Research on SDN Enabled by Machine Learning: An Overview

Pu Zhao, Wentao Zhao, and Qiang Liu(✉)

College of Computer, National University of Defense Technology, Changsha 410005, Hunan, China
qiangliu06@nudt.edu.cn

Abstract. Network abstraction brings the birth of Software Defined Network (SDN). SDN is a promising network architecture that separates the control logic the network from the underlying forwarding elements. SDN gives network centralized control ability and provides developers with programmable ability. In this review, the latest advances in the field of artificial intelligence (AI) have provided SDN with learning capabilities and superior decision-making capabilities. In this study, we focus on a sub-field of artificial intelligence: machine learning (ML) and give a brief review of recent researches on introducing ML into SDN. Firstly, we introduce the backgrounds of SDN and ML. Then, we conduct a brief review on existing works about how to apply several typical ML algorithms to SDN. Finally, we give conclusion towards integrating SDN with ML.

Keywords: Software Defined Networking · Machine learning · Artificial intelligence

1 Introduction

The Internet has led to the birth of a digital society in which almost everything is connected and accessible from anywhere. The network usually involves different devices, runs different protocols, and supports different applications. With new network devices, resources and protocols deployed in the network, the network is becoming more and more complex and heterogeneous. Heterogeneous network infrastructure enhances the complexity of the network and brings many challenges in effectively organizing, managing and optimizing network resources [47].

Fortunately, Software Defined Networking (SDN) based on the idea of logically centralized management proposes a simplified solution for complex tasks, such as traffic engineering [36], network optimization [21], orchestration, and so on. In SDN network paradigm, a logic centralized SDN controller manages network devices and arranges network resources. The SDN controller has a overall perspective of the network by monitoring and gathering the timely status of the network and configuration information of the network, and supports a stream

X. Wang et al. (Eds.): 6GN 2020, LNICST 337, pp. 190–203, 2020.
https://doi.org/10.1007/978-3-030-63941-9_14

level resource scheduling of the underlying layers. This kind of creation leads to a huge transformation in the way of networks construction, operation and maintenance. Therefore, SDN framework can be regarded as a means to solve multifarious problems in the network from another perspective, and can also be used to meet the demand of new technologies, such as the IoT and the fifth generation (5G) [8]. As a promising way to rebuild the network, SDN has become the frontier of innovation in industry and academia. The Open Networking Foundation (ONF) is a leader in SDN standardization, and it has the support of more than 100 companies that together accelerate the creation of standards, products, and applications, such as NEC, Google, IBM, and VMware [26].

However, the key to the success of SDN is whether it can effectively solve the problems that can not be well solved in the traditional networking architectures, such as scalability, network awareness, on-demand quality assurance, intelligent traffic scheduling, and so on. Noteworthily, machine learning (ML) provides great potential for SDN innovation. The researches shows that ML technology has been widely used to solve various problems in the network, such as resource allocation, network routing, load balancing, traffic classification, traffic clustering, intrusion detection, fault detection, quality of service (QoS) and quality of experience (QoE) optimization, and so on [25]. Meanwhile, SDN creates conditions for the smooth deployment of ML in the network because of the unique advantages of SDN, such as programmability, global view, centralized control, and so on. Firstly, a mass of data is the key point to implement a data-driven ML algorithm. The SDN controller maintains a overall network view, as well as can monitor and collect all kinds of network data, which can provide a lot of timely and historical data for ML algorithms. Secondly, the optimized solution (e.g., configuration and resource allocation) can be easily deployed in the network due to the programmability of SDN [47]. Therefore, as a subset of artificial intelligence, ML technology has gained more and more the interest of the researchers in the application of SDN technology.

From the view of how to use ML technology to solve the problems faced by SDN, Xie et al. [47] have reviewed the ML technology which can better solve the key problems in the development of SDN, and discussed the research of using ML technology to improve the performance, intelligence, efficiency and security of SDN. On the other hand, from the standpoint of ML key algorithms, we further discuss the methods and characteristics of several typical ML technologies in the application of SDN paradigm. We argue that our work is a complement to research of Xie et al., to better reveal the important role and broad prospects of ML in SDN paradigm. In the paper, we first introduce the backgrounds of SDN and ML. Then, we conduct a brief review on existing works about how to apply widely-used ML algorithms to SDN. Finally, we give conclusion towards integrating SDN with ML.

2 Overview of Software Defined Networking

In this chapter, we will briefly introduce the background of SDN. Firstly, we discusses the background of the birth of SDN, points out the inevitability of the emergence of SDN, then introduces the framework of SDN.

