

Software Defined Machine Learning Based Anomaly Detection in Fog Based IoT Network

Qaisar Shafi(✉), Saad Qaisar(✉), and Abdul Basit(✉)

School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad, Pakistan
{qaisar.shafi,saad.qaisar,basit.khan}@seecs.edu.pk

Abstract. Adaptation of intelligent devices with smart connectivity has tremendously increased the Internet of Things (IoT) traffic. For the security of IoTs massive applications, anomaly detection in these large number of devices is resource intensive task. This anomaly detection neither can be done in a cloud where analytical applications run nor in IoT device due to its limited computation capability. The Software Defined Networking (SDN) promises better management of network due to centralized control. In this paper, we proposed a software-defined machine learning (ML) based anomaly detection framework that provides network control in two scenarios, first is cloud infrastructure, and other is collocated fog(network edge) infrastructure. We compare our work in terms of delay in cloud against a collocated fog (processing) node; furthermore, we have also evaluated both scenarios with respect to packet error rate and throughput. Moreover, we discuss that these factors are critical in attack detection and mitigation. In the end, we conclude that in machine learning based anomaly detection, fog nodes provide better computational (attack detection) results in comparison with a cloud infrastructure.

Keywords: SDN · DDoS · Machine learning · Internet of Things · Fog · Cloud

1 Introduction

Connected devices are next-generation networks that interact without human interaction. These devices produce massive data that require pre-processing before use by an analytical application that employs machine learning techniques for enhanced results [1]. Cloud based infrastructure is best fit in this scenario. Moreover, edge computing and Softwarization of network and its numerous applications are also gaining the interest of researchers [2]. These smart systems have applications in transport, grids, health-care, surveillance and many more we can name [4]. Fog based infrastructure [4,21] are considered to be best to do computation at the edge of the network to lodge latency. The Software Defined

© Springer Nature Switzerland AG 2019
S. Misra et al. (Eds.): ICCSA 2019, LNCS 11622, pp. 611–621, 2019.
https://doi.org/10.1007/978-3-030-24305-0_45

Networking (SDN) promises better management of network due to centralized control [5]. In securing application and anomaly detection, computational systems are considering SDN as their candidate that separate control plan from data plan [3].

The massive application of IoT networks also increase the security vulnerabilities in these applications. Intrusion Detection and Prevention System (IDPS) must operate in real-time, delay in mitigation of attack process can block IoT traffic. Anomaly and attack detection systems in combination with machine learning(ML) techniques provide best attack detection results, but this task is resource intensive as well. Collocated fog nodes process the IoT data (anomaly detection) before going towards cloud where already analytical application are running and also avoid processing at IoT device which is already computationally limited. In this paper, we proposed two network perspective methods of machine learning based anomaly detection techniques. First we present, that detection mechanism is located on the cloud that is represented by an intercontinental link/slow or delayed link because is at a distance. In this arrangement the cloud is providing anomaly detection services. In the second scenario, the fog layer is collocated that has a fast link from IoT network and providing anomaly detection services at the network edge.

We evaluated our proposed scheme in terms of a delay from cloud versus collocated fog computational node. Moreover, the packet error rate and throughput are also evaluated to validate our proposed methodology for attack detection. Moreover, we discuss that these factors are critical in attack detection and mitigation. For ML-based anomaly detection we make use of the novel approach of detecting low-intensity and high-intensity DDoS attacks proposed in [6]. The main focus of the work is on network parameters to analyze the cloud and fog computational platforms for effective anomaly detection.

1.1 Research Contributions

Referring to the above discussion, the main contribution of this research is briefed as:

- We devised a system for intrusion detection/prevention (IDPS) for IoT network and provide comparative parameters for computation at cloud versus at network edge (fog).
- For effective anomaly detection (computation), we train machine learning classifiers that provide results in an automated way.
- We implemented, evaluated and analyzed both scenarios (cloud vs. fog) in number of network factors.

