Communications Surveys & Tutorials. 2014; 16(3):1617– 34
[46] Kaur S, Singh J, Ghumman NS. Network Programmability using POX Controller. Proceedings of International Conference on Communication Computing & Systems (ICCCS). 2014; 1:134–8. PMid:25143894 PMCid:PMC4137641
[47] Floodlight. Available from: http://www.projectfloodlight. org/floodlight/
[48] Lithium. Available from: https://www.opendaylight.org/ lithium
[49] Erickson D. The Beacon OpenFlow controller. Proceedings of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN'13). Publisher ACM; 2013. p. 13–8
[50] Gude N, Koponen T, Pettit J, Pfaff B, Casado M, McKeown N, Shenker S. NOX Towards an operating system for net works. ACM SIGCOMM Computer Communication Review. 2008; 38(3):105–10
[51] Voellmy A, Hudak PN. Functional reactive programming of OpenFlow networks. Proceedings of workshop on Practical Aspects of Declarative Languages; 2011. p.235 49

[52] Jarraya Y, Madi T, Debbabi M. A Survey and a Layered Taxonomy of Software Defined Networking. Published in IEEE Communications Surveys & Tutorials. 2014; 16(4):1955–80
[53] Wang R,Butnariu D,Rexford, Open Flow based server load balancing. Proceedings of workshop on Hot ICE, USA; 2011. p.12.
[54] Sherwood R, Gibb G, Yap KK, Appenzeller G, Casado M, McKeown N, Parulkar G. Can the production network be the testbed. Proceeding of 9th USENIX conference on Operating systems design and implementation. Vancouver Canada Publisher ACM; 2010. p. 365–78
[55] Heller B, Seetharaman S, Mahadevan P, Yiakoumis Y, Sharma P, Banerjee S, McKeown N. ElasticTree Saving energy in data center networks. Proceedings of the 7th USENIX conference on Networked systems design and implementation San Jose; 2010. p. 17–21.
[56] Nayak A, Reimers A, Feamster N, Clark R. Resonance. Dynamic access control in enterprise networks. Proceedings of Workshop on Research on Enterprise Networking Barcelona Spain (WREN'09) Publisher ACM; 2009. p. 11–8.
[57] Erickson D. A demonstration of virtual machine mobility in an Open Flow network. ACM SIGCOMM'08 ACM 978 1 60558 175 0/08/08. Publisher ACM; 2008. p. 1.
[58] Caraguay ALV, Lopez LIB, Villalba LJG. Evolution and Challenges of Software Defined Networking. IEEE Communications Magazine; 2013. p. 1–7.
[59] Mysore RN, Pamboris A, Farrington N, Huang N, Miri P, Radhakrishnan S, Vahda A. Portland a scalable fault toler ant layer 2 data center network fabric. ACM SIGCOMM Computer Communication Review. 2009; 39(4):39–50.
[60] Flood light documentation. Available from: http://www. projectfloodlight.org/documentation/
[61] Heller B, Sherwood R, McKeown N. The controller place ment problem, Proceedings of the First Workshop on Hot Topics in Software Defined Networks (HotSDN'12) Helsinki Finland Publisher ACM; 2012. p. 7–12.
[62] Hunag Y, MinChi C, YaoTing T, YuChieh C, YanRen C. A Novel Design for Future on Demand Service and Security. Proceedings of the 12th IEEE International Conference on Communication Technology (ICCT); 2010. p. 385–8
[63] Porras, Shin P, Yegneswaran S, Fong V, Tyson M, Gu M, A Security Enforcement Kernel for OpenFlow Networks. Proceedings of ACM Workshop on Hot Topics in Software Defined Networks (HotSDN); Helsinki Finland; 2012. p.121–6.
[64] Implementation and Performance Analysis of Firewall on Open vSwitch. Available from: https://pdfs.semantic scholar.org9bb1/4f91b269e72a98d9063d22da2092dc5f5 52d.pdf.
[65] Casado M, Koponen T, Shenker S, Tootoonchian A. Fabric A retrospective on evolving SDN. Proceedings of the ACM Workshop on Hot Topics in Software Defined Networks (HotSDN); Helsinki; 2012. p. 1–5
[66] Lara A, Kolasani A, Ramamurthy B. Network Innovation using OpenFlow A Survey. IEEE Communications Surveys & Tutorials. 2014; 16(1):493–512.
[67] Open Flow Switch Specification. Open Networking Foundation ONF. p. 106 Available from: https://wand.net. nz/~brad/papers/openflow spec v1.5.0.pdf
[68] Shin MK, Nam KH, Kim HJ. Software Defined Network (SDN) A Reference Architecture and Open APIs in ICT Convergence. Proceeding of IEEE International Conference on ICT Convergence (ICTC); Jeju Island; 2012. p. 360–1.