2.1 The Origin and Development of SDN

The distributed control and transport network protocols deployed by the distributed forwarding device are the key point to make traditional Internet successful. However, with the rapid development of network, traditional networks are complex and hard to manage [22]. Therefore, the traditional network architecture needs to be reformed. The related research of programmable network provides a theoretical basis for the generation of SDN [45]. Active network [44,45] allows data packets to carry user programs and can be automatically executed by network devices. Users can dynamically configure the network by programming, which facilitates the management of the network. However, due to the low demand and poor compatibility of protocol, it has not been deployed in the industry. The 4D architecture [18,48] separates the programmable decision plane (i.e. the control plane) from the data plane, centralizes and automates the decision plane. Its design idea generates the rudiment of SDN controller [19]. The term SDN was originally used to describe Stanford's ideas and work around OpenFlow. According to the original definition, SDN refers to a centralized network architecture, in which the data forwarding plane is separated from the distributed control and controlled by a remote centralized controller.

In addition, many standardization organizations have joined in the formulation of SDN standards. The Open Networking Foundation (ONF) is a famous organization specializing in SDN interface standards. The OpenFlow protocol formulated by this organization has become the mainstream standard of SDN interface. Many operators and manufacturers have developed according to this standard. The ForCES Working Group of the Internet Engineering Task Force (IETF), the SDN Research Group of the Internet Research Task Force (IRTF) and several working groups of the International Telecommunication Standardization Sector (ITU-T) also aim at the new methods and new technologies of SDN [12]. The follow-up of standardization organization has promoted the rapid development of SDN market. With the development and application of 5G communication technology, SDN has become an important enabling technology for 5G. Thus, SDN has broad prospects for development and great research value.

2.2 Network Architecture

The design idea of SDN is to separate the control plane of the network from the data forwarding plane to realize a centralized network control and provide a programmable network for developers. Referring to the structure of computer system, there will be three kinds of virtualization concepts in the SDN architecture: forwarding abstraction, distributed state abstraction and configuration

abstraction. According to the original design intention of SDN, the forwarding abstraction should be able to support any forwarding behavior required by network applications, and hide the implementation information of the underlying hardware. Openflow is a practical implementation according to this design idea. Compared with the traditional computer operating system, it can be regarded as the "device driver" in an operating system. At the same time, the SDN applications are not affected by the distributed state of forwarding plane, and they enjoy a unified network view. The unified network view is provided by the distributed state abstraction. The control plane can gather the distribution state information of devices and construct an overall network view so that the applications can set the network uniformly through the whole network state. Configuration abstraction can provide users with a more simplified network model. The users can automatically complete the unified deployment of forwarding devices along the path, through the application interface provided by the control layer. Therefore, network abstraction is the decisive factor for generation of SDN architecture decoupling between data and control planes and providing unified interface.

According to different requirements, many organizations have proposed corresponding SDN reference architectures. SDN Architecture was first proposed by ONF and has been widely accepted in academia and industry. The typical SDN architecture is shown in Fig. 1. SDN consists of three parts: data plane, control plane and application plane. The data plane contains a series of forwarding devices interconnected through wireless channels or wired cables. They are responsible for data processing, forwarding, and status collection based on flow tables. The forwarding plane communicate with the control plane by the southbound interface (SI). The SI defined the communication protocol between the forwarding elements and the controllers, such as OpenFlow protocol. The protocol formalize the way that the controllers and the forwarding elements interact. The control plane includes a series of logically centralized controllers regarded as the brain of the network. The controller is mainly responsible for the arrangement of data plane resources, maintenance of network topology, status information, and so on. The SDN controller can offer the APIs to application developers. The APIs represent the northbound interface (NI), i.e., a common interface for developing applications. The application plane includes a variety of businesses and applications such as load balancers, network routing, firewalls, monitoring, and so on. The network application program communicates with SDN controller by NI to control the network reasonably, so as to realize the business logic of the application program itself.

3 Overview of Machine Learning

The general definition of ML is that intelligent machines learn from experience (i.e. from available data in the environment) and use learned methods to improve overall performance [33,38]. In the case, ML technology can be divided into four groups: supervised, unsupervised, semi-supervised, and reinforcement learning. In this section, Each category is briefly explained to help the reader understand

what follows. A more in-depth discussion of ML technology and the basic concepts about ML, please refer to [33,38].