The remaining paper is structured as follows. Related work in Sect. 2, followed by main proposed architecture in Sect. 3. Main setup, implementation and result discussion in Sect. 4. In the end, we conclude the work in Sect. 5.

2 Background and Related Work

In this section, we will discuss the background of the proposed work. Intrusion Detection and Prevention System (IDPS) must operate in real-time; delay in mitigation of attack process can chock IoT traffic. Entropy-based schemes are in considering for attack detection due to limitations of volume centric based schemes for detection of passive and active attacks [7]. Entropy- based schemes provides measure for better classification of attacks [8,9]. Using entropy based schemes detects attacks due to the change in the distribution of traffic across network [10]. Multiple features are also utilized to detect notorious attacks. For example, change in the rate of occurrence of destination address points to Distributed denial of service (DDoS). The core idea behind machine learning based anomaly detection and detection of DDoS attacks at the network edge are motivated by the work presented in [1]. Machine learning based techniques are widely used in intrusion detection and anomalous detection methods. The detailed survey is presented in [11]. The SDN and machine learning techniques are coupled and used in many research areas including traffic classification, security, routing and quality of service. The detail work is presented in [12]. The mininet based evaluation of SDN networks is presented in [13]. This work analyzed resources capability and utilization of SDN resources. The work in [14] analyzed throughput and packet loss in different SDN scenario and they also used multiple platforms. Mininet has been using in the development of many SDN based systems. The work is presented in [15] for development of the framework for custom network topology using mininet [16]. Also presented mininet as SDN based testing platforms. The work in [17] also used mininet for prototyping and emulating SDN based networks.

3 Proposed Network (Attack Detection) Architecture

This is section we present the complete system model and architecture of the proposed scheme. Figure 1 represents the two proposed scenario of the system. The lower portion of the figure shows the IoT network connected by a switch. The figure upper portion describe shows the SDN controller is far-apart and it has machine learning based anomaly detection module running in it that will be described in a later section. The main purpose of this fig is to show that the cloud is at distance and it can be represented by an intercontinental link that is slow or delayed link. In this figure, the whole ML-based anomaly detection mechanism will be carried out in cloud which is obviously at some distance as shown using intercontinental link.

The Fig. 1 lower portion illustrates the second scenario of the work in which the ML-based anomaly detector mechanism is collocated with fog node. It means that detection of anomalies is done at the network edge so it is represented by a fast link as shown in the figure. The attacker is trying to launch an attack (DDoS) on IoT network, the proposed system will detect and mitigate the attack in both above arrangements and more effective results will be compared in later

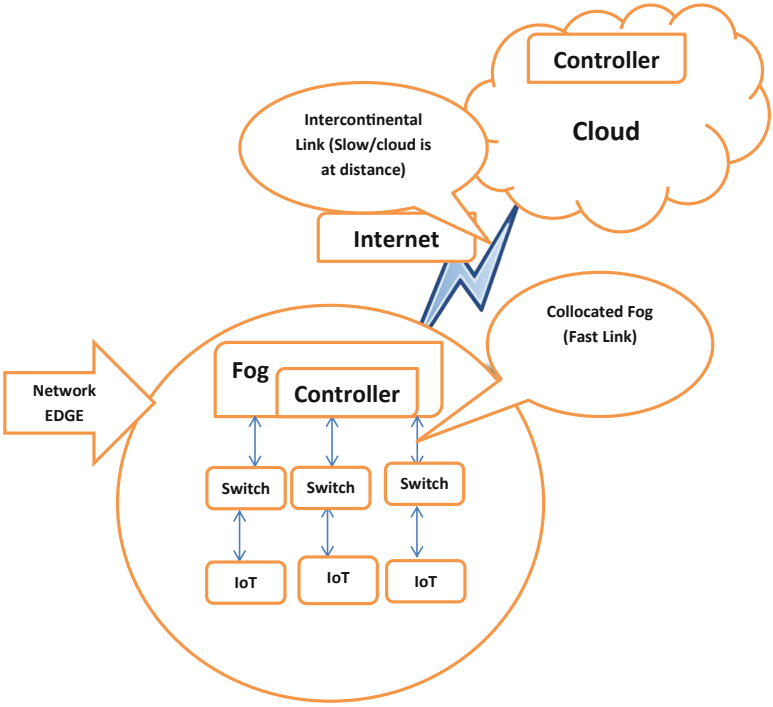

Fig. 1. Network architecture of cloud and collocated Fog

sections. In other words, attack detection is on cloud in the first arrangement and in second scenario attack detection is on the collocated fog node as already explained.

