1. momentum

2021 IEEE 25th International Enterprise Distributed Object Computing Workshop (EDOCW) | 978-1-6654-4488-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/EDOCW52865.2021.00031

Intrusion Detection System for SDN-enabled IoT Networks using Machine Learning Techniques

1st Javed Ashraf,
National University of Sciences and Technology (NUST)
Islamabad, Pakistan

2nd N Moustafa , *Senior Member, IEEE*,
School of Engineering and Information Technology, UNSW
Canberra, Australia

3rd Asim D Bukhshi,
National University of Sciences and Technology (NUST)
Islamabad, Pakistan

4th Abdullah Javed,
Sir Syed Centre for Advanced Studies in Engineering (CASE)
Institute of Technology Islamabad, Pakistan

***Abstract*—With the rapid increase in smart devices and lowering prices of sensing devices, adoption of the Internet of Things (IoT) is gaining impetus. These IoT devices come with a greater threat of being attacked or compromised that could lead to the Denial of Service (DoS) and Distributed Denial of Service (DDoS). The high volume of IoT devices with a high level of heterogeneity, amplify the possibility of security threats. So far, the security of IoT devices is a big research challenge. But to enable resilience, continuous monitoring is required along with an adaptive decision making. These challenges can be addressed with the help of Software Defined Networking (SDN) which can effectively handle the security threats to the IoT devices in a dynamic and adaptive manner without any burden on the IoT devices. In this paper, we propose an SDN-based IoT Anomaly detection system which detects abnormal behaviors and attacks as early as possible. Three machine learning (ML) techniques, that is support vector machines (SVM), k-nearest neighbour (KNN) and multilayer perceptron (MLP), are used to design an IDS which is proposed to be deployed at the SDN controller to monitor and learn the behavior of IoT devices over time and any deviation from the normal behaviour is labelled as an attack. We test our algorithms on the two benchmark datasets. We present comparison of the results of the three ML techniques which demonstrate comparable detection accuracy.**

***Index Terms*—Internet of things, software defined networks, intrusion detection system, Machine Learning, SVM, MLP, KNN**

I. Introduction

Internet of Things (IoT) traffic has increased enormously during last five years or so due to the rapid rise in use of the low-cost ubiquitously connected smart devices. Use of these intelligent devices has brought significant improvement in quality of life through provision of the modern services like smart healthcare, smart cities, smart homes, intelligent transport system, and so on. However, the IoT devices, and the networks and systems they are connected to, are highly vulnerable to cyber-attacks. This is because an increasing number of IoT devices amplifies complexity which is required for operation of these devices; and this amplified complexity results in new challenges and threats related to security, privacy and usability. Also, because of the interdependent and interconnected settings of the IoT, new attack surfaces are emerging very frequently [1]. Furthermore, IoT devices cannot afford implementation of the advanced security features because of their restricted energy and computation resources [1]. Some of the examples of attacks initiated from compromised IoT devices include Bashlite and Mirai malware attacks launched from the surveillance cameras and wireless routers which resulted in paralysis of internet based services [2].

Detection of such an attack, called as distributed denial of service (DDoS) attack, using traditional network routers and firewalls is highly challenging, because, due to the stringent memory and complexity constraints of the network devices, it is not practical for an in-line anomaly detection system (ADS) to inspect every packet in detail [3]. It is therefore the destination network's ADS or firewall task to perform the detailed inspection of the traffic flows and drop the anomalous flows/ packets. The Software Defined Networking (SDN) architecture offers that unique flexibility to detect such DDoS attacks. The SDN architecture provides separation of the network's control plane from the data plane [4], [5]. This separation provides a centralized control plane allowing decoupling of the network control from the data plane forwarding. These features of the SDN can be leveraged to implement several security measures to address the IoT security issues mentioned above. SDN-enabled switches can detect and respond earlier to the suspicious traffic flows in the IoT networks. SDN facilitates dynamic flow management and attack prevention against IoT devices through rate-limiting and blocking the malicious flows. The early detection of the suspicious activity facilitates in early isolation of attacking IoT devices as well as mitigation of attacks. By detecting attacks in earlier time-frame, SDN ensures minimizing wastage of network resources, forced by attacking traffic, like DDoS or DoS attacks which consume network bandwidth [6]. And, one of the major advantages of SDN is that, unlike conventional switches and routers, it provides sufficient computing power to run complex, intelligent and configurable algorithms for improved detection accuracy.