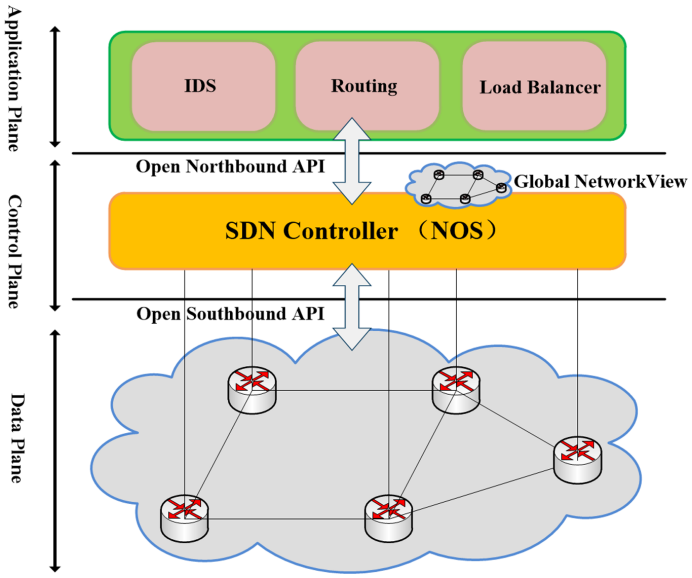

Fig. 1. Illustration of SDN architecture.

3.1 Supervised Learning

Supervised learning methods require predefined knowledge. For example, a training data set consisting of "input-output combinations" in which the model learns a function that maps a given input to an corresponding output [38]. The method needs a test data set that represents the best performance of the current research system. The test data set can be used to evaluate the performance of the final learning method [31].

3.2 Unsupervised Learning

Unsupervised learning is carried out without pre-defined knowledge (that is, only unlabeled data) [38]. Therefore, the system mainly focuses on finding rules or knowledge in the input data. A common use case of unsupervised learning is clustering algorithm, which is used to distinguish meaningful groups in input data according to similar attributes defined by appropriate distance measures (such as Euclidean distance and cosine distance measure) [34,38].

3.3 Semi-supervised Learning

The semi-supervised learning model learns from labeled and unlabeled data. Labeled and unlabeled data may contain random noise in supervised learning and unsupervised learning [38]. As in many practical applications, since the data is labeled manually by experts, it is more realistic to collect many labeled data, while it is easier to collect a large number of unlabeled data [16]. Semi-supervised learning is superior to unsupervised learning because it contains some small labeled data [16].

3.4 Reinforcement Learning

The reinforcement learning (RL) model is based on a set of "reinforcement" in the environment to learn a superior behavior. For example, the system is rewarded or punished according to whether it works well [38]. Every time the system interacts with the environment, it gets feedback information, and it will make full use of the feedback to update its performance [33]. An important property of reinforcement learning is Markov property, because of this property, the subsequent state of reinforcement learning system is determined by the current state [3].

4 Discussion of Applying Machine Learning to SDN

Due to the great efforts of industry and academia, the role of ML in the network has been significantly enhanced. ML technology has been widely used to solve various network-related problems, such as network routing, load balancing, traffic classification and clustering, fault detection, intrusion detection, QoS and QoE optimization, and so on. In this section, we will investigate the application areas of several typical ML methods in SDN.

4.1 Application of Neural Network in SDN

The advantage of neural network (NN) is that it can approximate any function, but because of the need to adjust a large number of parameters, the computational cost is very high. Neural network method is used mainly for intrusion detection [1,14,17], traffic classification [4,28,32], load balancing [15], performance prediction [13,39,41], service level agreement (SLA) execution [6,7], solving the problems of controller placement [2,20] and optimal virtual machine (VM) placement [30], etc.

In this paper, six application examples are discussed. Sander et al. [41] presented the design and performance of DeePCCI, a passive congestion control identification method based on deep learning which only needs to train the traffic of congestion control variables. Compared with the traditional methods, it can be directly applied to encrypted traffic and easier to expand, because it only needs the time of arrival information of the packets. To solve the problem of

weighted controller configuration, He et al. [20] introduced a multi-label classification method to forecast the entire network allocation. Compared with decision tree method and logistic regression method, the neural network method shows superior results and saves up to two-thirds of the running time of the algorithm. Carner et al. [13] compared the performance of traditional methods and neural network methods for network transmission delay prediction. By training a model, network delay is automatically predicted according to traffic load and overlapping routing strategy. The M/M/1 network model and NN model are introduced for network transmission delay prediction in their works. The experimental results show that the network transmission delay predicted based on neural network has better accuracy than the method based on M/M/1 model. In [32], the researchers used an 8-layer deep neural network to identify mobile applications. The quintuple included destination IP address and port number and so on is used to feature a flow, which is the training data of an 8-layer deep NN. The experiment show that the recognition accuracy of the trained model for 200 mobile applications reaches 93.5%. Abubakar et al. [1] introduced a SDN intrusion detection system using the neural network method, which achieves 97.3% high accuracy in NSL-KDD data sets.