3.1 Anomaly/Attack Detection Methodology

The proposed scheme for ML-based anomaly detection is presented in this section. The idea of feature selection, ML classifier selection, training phase and prediction is motivated from the work in [1]. The anomaly detection offered in this work is the basic computation that we will carry out in this work. It combines the proposed work that this detection is carried out in the cloud or in collocated fog node at the edge of the network. The basic idea of anomaly detection used in this work is that there is a resource manager that assigns resources for labeling features after the selection of features from network traffic as shown in Fig. 2.

Recurrent Neural Network (RNN), Multi-Layer Perceptron (MLP) and Alternate Decision Tree (ADT) form the constituent building blocks of Entropy-based features with Triple Machine Learning Classifier Detection system (E3ML) [6] multi-classifier for detection and prevention of anomalies in IoT network as illustrated in Fig. 2. The idea behind detection protocol is to exploit three ML

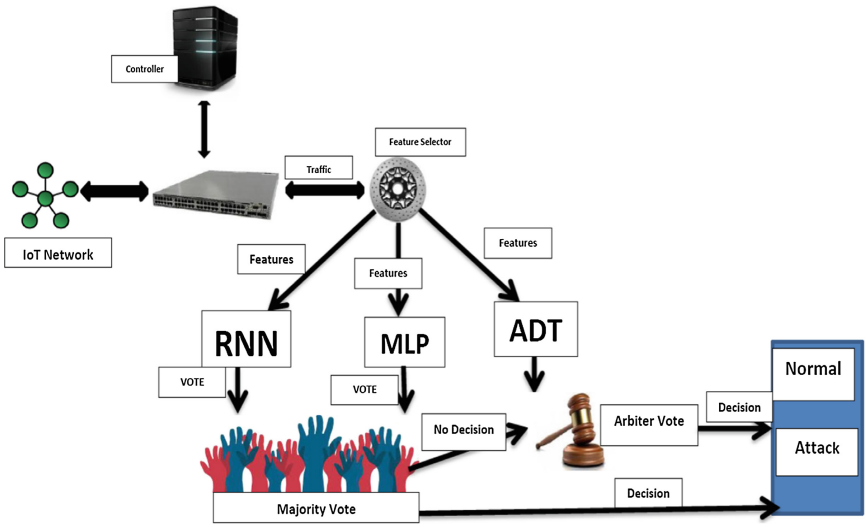

Fig. 2. ML classifier based Anomaly detection module(Multi-classifer System)

classifiers to boost the accuracy of detection of DDoS attacks. Between two standard ML classifiers (i.e. MLP and RNN), voting is carried out as illustrated in Fig. 2. Vote (attack or normal) is contributed by each classifier using its classification results. If attack and normal votes are equal third classifier (i.e. ADT) is introduced that is an arbiter. There is the number of processing units for processing of the data. After suitable processing, it produces labeled attributes. These labeled attributes are dispatched to the allocated processor running entropy calculation code. The code calculates the entropy of each attribute to produce a labeled feature required for traffic classification into normal and malicious categories. We have used a total of twenty (20) features, in contrast to six (06) features based classification model used by a majority of machine learning based intrusion detection systems. In addition to employing more number of features for intrusion detection, we classify traffic with Triple Machine Learning Classifier Detection system (E3CL).