In this paper, we describe our work to address issues of IoT security. We propose an SDN-based scheme which prevents attacks generated from network or IoT devices. We

978-1-6654-4488-0/21/$31.00 ©2021 IEEE
DOI 10.1109/EDOCW52865.2021.00031

implement our proposed model on Mininet emulator using Ryu based SDN controller to detect malicious traffic in IoT networks using three machine learning (ML) techniques, that is, Support Vector Machine(SVM), Multilayer Perceptron (MLP) and k-Nearest Neighbours (KNN) .
The rest of the paper is organized as follows. Section II describes main contributions of this work. Related Work is presented in Section III. Overview of SDN and OpenFlow are discussed in Section IV. Section V discusses our proposed scheme of IDS. Experiments and Results are described in Section VI. Finally, the paper is concluded with scope for future work in Section VII.

II. Main Contribution

In this paper, we present a novel work about statistical learning-based anomaly detection in IoT networks, leveraging strengths of SDN. The key contributions of this work are described below: -

- We develop a light statistical feature extraction model which captures the network flow behavior that contributes towards botnet detection.
- We develop ML-SDN based schemes for effectively detecting botnet attacks from IoT devices and IoT based networks.
- We implemented our IDS using mininet emulator for SDN.
- We evaluated our proposed scheme using two benchmark IoT botnet datasets collected from common IoT devices which were infected by several types of botnets and malware. Results demonstrate better or competitive results as compared to other related work.

III. Related Work

There is a large number of papers and research work [1], [6], [7], [8], [9], [10], [11], [12], [13] proposing ML based IoT anomaly detection with or without using SDN technologies. A very comprehensive review of IoT threats, and ML and deep learning based IDS can be found in recently published work [1]. In this section, we describe some of the recent work on SDN-based anomaly detection in IoT network. Tsogbaatar et al. [8] use deep auto-encoders to extract network features and stack it into an ensemble learning model. Then SDN controller is used for deployment of the learned model which is used to detect anomalies in IoT network by handling the class imbalance problem. Bhunia et al. [6] propose SDN and ML based anomaly detection and prevention system called SoftThings. They implement SVM based model at the SDN controller to capture and learn the behavior of IoT devices. Authors use Mininet based SDN emulator and POX based SDN Controller and claim to detect multiple attack types with good accuracy.

In [14], authors propose an entropy based DoS and DDoS attacks detection system in IoT networks using Stateful SDN data plane. Authors use OpenState, an extension of Openflow framework, and Ryu SDN Controller for designing and validating their proposed framework. Authors claim to have achieved very good detection accuracy validated on real IoT data traffic.

In [15], authors propose SDN based distributed collaborative framework which offers DDoS mitigation service for ISPs. In that, ISPs can redirect the malicious traffic to security middleboxes, and attack detection modules are implemented at customer network or devices thus addressing privacy and legal issues. Authors in [16] propose two-stage hierarchical ML based model, integrated into an SDN architecture for anomaly detection and mitigation in the network traffic. The classifier is implemented in SDN controller which serves as the central classifier. The central classifier is used to detect malicious traffic which is then fed to classifier 2, which evaluates per-packet traffic at the network edge. Classifier 2 is implemented at the edge in a processing device co-located with the SDN switch. Authors compare the performance using six ML algorithms: Linear Regression (LR), Linear Discriminant Analysis (LDA), KNN, Classification and Regression Tree (CART), Naive Bayes (NB) and Support Vector Classification (SVC). Authors demonstrate that CART and KNN both produced good detection scores of approximately 99% across all feature-set combinations.