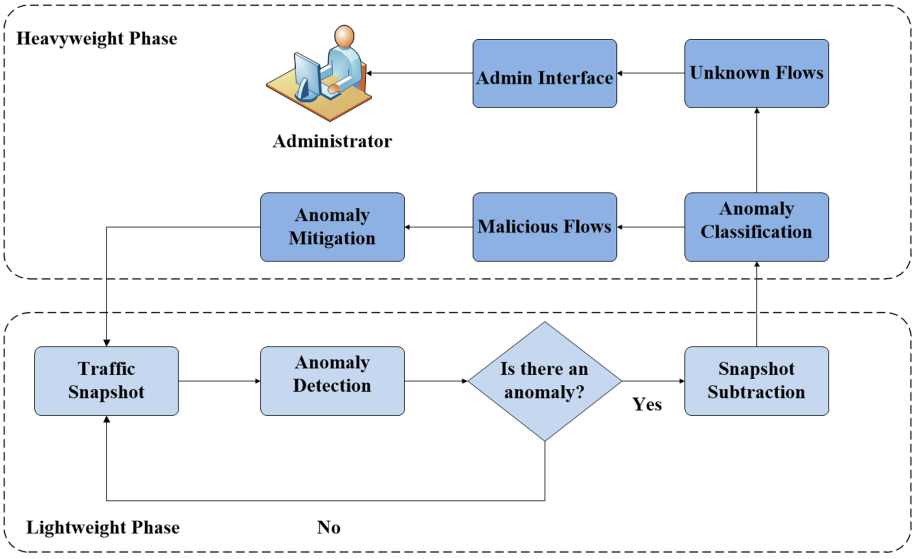

Fig. 2. Atlantic system workflow.

4.2 Application of Support Vector Machine in SDN

The support vector machine (SVM) has the advantage of processing high-dimensional data sets well, but it is difficult to train large data sets because

of the large amount of training calculation. The support vector machine method is mainly used to intrusion detection and traffic classification in SDN paradigm [10,23,23,24,35,37,43,49].

Silva et al. [43] propose a prototype system called Atlantic for joint abnormal traffic detection, classification and mitigation in SDN. The Fig. 2 illustrates a work view of the Atlantic. The Atlantic framework implements anomaly detection and classification in two stages: lightweight stage and heavyweight stage. In the lightweight stage, The authors adopt some methods with low computational cost (such as information theory), The lightweight methods can be called more continually to quickly remark the potential malicious traffic. In the heavyweight stage, by using an SVM algorithm to leverage historical knowledge about past anomalies, the flows are analyzed and classified according to their abnormal behavior. Atlantic then takes appropriate mitigation measures to automatically handle malicious traffic, and human administrators manually analyze unknown traffic. Kokila et al. [23] introduced an approach to detect DDOS attacks on the SDN controllers. Compared with the traditional classifiers, their method based on SVM has higher accuracy and lower error rate. Boero et al. [10] used SVM to detect malicious software based on SDN, and the information gain (IG) measures were used to select the most dependent features. Their models achieve 80% and 95% malware and normal traffic detection rates. Furthermore, the false alarm rates of malware and normal traffic were 5.4% and 18.5% respectively. In [37], the authors implement an application aware traffic classification system using SVM. The system classifies UDP traffic according to NetFlow records (such as received packets and bytes). The experimental results show that the classification accuracy of the model is more than 90%.

4.3 Application of *k*-Means Clustering in SDN

The *k*-means clustering algorithm is easy to implement and explain clustering results, but the calculation cost is linear with the number of training data. *k*-means clustering method is mainly used to deploy intrusion detection system in SDN paradigm [5], routing decision [11], solve the placement problem of optimal controller [40], and analyze user traffic [9].

Bernaille et al. [9] introduced an approach based on a Simple K-Means algorithm that classified different types of TCP-based applications using a first few packets of the flows. Budhraja et al. [11] proposed a routing protocol in a strictly compliant environment. Firstly, the network traffic is divided into multiple risk ratio clusters by a *k*-means algorithm in an offline way. Then, the authors adopt an ant colony optimization (ACO) algorithm to select the path with the least risk of privacy exposure and compliance for a given data transmission session in an online way. Sahoo et al. [40] used *k*-means method to treat the placement of optimal controllers. They compared two kinds of clustering algorithms: *k*-Medoids and *k*-Center. The results of the comparative experiment show that *k*-Center algorithm has superior results than *k*-Medoid algorithm. Barki et al. [5] compared the performance of four ML methods (i.e., naive Bayesian, *k*-nearest

neighbor, k-mean and k-center) in detecting DDoS attacks in SDN. The experiments show that naive Bayesian method achieved the highest detection rate. However, k-means clustering method achieves good results in processing time.