[69] Kim H, Feamster N. Improving Network Management with software defined networking. IEEE Communications Magazine. 2013
[70] Rothenberg EC, Nascimento RM, Salvador MR, Corrêa NAC, Lucena CS, Raszuk R. Revisiting routing control platforms with the eyes and muscles of software defined networking. Proceedings of the ACM Workshop on Hot Topics in Software Defined Networks (HotSDN); 2012. p. 13–7. Crossref PMid:28083220 PMCid:PMC4945262
[71] 32. Bakshi K. Considerations for Software Defined Networking (SDN) Approaches and use cases. Proceedings of the mIEEE Aerospace Conference Big Sky MT; 2013. p.1–9.
[72] Xia W, Wen Y, Foh CH, Niyato D, Xie H. A Survey on Software Defined Networking. IEEE communication sur veys and tutorials. 2015; 17(1):27–51.
[73] Xie H, Tsou T, Lopez D, Sidi R, Yin H, Aranda PA. Software defined Networking Efforts. IETF Journal Published by the Internet Society in cooperation with the Internet Engineering Task Force; 2012.
[74] SDN Series Part Eight Comparison of Open Source SDN Controllers. Available from: https://thenewstack.io/ sdn series part eight comparison of open source SDN controllers.
[75] The NOX Controller. Available from: https://github.com/ noxrepo/nox
[76] Fernandez, Marcial. Evaluating OpenFlow controller para digms. Proceeding of The 3rd ACM SIGCOMM Workshop on Information Centric Networking; 2013. p. 151–7
[77] Tootoonchian A, Gorbunov S, Ganjali Y, Casado M, Sherwood R. On Controller Performance in Software defined networks. Proceeding of the 2nd USENIX conference on Hot Topics in Management of Internet Cloud and Enterprise Network and Services; 2012. p. 10.
[78] Khondoker R, Zaalouk, Marx R. Feature based Comparison and Selection of Software. World Congress on Computer Applications and Information Systems (WCCAIS 2014); 2014. p. 1–7
[79] Gandibleux X. Multiple criteria optimization state of the art annotated bibliographic surveys. Springer Science & Business Media; 2006
[80] Ishizaka A, Labib A. Analytic hierarchy process and expert choice Benefits and limitation. Publisher Springer link. 2009; 22(4):201–20
[81] J David, D Saaty. Use analytic hierarchy process for project selection ASQ Six sigma forum magazine; 2007. p. 1–8 Kotani D, Suzuki K, Shimonishi H. A Design and Implementation of Open Flow Controller handling IP Multicast with Fast Tree Switching. Proceedings of the IEEE/IPSJ International Symposium on Applications and the Internet (SAINT) Izmir Turkey; 2012. p. 60–70.
[82] Bruno Astuto A. Nunes, Marc Mendonca, Xuan Nam Nguyen, Katia Obraczka Thierry Turletti. A Survey of Software Defined Networking Past Present and Future of Programmable Networks. IEEE Communications Surveys & Tutorials. 2014; 16(3):1617–34.

[83] Mattos MFD, Fernandes NC, Costa TV, Cardoso PL, Campista MEM, Costa MKLH, Duarte OMNI. Open Flow Management Infrastructure. Proceedings of the 2011 IEEE International Conference on the Network of the Future; 2011. p. 52–60.
[84] Reitblatt M, Foster N, Rexford J, Schlesinger C, Walker D. Abstractions for Network Update. Proceedings of the ACM SIGCOMM Conference on Applications Technologies Architectures and Protocols for Computer Communication Helsinki Finland; 2012. p. 323–34.
[85] Wang R, Butnariu D, Rexford J. Open Flow Based Server Load Balancing Gone Wild. Proceedings of the USENIX Conference on Hot Topics in Management of Internet Cloud and Enterprise Networks and Services; 2011. p. 12
[86] Boucadair M, Jacquenet C. Service Function Chaining Framework &Architecture; 2014
[87] YuHunag C, MinChi T, YaoTing C, YuChieh C, YanRen C. A Novel Design for Future on Demand Service and Security. Proceedings of the 2010 IEEE 12th International Conference on Communication Technology Publisher IEEE; 2010. p.385–8.
[88] Erickson D. The Beacon OpenFlow controller. Proceedings of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN'13) Publisher ACM; 2013. p. 13–8
[89] Yang M, Li Y, Jin D, Zeng L, Wu X, Vasilakos AV. SoftwareDefined and Virtualized Future Mobile and Wireless Networks A Survey. Springer. 2015; 20(1):4–18