Another IoT related research work proposed in [17], in which authors present a new framework along-with set of techniques which can assist criminal investigators in their investigation process by taking advantages of IoT enabled processes which produce enormous useful data. Authors develop a research prototype and a system which they name as IoT-enabled COP (iCOP). It has three main components: IoT-Enabled Data Collection, Data Transformation and Data Analytics. Authors also present a real-time dashboard which helps the knowledge workers access the useful data in an easy way, as well as provides useful monitoring data. iCOP can benefit from our proposed model in collection of data from IoT devices as well as in ensuring the defense against malicious events which might be generated from the compromised IoT devices which might also affect the validity of the police investigation process.

In the recently published work [18] authors propose the architecture of edge blockchain-assisted SDN (BC-SDN) for flow conformance in an IoT system. Edge-blockchain helps maintain a distributed, immutable flow ledger for the SDN. They propose a design of flow verification and validation for the SDN-IoT system with the assistance of the edge blockchain. To support blockchain technology, they deploy blockchain agents (BCAs) with edge computing servers in order to lessen the computational burden on the IoT systems.

IV. Overview of SDN and OpenFlow

The comprehensive details about SDN and OpenFlow architecture and functioning can be found easily in the literature, like [19]–[24]; in this section, we briefly describe the components and functioning of SDN and OpenFlow. As shown in Fig 1, the brain of SDN is the controller, which is the main component of the control plane. It offers the

interfaces to the other two planes. The northbound interface is used for interface to the application plane, which is a set of APIs to facilitate process of creating network applications. The southbound interface is used for the interaction with the data plane that enables communication between the controller and switches or routers. The OpenFlow [5] protocol is the most common protocol used for southbound interface, developed and maintained by Open Network Foundation [5] and supported by all major network equipment producers. The functioning of OpenFlow switch is briefly described as follows. The OpenFlow switch comprises of group tables and flow tables which facilitate packet lookup and forwarding. The Controller creates, modifies, and removes flow entries using OpenFlow protocol. The secure channel between the controller and switch is established using Transport Layer Security (TLS) or Transmission Control Protocol (TCP). Finally, the Group table maintains the group entries which specify actions applicable to packets sent to specific groups.

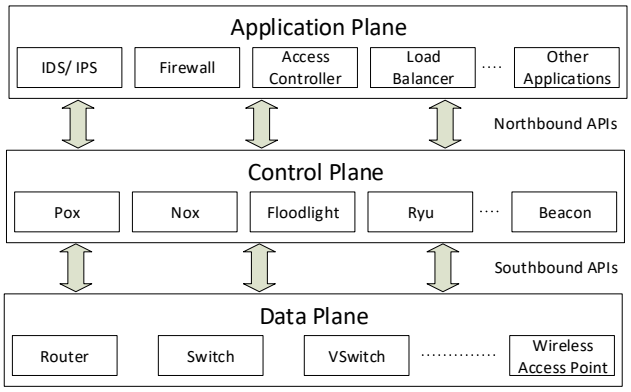

Fig. 1: Illustration of architecture of the SDN.

V. Proposed Scheme for the SDN-Based IDS

This section describes the design and implementation of our proposed ML-SDN based IDS for IOT security.