In addition to six known features used by the majority of intrusion detection systems i-e. protocol, delta time, source and destination IP addresses, source and destination port addresses, we have considered another ten (10) independent features in addition to these commonly used features including source and destination MAC as well as network addresses, packet length, Differentiated Services Code Point (DSCP) value, TCP length, its window size and sequence number.

We have also used five (05) extended features constructed through independent features. These features include the rate of separation between source and destination IP, source and destination port, source and destination MAC, source and destination network and source TCP window size and destination TCP segment length, each of which is effective to detect the different version

of the attack. For example, Separation IP tells occurrence of Distributed Denial of Service attack through separation rate between source and destination IP addresses.

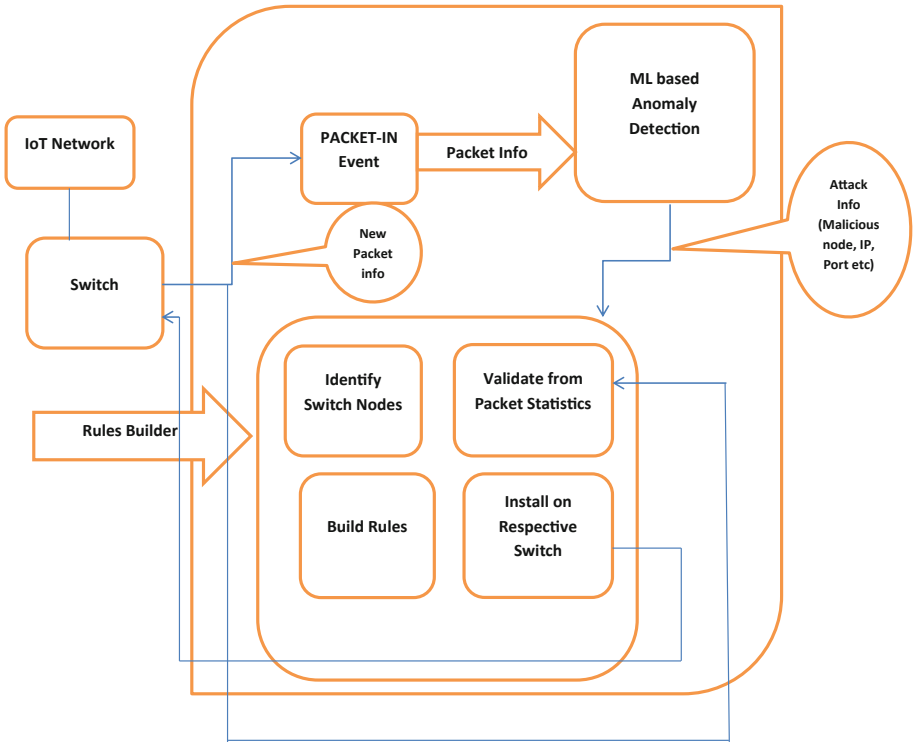

Fig. 3. The controller architecture

3.2 Inside the SDN Controller (Cloud and Fog)

The Fig. 3 shows the inside working of the controller with machine learning modules running in them. The machine learning based attack detection module is triggered with a PACKET-IN messages as shown in figure. The detection rules based on attack feedback and information (features) on which attack is calculated, RULE BUILDER is triggered. Switch inputs VALIDATE PACKET STATISTICS on which module are updated on respective switch that on the bases on packet statistics update and INSTALL RULES OF RESPECTIVE SWITCH. ML based IDPS is already discussed on which attack is detected.

3.3 Attack Mitigation Strategy with SDN Control

SDN has provided us ways to account for attack mitigation. SDN controller of an IoT network controls both in and out traffic flows across the IoT devices. Each controller preserves an access list that comprises of blacklist prefixes. Applicable rules are installed in respective flow tables of all switches so that switches can act as the sink for incoming traffic from blacklist prefixes.

4 Implementation and Experimentation

In this section, we will explain how we have set up our proposed work and how experiments are conducted. We have used the mininet emulation tool to evaluate the proposed framework employing different network topologies and links as shown in Fig. 1. We have used python based Ryu controller in our experiment. A tree topology is created with four(4) depth value and eight(8) fanout value in mininet. Random switches are associated with IoT network (Cooja simulated IoT nodes) using SDN-WISE [18,19].