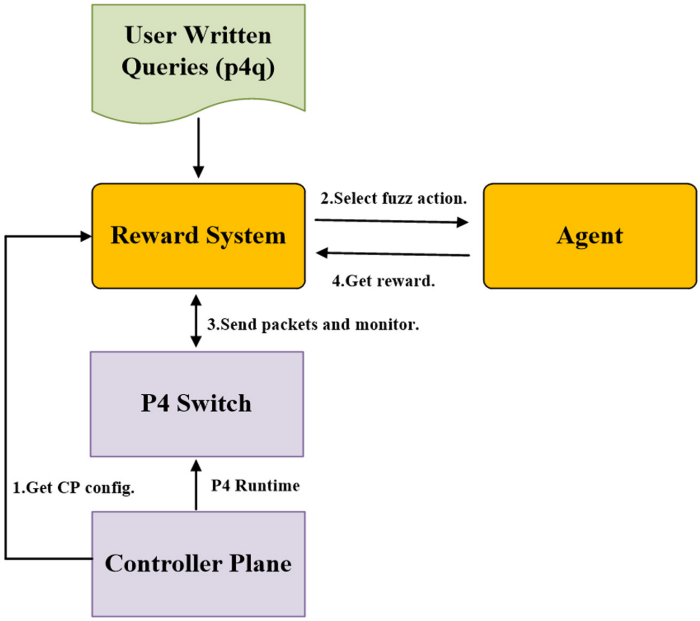

Fig. 3. p4rl system workflow.

4.4 Applications of Other Machine Learning Methods in SDN

In addition to some common ML approaches mentioned above, the combined use of other ML algorithms in SDN broadens our thinking for further expanding the performance of SDN. Apoorv Shukla et al. [42] presents a novel approach for P4 switch verification. They implement a prototype called p4rl which can use reinforcement learning model to guide the fuzz testing to verify the P4 switches automatically during execution time. The Fig. 3 illustrates a work view of the p4rl framework. First, the regulator sets the behavioral attributes of the network to be verified. It coming with configuration information about the network is used as the input of reward system which can provide the basis for verification. At the same time, the authors use an agent based on reinforcement learning to select variant network actions. These mutated actions are used to guide the generation of network packets as test cases. The information about how P4 switch processes the packets is used as the feedback of the reward system. The agent can updates its next action based on the feedback returned by the reward system. The experiment demonstrates that reinforcement learning approach can

make the fuzzing test process focused and improve the test efficiency. In [29], the authors propose an architecture of Self-Driving Network. To enhance the QoE of bandwidth and latency sensitive applications, a ML-based classifier is introduced to class the current experience states of the applications, then the states are sent to a state-machine. If a critical event of the application behavior is detected by the state-machine (arising due to a transition among states), the network controller will execute the corresponding action in order to elevate the performance of the applications. Wang et al. [46] introduced an improved behavior-based SVM to classify network attacks. In order to improve the accuracy of intrusion detection and accelerate the learning rate in the normal mode and intrusion mode, the authors use a decision tree method to reduce features. Firstly, they sort the original feature set and select the features with the best representation. Then, they use these selected features as the input of the model to train a SVM classifier. The model also uses ID3 decision tree method for feature selection. The experimental record on KDD-CUP99 dataset show that the classification accuracy of the model is 97.60%.

4.5 Comparisons of Different Machine Learning Methods Enabling SDN

In order to carry out more in-depth research in ML enabled SDN, we further analyze the characteristics of various ML algorithms in SDN applications.

Overall, the supervised learning algorithm is the common algorithm in intrusion detection, because the core task of the intrusion detection system is often regarded as a classification work. In SDN, ML-based intrusion detection system has been studied extensively. QoS prediction is usually regarded as a regression work, while QoE prediction is regarded as a classification work. Therefore, supervised learning method can also be well handled the QoS or QoE prediction task. Nevertheless, the key to using supervised learning is whether it is convenient to obtain enough labeled training data sets. Compared with supervised learning, the semi-supervised learning approaches only require a small amount of labeled data. Thus, the semi-supervised learning approaches are more easily applied to QoS/QoE prediction. Compared with supervised learning and semi supervised learning, RL algorithm has obvious advantages and application potential in those applications where it is difficult to obtain a large number of training data. On the one hand, the RL algorithm does not require labeled training data sets. Moreover, the optimization objectives (such as network delay, bandwidth utilization, and energy utilization) can be flexibly set through various incentive functions. Specifically, we discuss the advantages and disadvantages of different ML algorithms in Table 1.