A. Proposed Model

We present ML technique-based anomaly detection system for IoT leveraging features of SDN framework. The proposed model aims to offer security of IoT devices and networks through employment of IDS at SDN Controller and monitoring and controlling SDN switches. Fig 2 below depicts high level design of our proposed ML-SDN based IoT anomaly detection scheme. The scheme consists of three main components: IoT devices, SDN switch and SDN Controller. The major sub-components of our IDS are housed in SDN controller, which include feature extraction, learning, detection and flow management modules. The SDN Controller segregates policies into service-specific rules and converts into flow tables of the SDN switches by using OpenFlow [5], [25] protocol. Forwarding of each packet is then controlled based on these rules in the flow table. There are fifteen fields that can be stored in flow table against each flow entry in OpenFlow 1.10, while some of the fields are optional, the commonly used fields are: matching fields, priority, actions, statistical counter, and timeout mechanism. Whenever new packet arrives, it is matched with the flow table rules, in case of match the controller must take corresponding action stored in the action field and update the counters. In case match is not found, a new rule is added to the flow table. In the following subsections we discuss data pre-processing, feature extraction, training and detection components of our proposed scheme.

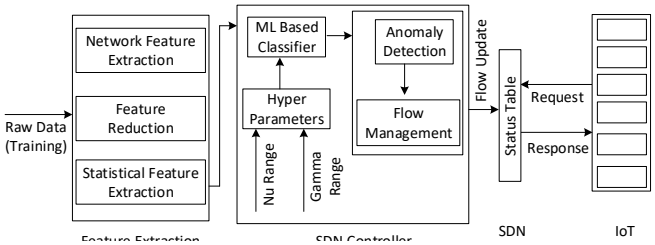

Fig. 2: System Architecture of the Proposed IDS

B. Data Preprocessing

We use two datasets for evaluation of the proposed IDS: 1) UNSW-NB15 dataset [26] and ISCX dataset [27]. Brief description of the two datasets is given as under.

1) UNSW-NB15 dataset: One of the most used datasets for evaluating IDS is UNSW-NB15 dataset [26]. This dataset is available in pcap and comma-separated value(csv) file formats. We selected 13 important features from total of 49 network flow features. The 13 selected features, which contribute to construction of normal behaviour of the network traffic effectively, are: source port, destination port, source IP, destination IP, source bytes, destination bytes, source TTL, destination TTL, source load, destination load, source packets, destination packets and duration. Details about normal and attack data used from UNSW-NB15 dataset are described in Table I.

TABLE I: Data types and sizes of UNSW-NB15 dataset

Attack Type	Number of Packets
Normal	677785
Exploits attack	5408
Generic attack	7522
DoS attack	1167
Fuzzer attack	5051
Recon attack	1759

2) ISCX Dataset: ISCX dataset [27] is one of the many datasets generated by the The Canadian Institute for Cybersecurity(CIC). The dataset consists of 7 days of network activity which includes normal and malicious traffic. In ISCX dataset, the notion of profiles to generate datasets in a systematic manner are used, which contains detailed descriptions of intrusions and abstract distribution models for applications, protocols, or lower level network entities. These profiles can

be used by agents or human operators to generate events on the network. Due to the abstract nature of the generated profiles, we can apply them to a diverse range of network protocols with different topologies. Same 13 features described in above section under UNSW dataset were selected from this dataset also. Details about the normal and attack traffic traces are described in Table II below.

TABLE II: Data types and sizes of ISCX dataset

Attack Type	Number of Packets
Normal	328680
Neris	47126
RBot	39399
IRC Bot	1816
Smokbot	78

C. Feature Extraction

Feature engineering is one of the most important functions in design of an IDS. This section explains the details about feature extraction mechanism.

1) Network Flow Features Extraction: Initially, the raw data is collected from the two datasets in pcap format. The data is converted to csv file which can be used for further analysis. At this stage, the 13 features selected above are now reduced to 5 features by adding values (destination to source and source to destination) of related features, which are: source bytes, time to live (TTL), duration, load and packets. The feature set present the most representative information of their patterns and thus yield discriminate patterns of botnet activities. To encapsulate the behaviour of the hosts which sent the particular packet, three-second snapshot is captured comprising network traffic statistics. The details about the 5 reduced features are summarized as under [7]:

- Traffic generated from the packet's source IP.
- Traffic shared between the packet's source and destination IPs.
- Bits per second for the packet's source to destination and destination to source.
- Time to live from the packet's source to destination and destination to source.
- The count of Packets shared from the packet's source to destination and destination to source.