Appropriate flow rules are installed to setup to verify traffic from malicious nodes. We have used UNSW-NB15 intrusion detection dataset [20] to verify our work. UNSW-NB15 consists of raw network packets in pcap format. They are created by IXIA PerfectStorm tool. Australian Centre for Cyber Security(ACCS) has generated this traffic. It consists of 100 GB traffic in a Pcap separated files. The dataset consists of nine types of attacks as discussed in [1].

The ML-based anomaly detection module is running in the controller are two cases, collocated or far-apart (fog vs cloud). Ryu controller instance is running over Virtual Machine (VM). Controller over each VM is connected to a mininet instance, emulating an area network that contains network switches and IoT devices as mininet hosts. As already discussed, we have evaluated our setup in three criteria. Packet delay as in case of cloud that is located far apart. The distant cloud on which attack detection module is running can take time to stop the attack as packets are delayed that can cause an attack to disseminate in the IoT network.

5 Results and Discussion

The results that we get from our proposed setup are shown in Figs. 4, 5 and 6. In these figures, the network evaluation criteria in terms of cloud and fog architecture are discussed. These evaluation criteria are packet delay in terms of ML-based anomaly detection running on cloud that is far apart versus the fog that is collocated and immediately acts to prevent any anomaly in the system. Second criteria are network throughput that is improved as fog computing layer is located near the IoT network and it helps in effective mitigation of attacks like DDoS that is started to originate in the IoT network. The third, evaluation criteria are network fairness. This criterion relates to varying the data rate and focus is networks packet loss.

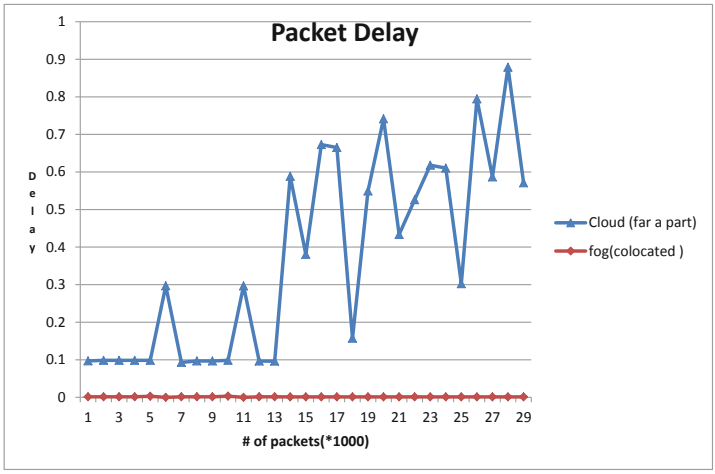

Fig. 4. Packet delay for effective attack detection (cloud vs fog)

Figure 4 shows the packet delay in terms of ML anomaly detection located at distance at cloud versus near (collocated fog) the IoT network. The x-axis shows the number of packets (multiple of 1000) and the y-axis shows the difference of time between sending and receiving of the packets. The cloud shows high packet delay thus more time for anomaly detection method to be implemented for the IoT network. Whereas, the collocated fog shows very less packet delay that will help the controller to analyze anomalous traffic with minimum delay and act as attack mitigation methodology to prevent DDoS and similar 9 attacks in the dataset as discussed earlier.