Table 1. Strengths and weaknesses of various ML models when applying to SDN

ML Algorithm	Strengths	Weaknesses
Neural Network	Once trained, execution speed is fast The ability to approximate arbitrary functions to predict complex network data Work well on high-dimensional network datasets	Expensive computing makes online training difficult Hard to guide researchers to set structure of NN
SVM	Work well on both linearly separable and non-linearly separable dataset	Hard to train large-scale network datasets Sensitivity to noise Data in SDN
K-Means	Simple to implement deployment in SDN	Sensitive to initial points and outliers Computing increases linearly with the size of network datasets
RL	Working well without prior knowledge can flexibly handle different optimization objectives	Hard to solve problems in high-dimensional space
Semi-supervised learning	Using labeled and unlabeled data to effectively deal with situations where subjective datasets, such as QoE, are difficult to obtain	Rely on assumptions

5 Conclusions

This paper summarizes the research work on the application of ML technology in SDN paradigm. The research shows that an increasing number of the ML technologies are used to solve a wide range of network problems. The ML technologies have been proved to be the valuable means in SDN. The advantages of ML technology in classification, prediction, and feature extraction can better solve the security protection, resource allocation, routing, load balancing, and other issues in SDN. Compared with traditional methods and other artificial intelligence technologies, ML technology shows a broader application. In addition, compared with traditional ML technology, deep learning, can provide better results. However, The ML also brings new challenges to SDN. ML models and related training data are faced with various security risks [27]. More attention should be paid to the robustness of ML in confrontational environments.

Acknowledgement. The work is supported by National Key Research and Development Program of China under Grant No. 2018YFB0204301, National Natural Science Foundation of China under Grant Nos. 61702539 and U1811462, Hunan Provincial Natural Science Foundation of China under Grant No. 2018JJ3611, and NUDT Research Project under Grant No. ZK-18-03-47.

References

1. Abubakar, A., Pranggono, B.: Machine learning based intrusion detection system for software defined networks. In: Proceedings of the 7th International Conference on Emerging Security Technologies, pp. 138–143 (2017)
2. Alvizu, R., Troia, S., Maier, G., Pattavina, A.: Matheuristic with machine learning based prediction for software defined mobile metro core networks. IEEE/OSA J. Opt. Commun. Network. **9**(9), D19–D30 (2017)
3. Arulkumaran, K., Deisenroth, M.P., Brundage, M., Bharath, A.A.: Deep reinforcement learning: a brief survey. IEEE Signal Process. Mag. **34**(6), 26–38 (2017)