The same set of features is extracted after time window of every three seconds from both the datasets. The results are then input to statistical feature extraction stage for further action.

2) Statistical Features Extraction: Iterating over a three second time window on the data, the number of packets generated is employed to extract more statistical features that enhance the detection accuracy of the proposed deep learning-enabled IDS. At this stage, statistical features from the two datasets are generated. Table III describes the statistical features extracted from the datasets. The statistical features include mean and standard deviation computed from packets or messages data-frame size and packets' count as shown. It is hypothesized that statistical characteristics of the normal behavior will not change abruptly and the anomalous or attack behavior will result in a sharp change in statistical features, which can be used to detect an attack in comparison to the threshold of normal data.

D. Classifier Design

In this section, we explain the design and components of the proposed classifier.

1) Normalized Likelihood Transformation: After feature extraction, the network flow can be modeled as an independent and identically distributed $N-$dimensional discrete stochastic process $\{\mathbf{x}_m \in \mathbb{R}^N, m = 1, 2, \cdots, M\}$, given M features and N time instants. Assuming the feature subspace $\mathbf{X} = [\mathbf{x}_0, \cdots, \mathbf{x}_{M-1}] \in \mathbb{R}^{N \times M}$ as an M component univariate mixture, a feature likelihood of each element x_{mn} of $\mathbf{X}$ can be estimated using the probability density function, as given by:

$$f_{\mathbf{x}_m}\left(x_{mn}|\mu_m, \sigma_m^2\right) = \left(2\pi\sigma_m^2\right)^{-\frac{1}{2}} \exp\left(-\frac{(x_{mn} - \mu_m)^2}{\sigma_m^2}\right) \tag{1}$$

where

$$\hat{\mu}_m = \frac{1}{N}\sum_{n=0}^{N-1} x_{mn} \tag{2}$$

and

$$\hat{\sigma}_m = \frac{1}{N}\sum_{n=0}^{N-1} (x_{mn} - \hat{\mu}_m)^2 \tag{3}$$

where, $\hat{\sigma}_m$ is the moment estimate for the parameters μ_m, σ_m^2 of the univariate Gaussian distribution for mth feature. The resultant likelihood space

$$\mathbf{Y} = \theta(\mathbf{X}) \tag{4}$$

such that, $\mathbf{Y}$ is a parameterized one-to-one transformation obtained through repeated application of (1) on elements of $\mathbf{X}$. The new parameterized space $\mathbf{Y} = [\mathbf{y}_0, \cdots, \mathbf{y}_{M-1}] \in \mathbb{R}^{N \times M}$ represents the instantaneous Gaussian likelihood corresponding to each parameter of the data. The aggregate normalized likelihood characterization can then be obtained by:

$$\mathbf{z} = \frac{\sum_{m=0}^{M-1} \mathbf{y}_m}{\max \sum_{m=0}^{M-1} \mathbf{y}_m} \tag{5}$$

where $\mathbf{z} = [z_0, \cdots, z_{N-1}] \in [0, 1]$.

2) Supervised Classification of the Normalized Sequence: In order to perform supervised classification of the normalized likelihood sequence $\mathbf{z}$, ground truth labels were generated from the dataset annotations. Binary classification was then performed using three different well known classification schemes, i.e., support vector machines (SVM), k-nearest neighbours (kNN) and multilayer perceptron (MLP).

TABLE III: Statistical features extracted from the datasets.