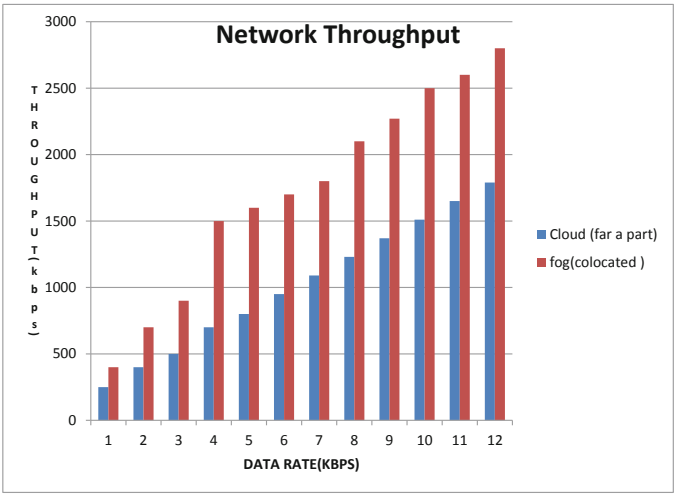

Fig. 5. Network throughput for effective attack detection (cloud vs fog)

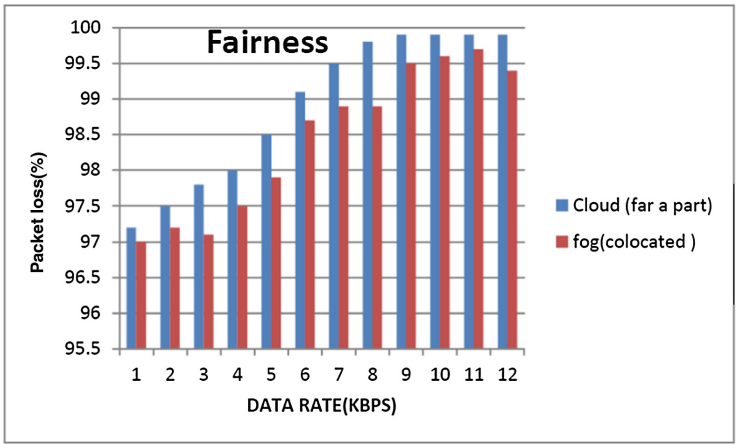

Fig. 6. Network fairness for effective attack detection (cloud vs fog)

The Fig. 5 illustrates the network throughput that is the second criteria on which proposed work is evaluated. The graph shown in this figure illustrates as data rates increase the network throughput also increases. The collocated fog shows better network throughput as compared to the cloud which is located at a distance. In the end, Fig. 6 has shown the fairness of the network against the data rate. The figure shows that amount of data loss in collocated fog is less compared to cloud arrangement. Hence the fairness of the fog arrangement is better than the ML-based anomaly detection module running in cloud infrastructure.

6 Conclusion

In this work, we have taken motivation that anomaly detection neither can be done in a cloud where analytical applications run nor in IoT device due to its limited computation capability. The Software Defined Networking (SDN) potentials better organization of network due to centralized control. In this paper, we proposed a software-defined machine learning (ML) based anomaly detection framework that provides network control in two scenarios, first is cloud Infrastructure, other in collocated fog infrastructure. We compare our work in the number of network parameters including delay in cloud against a collocated fog (processing) node; furthermore, we also evaluated both scenarios with respect to packet error rate, fairness and throughput. We conclude using our results and evaluation that in ML-based anomaly detection fog nodes provide better computational results in comparison with a cloud infrastructure. hence, fog or edge infrastructure provides better attack detection and mitigation system in terms of network arrangements against cloud infrastructure.

Acknowledgement. This work was supported in part by NUST, Ignite Project Xbits and Higher Education Commission, Pakistan.