4. Ashifuddin Mondal, M., Rehena, Z.: Intelligent traffic congestion classification system using artificial neural network. In: Companion Proceedings of The 2019 World Wide Web Conference, WWW 2019, pp. 110–116. , Association for Computing Machinery, New York (2019). https://doi.org/10.1145/3308560.3317053
5. Barki, L., Shidling, A., Meti, N., Narayan, D.G., Mulla, M.M.: Detection of distributed denial of service attacks in software defined networks. In: Proceedings of the IEEE International Conference on Advances in Computing, Communications and Informatics, pp. 2576–2581 (2017)
6. Bendriss, J., Yahia, I.G.B., Chemouil, P., Zeghlache, D.: AI for SLA management in programmable networks. In: Proceedings of the 13th International Conference on Design of Reliable Communication Networks, pp. 1–8 (2017)
7. Bendriss, J., Yahia, I.G.B., Zeghlache, D.: Forecasting and anticipating SLO breaches in programmable networks. In: Proceedings of the 20th Conference on Innovations in Clouds, Internet and Networks, pp. 127–134 (2017)
8. Bera, S., Misra, S., Vasilakos, A.V.: Software-defined networking for internet of things: a survey. IEEE Internet Things J. **4**(6), 1994–2008 (2017)
9. Bernaille, L., Teixeira, R., Akodkenou, I., Soule, A., Salamatian, K.: Traffic classification on the fly. SIGCOMM Comput. Commun. Rev. **36**(2), 23–26 (2006). https://doi.org/10.1145/1129582.1129589
10. Boero, L., Marchese, M., Zappatore, S.: Support vector machine meets software defined networking in IDS domain. In: Proceedings of the 29th IEEE International Teletraffic Congress, pp. 25–30 (2017)
11. Budhraja, K.K., Malvankar, A., Bahrami, M., Kundu, C., Kundu, A., Singhal, M.: Risk-based packet routing for privacy and compliance-preserving SDN. In: Proceedings of the 10th IEEE International Conference on Cloud Computing, pp. 761–765 (2017)
12. Cai, Z., Zhang, H., Liu, Q., Xiao, Q., Cheang, C.F.: A survey on security-aware network measurement in SDN. Security and Communication Networks **2018**, 14 (2018). Article ID 2459154
13. Carner, J., Mestres, A., Alarcon, E., Cabellos, A.: Machine learning-based network modeling: an artificial neural network model vs a theoretical inspired model. In: Proceedings of the 9th International Conference on Ubiquitous and Future Networks, pp. 2576–2581 (2017)
14. Chen, X.F., Yu, S.Z.: CIPA: a collaborative intrusion prevention architecture for programmable network and SDN. Comput. Secur. **58**, 1–19 (2016)
15. Chen-Xiao, C., Ya-Bin, X.: Research on load balance method in SDN. Int. J. Grid Distrib. Comput. **9**(1), 25–36 (2016)
16. Fan, X., Guo, Z.: A semi-supervised text classification method based on incremental EM algorithm. In: Proceedings of the WASE International Conference on Information Engineering, pp. 211–214 (2010)
17. Gabriel, M.I., Valeriu, V.P.: Achieving DDoS resiliency in a software defined network by intelligent risk assessment based on neural networks and danger theory. In: Proceedings of the 15th IEEE International Symposium on Computational Intelligence and Informatics, pp. 319–324 (2014)
18. Greenberg, A., et al.: A clean slate 4D approach to network control and management. ACM SIGCOMM Comput. Commun. Rev. **35**(5), 41–54 (2005)
19. Gude, N., et al.: NOX: towards an operating system for networks. ACM SIGCOMM Comput. Commun. Rev. **38**(2), 105–110 (2008)
20. He, M., Kalmbach, P., Blenk, A., Kellerer, W., Schmid, S.: Algorithm-data driven optimization of adaptive communication networks. In: Proceedings of the 25th IEEE International Conference on Network Protocols (ICNP), pp. 1–6 (2017)

21. Heorhiadi, V., Reiter, M.K., Sekar, V.: Simplifying software-defined network optimization using SOL. In: Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2016), pp. 223–237 (2016)
22. Jain, R.: Ten problems with current internet architecture and solutions for the next generation. In: Proceedings of the IEEE MILCOM 2006, pp. 1–9 (2006)
23. Kokila, R.T., Selvi, S.T., Govindarajan, K.: DDoS detection and analysis in SDN-based environment using support vector machine classifier. In: Proceedings of the 66h IEEE International Conference on Advanced Computing, pp. 205–210 (2014)
24. Latah, M., Toker, M.: A novel intelligent approach for detecting dos flooding attacks in software defined networks. Int. J. Adv. Intell. Inf. **4**(1), 11–20 (2018)
25. Latah, M., Toker, L.: Artificial intelligence enabled software defined networking: a comprehensive overview (2018)
26. Lin, P., Bi, J., Wolff, S., et al.: A west-east bridge based SDN inter-domain testbed. IEEE Commun. Mag. **53**(2), 190–197 (2015)
27. Liu, Q., Li, P., Zhao, W., Cai, W., Yu, S., Leung, V.C.M.: A survey on security threats and defensive techniques of machine learning: a data driven view. IEEE Access **6**, 12103–12117 (2018)
28. Lyu, Q., Lu, X.: Effective media traffic classification using deep learning. In: Proceedings of the 2019 3rd International Conference on Compute and Data Analysis, ICCDA 2019, pp. 139–146. Association for Computing Machinery, New York (2019). https://doi.org/10.1145/3314545.3316278
29. Madanapalli, S.C., Gharakheili, H.H., Sivaraman, V.: Assisting delay and bandwidth sensitive applications in a self-driving network. In: Proceedings of the 2019 Workshop on Network Meets AI & ML, NetAI 2019, pp. 64–69. Association for Computing Machinery, New York (2019). https://doi.org/10.1145/3341216.3342215
30. Mestres, A., et al.: Knowledge-defined networking. ACM SIGCOMM Comput. Commun. Rev. **47**(3), 2–10 (2017)
31. Mogul, J.C., Congdon, P.: Hey, you darned counters!: Get off my ASIC! In: Proceedings of the ACM SIGCOMM Workshop on HotSDN, pp. 25–30 (2012)
32. Nakao, A., Du, P.: Toward in-network deep machine learning for identifying mobile applications and enabling application specific network slicing. IEICE Trans. Commun. **E101.B**(7), 1536–1543 (2018). https://doi.org/10.1587/transcom.2017CQI0002
33. Negnevitsky, M.: Artificial Intelligence - A Guide to Intelligent Systems, 2nd edn. Addison-Wesley, Essex (2005)
34. Nguyen, T.T., Armitage, G.: A survey of techniques for internet traffic classification using machine learning. IEEE Commun. Surv. Tutor. **10**(4), 56–76 (2008)
35. Phan, T.V., Van Toan, T., Van Tuyen, D., Huong, T.T., Thanh, N.H.: OpenFlowSIA: an optimized protection scheme for software defined networks from flooding attacks. In: Proceedings of the 6th IEEE International Conference on Communications and Electronics, pp. 13–18 (2016)
36. Raza, S., Huang, G., Chuah, C.N., Seetharaman, S., Singh, J.P.: MeasuRouting: a framework for routing assisted traffic monitoring. IEEE/ACM Trans. Network. **20**(1), 45–56 (2012)
37. Rossi, D., Valenti, S.: Fine-grained traffic classification with Netflow data. In: Proceedings of the 6th International Wireless Communications and Mobile Computing Conference, pp. 479–483. Association for Computing Machinery, New York (2010). https://doi.org/10.1145/1815396.1815507
38. Russell, S., Norvig, P.: Artificial Intelligence (A Modern Approach), 3rd edn. Prentice Hall, New Jersey (1995)