Packets Feature	Statistical Feature	Aggregated By	# of Features	Explanation
Size	μ, σ	Source_IP	2	Bandwidth of outbound traffic
Time Duration	μ, σ	Source_IP	2	Time duration of network flow from unique IP
Time Duration	μ, σ	Source_IP - Destination_IP	2	Duration of ttl from unique pair of Source-destination IP
Size	μ, σ	Source_IP - Destination_IP	2	Bandwidth of out-bound and inbound traffic together
Count	c	Source_IP - Destination_IP	2	Packet count of outbound and inbound traffic together

3) Detection Methodology: We used mininet SDN network emulator [28] and Ryu [29] SDN Controller in our experiments. Both enable implementation of OpenFlow based switch, communication and the controller. Proposed IDS workflow is depicted in Fig 2. In the training phase the proposed model is trained on normal and attack vectors. The concept is that the botnet attack is generated from the Mininet hosts or from the network. In the detection phase, the OF switch receives the packet and attempts to match it with the flow rules of its table. In case match is found, it means it's a benign request and hence the OF switch forwards the packets as per the existing rule. In case no match is found, then the switch forwards this flow to the controller to request a new flow. The Controller then sends the new forwarding rule. In case of mismatch, the switch will forward the packet to the IDS, if the packets, in one second time window, found to be anomalous, the IDS will notify to the controller and the controller sends the blocking rule to the switch.

E. Experiments and Results

This section describes the evaluation metrics, experiments setting and results of our proposed scheme.

1) Evaluation Metrics: The following evaluation criteria are used to estimate the performance of the proposed IoTBoT-IDS framework: True Positives (TP), True Negatives (TN) that represent rightly predicted results or classifications while False Positives (FP), and False Negatives (FN) denote mis-classified records. TPs and TNs represent correctly predicted abnormal and normal instances, respectively. FPs and FNs denote the wrong classification of legitimate and suspicious examples, respectively [30]. Using these four terms we evaluate our framework by estimating the following measures:

- True Positive Rate (TPR) (also knows as the Detection Rate (DR)) is the rate of rightly identified abnormal instances of the total number of abnormal observations that is given by:

$$TPR = TP/(TP + FN) \tag{6}$$

- True Negative Rate (TNR) (also known as the specificity) is the rate of rightly identified legitimate observations of the total number of legitimate ones that is calculated by:

$$TNR = TN/(TN + FP) \tag{7}$$

- The False Positive Rate (FPR) is the percentage of legitimate vectors of the total number of legitimate vectors wrongly classified as attacks that is estimated by:

$$FPR = FP/(FP + TN) \tag{8}$$

- The False Negative Rate (FNR) is the rate of wrongly identified attack examples of the total number of attack instances that is calculated by:

$$FNR = FN/(FN + TP) \tag{9}$$

- Accuracy is a measure that computes the overall rates of detection and false alarms that an IDS produces, which represents the overall success rate of any IDS; it is calculated by

$$Accuracy = (TP+TN)/(TP+FP+TN+FN) \tag{10}$$

- Precision (PREC) is estimated as the number of correct positive estimations divided by the total number of positive estimations, which is measured by:

$$Precision = TP/TP + FP \tag{11}$$

- Recall is estimated as the number of correct positive estimations divided by the total number of positive estimations and total number of false negative estimates, which is measured by:

$$Recall = TP/TP + FN \tag{12}$$

- F1 Score is a function of precision and recall. It is the harmonic mean of precision and recall, which is measured by:

$$F1 = 2 * (Precision * Recall/Precision + Recall) \tag{13}$$

2) Detection Results: Three different classifiers, i.e., SVM, kNN and MLP were employed to detect attack vectors. The results of each classifier are shown in TABLE IV, TABLE V and TABLE VI, respectively.