References

1. Shafi, Q., Basit, A., Qaisar, S., Koay, A., Welch, I.: Fog-assisted SDN controlled framework for enduring anomaly detection in an IoT network. IEEE Access **6**, 73713–73723 (2018)
2. Salahuddin, M.A., Al-Fuqaha, A., Guizani, M., Shuaib, K., Sallabi, F.: Softwarization of internet of things infrastructure for secure and smart healthcare.: arXiv preprint arXiv:1805.11011 (2018)
3. Nunes, B.A.A., Mendonca, M., Nguyen, X.-N., Obraczka, K., Turletti, T.: A survey of software-defined networking: past, present, and future of programmable networks. IEEE Commun. Surv. Tutorials **16**(3), 1617–1634 (2014)
4. Ala, A.-F., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M.: Internet of things: a survey on enabling technologies, protocols, and applications. IEEE Commun. Surv. Tutorials **17**(4), 2347–2376 (2015)
5. Kreutz, D., Ramos, F.M.V., Verissimo, P., Rothenberg, C.E., Azodolmolky, S., Uhlig, S.: Software-defined networking: a comprehensive survey. Proc. IEEE **103**(1), 14–76 (2015)
6. Koay, A., Chen, A., Welch, I., Seah, W.K.G.: A new multi classifier system using entropy-based features in DDoS attack detection. In: 2018 International Conference on Information Networking (ICOIN), pp. 162–167 (2018)
7. Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K.: A multi-step outlier-based anomaly detection approach to network-wide traffic. Inf. Sci. **348**, 243–271 (2016)
8. Bawany, N.Z., Shamsi, J.A., Salah, K.: DDoS attack detection and mitigation using SDN: methods, practices, and solutions. Arab. J. Sci. Eng. **42**(2), 425–441 (2017)
9. Kamboj, P., Trivedi, M.C., Yadav, V.K., Singh, V.K.: Detection techniques of DDoS attacks: a survey. In: 2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON), pp. 675–679. IEEE (2017)
10. Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K.: An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection. Pattern Recogn. Lett. **51**, 1–7 (2015)
11. Mishra, P., Varadharajan, V., Tupakula, U., Pilli, E.S.: A detailed investigation and analysis of using machine learning techniques for intrusion detection. IEEE Commun. Surv. Tutorials **21**, 686–728 (2018)
12. Xie, J., et al.: A survey of machine learning techniques applied to software defined networking (SDN): research issues and challenges. IEEE Commun. Surv. Tutorials **21**(1), 393–430 (2018)
13. Keti, F., Askar, S.: Emulation of software defined networks using mininet in different simulation environments. In: 2015 6th International Conference on Intelligent Systems, Modelling and Simulation, pp. 205–210. IEEE (2015)
14. Tantayakul, K., Dhaou, R., Paillassa, B., Panichpattanakul, W.: Experimental analysis in SDN open source environment. In: 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp. 334–337. IEEE (2017)
15. Veena, S., Pal, C., Rustagi, R.P., Murthy, K.N.B.: A framework for implementing realistic custom network topology in Mininet. Int. J. Sci. Res. (IJSR) **7**, 1316–1323 (2014)
16. Kaur, K., Singh, J., Ghumman, N.S.: Mininet as software defined networking testing platform. In: International Conference on Communication, Computing & Systems (ICCCS), pp. 139–142 (2014)

17. De Oliveira, R.L.S., Schweitzer, C.M., Shinoda, A.A., Prete, L.R.: Using mininet for emulation and prototyping software-defined networks. In: 2014 IEEE Colombian Conference on Communications and Computing (COLCOM), pp. 1–6. IEEE (2014)
18. Galluccio, L., Milardo, S., Morabito, G., Palazzo, S.: SDN-WISE: design, prototyping and experimentation of a stateful SDN solution for Wireless Sensor networks. In: 2015 IEEE Conference on Computer Communications (INFOCOM), pp. 513–521. IEEE (2015)
19. Dunkels, A., Gronvall, B., Voigt, T.: Contiki-a lightweight and flexible operating system for tiny networked sensors. In: 29th annual IEEE international conference on local computer networks, pp. 455–462. IEEE (2004)
20. Moustafa, N., Slay, J.: UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 Military Communications and Information Systems Conference (MilCIS), pp. 1–6. IEEE (2015)
21. Mouradian, C., Naboulsi, D., Yangui, S., Glitho, R.H., Morrow, M.J., Polakos, P.A.: A comprehensive survey on fog computing: State-of-the-art and research challenges. IEEE Commun. Surve. Tutorials **20**(1), 416–464 (2018)