39. Sabbeh, A., Al-Dunainawi, Y., Al-Raweshidy, H.S., Abbod, M.F.: Performance prediction of software defined network using an artificial neural network. In: Proceedings of the SAI Computing Conference (SAI), pp. 80–84 (2016)
40. Sahoo, K., Sahoo, S., Mishra, S., Mohanty, S., Sahoo, B.: Analyzing controller placement in software defined networks. In: Proceedings on National Conference on Next Generation Computing and Its Applications in Computer Science and Technology, pp. 12–16 (2016)
41. Sander, C., Rüth, J., Hohlfeld, O., Wehrle, K.: DeePCCI: deep learning-based passive congestion control identification. In: Proceedings of the 2019 Workshop on Network Meets AI & ML, NetAI 2019, pp. 37–43. Association for Computing Machinery, New York (2019). https://doi.org/10.1145/3341216.3342211
42. Shukla, A., Hudemann, K.N., Hecker, A., Schmid, S.: Runtime verification of P4 switches with reinforcement learning. In: Proceedings of the 2019 Workshop on Network Meets AI & ML, NetAI 2019, pp. 1–7. Association for Computing Machinery, New York (2019). https://doi.org/10.1145/3341216.3342206
43. da Silva, A.S., Wickboldt, J.A., Granville, L.Z., Schaeffer-Filho, A.: Atlantic: a framework for anomaly traffic detection, classification, and mitigation in SDN. In: Proceedings of the 2016 IEEE/IFIP Network Operations and Management Symposium, pp. 27–35 (2016)
44. Tennenhouse, D.L., Smith, J.M., Sincoskie, W.D., Wetherall, D., Minden, G.J.: A survey of active network research. IEEE Commun. Mag. **35**(1), 80–86 (1997)
45. Tennenhouse, D.L., Wetherall, D.J.: Towards an active network architecture. ACM SIGCOMM Comput. Commun. Rev. **37**(5), 81–94 (2007)
46. Wang, P., Chao, K.M., Lin, H.C., Lin, W.H., Lo, C.C.: An efficient flow control approach for SDN-based network threat detection and migration using support vector machine. In: Proceedings of the 13th IEEE International Conference on e-Business Engineering, pp. 56–63 (2016)
47. Xie, J., et al.: A survey of machine learning techniques applied to software defined networking (SDN). IEEE Commun. Surv. Tutor. **21**(1), 393–430 (2019)
48. Yan, H., Maltz, D.A., Gogineni, H., Cai, Z.: Tesseract: a 4D network control plane. In: Proceedings of the 4th USENIX Symposium on Networked Systems Design & Implementation (NSDI 2007), pp. 369–382 (2007)
49. Yuan, R., Li, Z., Guan, X., Xu, L.: An SVM-based machine learning method for accurate internet traffic classification. Inf. Syst. Front. **12**(2), 149–156 (2010)