TABLE IV: Detection Results for SVM based Decision Engine

UNSW				ISCX			
Attack	Precision	Recall	F1	Attack	Precision	Recall	F1
Dos	95	95	95	IRC_BoT	100	100	100
Exploits	97	97	97	NERIS	93	93	93
Recon	97	97	96	R_bOT	86	87	86
Generic	99	99	99	Smoke_bot	90	95	92

TABLE V: Detection Results for kNN based Decision Engine

UNSW				ISCX			
Attack	Precision	Recall	F1	Attack	Precision	Recall	F1
Dos	95	95	95	IRC_BoT	99	99	99
Exploits	97	97	97	NERIS	98	98	98
Recon	98	98	98	R_bOT	98	98	98
Generic	98	98	98	Smoke_bot	99	99	99

Performance of SVM based classifier remained equally satisfactory when tested on UNSW and ISCX dataset. The SVM classifier performed well in classifying all attack vectors of UNSW dataset, however, precision for R-bot and Smoke-bot remained relatively less in case of ISCX dataset.

In case of kNN algorithm, the proposed engine performed very well on ISCX dataset where it achieved approximately 99% detection rate. Whereas, detection accuracy slightly reduced against UNSW dataset but the difference seems negligible.

Finally, the MLP based classifier's results demonstrate good accuracy in detecting most of the attack types of the both datasets.

TABLE VI: Detection Results for MLP based Decision Engine

UNSW				ISCX			
Attack	Precision	Recall	F1	Attack	Precision	Recall	F1
Dos	95	95	95	IRC_BoT	99	99	99
Exploits	97	97	97	NERIS	98	98	98
Recon	97	97	96	R_bOT	84	85	84
Generic	95	95	95	Smoke_bot	95	97	96

VI. Conclusion

In this paper, we capture 13 important network flow features, which effectively contribute towards construction of normal behaviour of the network traffic, from total of 39 features of the selected datasets. The 13 features are then reduced to 5 features by adding values (destination to source and source to destination) of related features. Then we extracted the same set of features after time window of every three seconds from both the datasets. The results are then input to statistical feature extraction stage where the statistical features from both the datasets are computed. We demonstrated that the statistical characteristics of the malicious traffic differed abruptly in comparison to the characteristics of normal traffic. We believe that the emerging OpenFlow based SDN infrastructure has been able to overcome challenges faced by the conventional networks. We developed SDN based ADS empowered with three ML techniques, to detect anomalous traffic in IoT networks. The botnet attacks discussed above emanating from large number of infected IoT devices or hosts from varying geographical regions can be detected within the core-network. This avoids unavailability of attacked severs which could have been compromised if the botnet attacks were not detected and stopped by the IDS-configured SDN controller deployed in the core network. We proposed that, DDoS attacks launched from botnet compromised IoT devices could be efficiently mitigated before they arrive the targeted network or servers by examining packets at the edge SDN switches. The detection results are competitive or better as compared to other ML techniques based IDS, which were validated by using the two benchmarked datasets.

References

[1] J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, and A. Wahab, "A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions," *Electronics*, vol. 9, no. 7, p. 1177, 2020.

[2] A. Marzano, D. Alexander, O. Fonseca, E. Fazzion, C. Hoepers, K. Steding-Jessen, M. H. Chaves, Í. Cunha, D. Guedes, and W. Meira, "The evolution of bashlite and mirai iot botnets," in *2018 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2018, pp. 00 813–00 818.

[3] S. Ali, I. U. Haq, S. Rizvi, N. Rasheed, U. Sarfraz, S. A. Khayam, and F. Mirza, "On mitigating sampling-induced accuracy loss in traffic anomaly detection systems," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 3, pp. 4–16, 2010.

[4] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[5] "Open networking foundation," 2021. [Online]. Available: https://www.opennetworking.org

[6] S. S. Bhunia and M. Gurusamy, "Dynamic attack detection and mitigation in iot using sdn," in *2017 27th International telecommunication networks and applications conference (ITNAC)*. IEEE, 2017, pp. 1–6.

[7] J. Ashraf, A. D. Bakhshi, N. Moustafa, H. Khurshid, A. Javed, and A. Beheshti, "Novel deep learning-enabled lstm autoencoder architecture for discovering anomalous events from intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2020.

[8] E. Tsogbaatar, M. H. Bhuyan, Y. Taenaka, D. Fall, K. Gonchigsumlaa, E. Elmroth, and Y. Kadobayashi, "Sdn-enabled iot anomaly detection using ensemble learning," in *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer, 2020, pp. 268–280.

[9] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-baiot—network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[10] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," *arXiv preprint arXiv:1802.09089*, 2018.

[11] I. Alrashdi, A. Alqazzaz, E. Aloufi, R. Alharthi, M. Zohdy, and H. Ming, "Ad-iot: Anomaly detection of iot cyberattacks in smart city using machine learning," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, 2019, pp. 0305–0310.

[12] T. D. Nguyen, S. Marchal, M. Miettinen, N. Asokan, and A. Sadeghi, "Diot: A self-learning system for detecting compromised iot devices," *arXiv: Cryptography and Security*, 2018.

[13] J. Ashraf and S. Latif, "Handling intrusion and ddos attacks in software defined networks using machine learning techniques," in *2014 National Software Engineering Conference*. IEEE, 2014, pp. 55–60.

[14] J. Galeano-Brajones, J. Carmona-Murillo, J. F. Valenzuela-Valdés, and F. Luna-Valero, "Detection and mitigation of dos and ddos attacks in iot-based stateful sdn: An experimental approach," *Sensors*, vol. 20, no. 3, p. 816, 2020.

[15] R. Sahay, G. Blanc, Z. Zhang, and H. Debar, "Towards autonomic ddos mitigation using software defined networking," 2015.

[16] P. Amangele, M. J. Reed, M. Al-Naday, N. Thomos, and M. Nowak, "Hierarchical machine learning for iot anomaly detection in sdn," in *2019 International Conference on Information Technologies (InfoTech)*, 2019, pp. 1–4.

[17] F. Schiliro, A. Beheshti, S. Ghodratnama, F. Amouzgar, B. Benatallah, J. Yang, Q. Z. Sheng, F. Casati, and H. R. Motahari-Nezhad, "icop: Iot-enabled policing processes," in *International Conference on Service-Oriented Computing*. Springer, 2018, pp. 447–452.

[18] J. Hu, M. Reed, N. Thomos, M. F. Al-Naday, and K. Yang, "Securing sdn-controlled iot networks through edge blockchain," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2102–2115, 2021.

[19] J. Singh and S. Behal, "Detection and mitigation of ddos attacks in sdn: A comprehensive review, research challenges and future directions," *Computer Science Review*, vol. 37, p. 100279, 2020.

[20] B. Isyaku, M. S. Mohd Zahid, M. Bte Kamat, K. Abu Bakar, and F. A. Ghaleb, "Software defined networking flow table management of openflow switches performance and security challenges: A survey," *Future Internet*, vol. 12, no. 9, p. 147, 2020.
[21] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 27–51, 2015.
[22] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Communications surveys & tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.
[23] A. Lara, A. Kolasani, and B. Ramamurthy, "Network innovation using openflow: A survey," *IEEE communications surveys & tutorials*, vol. 16, no. 1, pp. 493–512, 2013.
[24] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 27–51, 2014.
[25] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM computer communication review*, vol. 38, no. 2, pp. 69–74, 2008.
[26] N. Moustafa and J. Slay, "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)*. IEEE, 2015, pp. 1–6.
[27] E. B. Beigi, H. H. Jazi, N. Stakhanova, and A. A. Ghorbani, "Towards effective feature selection in machine learning-based botnet detection approaches," in *2014 IEEE Conference on Communications and Network Security*. IEEE, 2014, pp. 247–255.
[28] "Mininet an instant virtual network on your laptop (or other pc)," 2021. [Online]. Available: mininet.org
[29] "Welcome to ryu the network operating system(nos)," 2021. [Online]. Available: https://ryu.readthedocs.io/en/latest/index.html
[30] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 303–336, 2014.