Al-Khwarizmi Engineering Journal
ISSN (printed): 1818 – 1171, ISSN (online): 2312 – 0789
Vol. 21, No. 3, Sptember, (2025), pp. 35- 47

Internet of Things Software-Defined Network Intrusion Detection Dataset: Inter- and Intra-Domain

Heba Dhirar[1]*, and Ali H. Hamad[2]
[1,2] *Department of Information and Communication Engineering, AL-Khwarizmi College of Engineering, University of Baghdad, Baghdad, Iraq*
*Corresponding Author's Email: heba.d@kecbu.uobaghdad.edu.iq

(Received 17 January 2024; Revised 20 May 2024; Accepted 28 August 2024; Published 1September 2025)
https://doi.org/10.22153/kej.2025.08.001

Abstract

Software-defined networks (SDNs) are extensively deployed in many network configurations. However, the development of new technology presents several vulnerabilities and risks that continue to pose challenges for manufacturers in addressing them. One of the primary obstacles encountered in deploying an intrusion detection system (IDS) is the absence of an openly accessible dataset, especially one obtained from SDN and SDN-based Internet of Things (IoT) networks. This work produces a comprehensive dataset to evaluate the effectiveness of anomaly-based IDSs in detecting inter- and intradomain attacks. The dataset comprises 86 features extracted from approximately 40 million records obtained from simulated SDN-based IoT networks captured within two flow profiles representing normal and 15 different attack types. In addition, the evaluation is demonstrated by employing six widely used machine learning and deep learning approaches for IDSs: decision tree classifiers, random forest classifiers, deep neural networks, K-nearest neighbours, Bernoulli naive Bayes, and logistic regression.

Keywords: *Dataset; intrusion detection system (IDS); Internet of Things (IoT); software-defined network (SDN); threat; attack vector; network intrusion dataset.*

1. Introduction

The increasing number of personal devices and Internet-enabled technology has led to increased complexity in computer networks. Furthermore, the increased adoption of cloud services has resulted in an unusual expansion of private and public cloud services. In such contexts, the implementation of effective management and the prevention of configuration errors present significant challenges. Whilst the Internet of Things (IoT) enables the connection and exchange of data amongst a multitude of devices [1,2], increasing IoT advancement contributes to the complexity of this issue [3]. This concern is addressed by implementing software-defined network (SDN) technology to transform network architecture and operations. SDNs facilitate the development of network services by using software applications and open application programming interfaces under centralised supervision. The approach centralises administration by abstracting the control plane from the data-forwarding function [4]. The use of centralised management can effectively simplify network administration processes and reduce the occurrence of configuration errors.

In practical implementation, SDN continues to face numerous security challenges, such as the presence of insider threats. These threats involve the exploitation of internal devices by unauthorised individuals to launch attacks on other devices within the network, including the central controller. Security measures are improved by implementing an intrusion detection system (IDS) in every node to assist in detecting and identifying potentially harmful nodes. In the context of IDSs, two primary approaches are commonly utilised: signature-based and anomaly-based. The signature-based approach

is frequently utilised in retail items, which can be attributed to its notable efficacy in detecting known threats and its ability to minimise false positives. However, this method falls short of identifying newer or unfamiliar network attacks that occur daily. However, anomaly-based detection systems have garnered significant interest from researchers in the academic community because of their capacity to identify previously unknown threats. The absence of publicly available datasets, specifically those derived from SDNs, is a significant obstacle to the deployment of IDSs. This challenge needs to be addressed, and researchers, industry stakeholders and regulatory agencies need to work together and create standardised datasets that adequately represent the complexities of modern network systems [5].

The primary objective of this work is to provide a dataset that encompasses the key attributes necessary for the identification of attacks and the assessment of anomaly-based IDSs in the context of SDN-based IoT environments. The work consists of two distinct parts: the initial module gathers traffic statistics to generate the dataset, whereas the subsequent module employs a method based on machine learning (ML) and deep learning (DL) to evaluate the generated dataset. The proposed dataset involves the inclusion of a diverse range of attacks, the implementation of a multidomain attack scenario and the utilisation of realistic traffic traces. Interdomain refers to an attack that occurs between distinct systems or entities. These attacks usually entail a criminal focusing on a system, network or organisation that is distinct from their own and an intra-attack, which refers to an attack that occurs within the same system or entity. These attacks involve malicious actions conducted by people or collectives who already possess entry to the system or organisation being targeted. The contributions of this research can be summarised as follows:

• The advantages of the current IDS, SDN and IoT datasets were analysed.
• Two traffic profiles, namely, the normal profile and the attack profile, were used.
• A total of 15 attack types were considered for two scenarios, namely, interdomain attack and intra-attack.
• The dataset was empirically evaluated via ML and DL methodologies.

The paper is structured as follows: Section 2 presents a comprehensive review of the relevant literature. Section 3 briefly examines the various attack vectors that target SDN and IoT networks. Sections 4 and 5 describe the experimental strategy and its execution and the results. Section 6 presents the conclusions drawn from the work.

2. Related Work

This section presents a summary of the well-known dataset that is most similar to the proposed dataset and has been used in the field of intrusion detection.

BoT–IoT [6]: This dataset contains 46 features that include normal network traffic and six types of attack traffic: DoS, DDoS, keylogging, service scanning, OS fingerprinting and data theft obtained from a simulated IoT network. Nevertheless, compared with others, the dataset is imbalanced, with certain attack types having many fewer entries. The data theft instances total 118, and keylogging instances total 1469, which cannot meet the requirements of the majority of ML algorithms.

SDN–IoT [7]: Two datasets with topologies, characteristics and packet transmission rates comparable to those of the BoT–IoT dataset were generated. The distinguishing factor between the datasets lies in the number of simulated IoT devices employed to evaluate the efficacy of attack detection models trained on these datasets. The datasets contain 33 extracted features from normal traffic and five attack types: DDoS, DoS, OS fingerprinting, port scanning and fuzzing. These attacks were implemented with diverse targets and configurations. This dataset represents the initial instance of a publicly accessible intrusion detection dataset specifically designed for SDN in the context of the IoT.

InSDN [8]: This dataset comprises an estimated number of flow-based features and covers a wide range of attack conditions. The authors used two profiles to generate normal and attack traffic patterns within a simulated SDN architecture. The dataset contains 83 packet properties that were collected from several common network attack types, such as botnets, DoS, web attacks, DDoS, password brute force attacks, probes and tool-based exploits.

DDoS–SDN [9]: This dataset contains 16 features obtained from the SDN to detect DDoS attacks. A hybrid model was employed, combining the support vector classifier (SVC) and random forest (RF) algorithms, for traffic classification. The initial classification was performed via the SVC, and the results were subsequently filtered via the RF algorithm. The model was trained on the generated dataset and achieved an accuracy rate of 98.8%.

ToN–IoT [10]: This dataset comprises a diverse range of regular and adversarial occurrences across several IoTs and industrial IoT services. This dataset comprises data from heterogeneous sources, which were gathered through a test that accurately

emulates an IoT architecture facilitating communication between edge, fog and cloud stages.

However, as neither sensor measurements nor IoT network traffic is present in these datasets, they do not incorporate the unique properties of IoT/IoT applications, even though some researchers have evaluated their IoT-related intrusion detection technologies. In addition, some datasets, such as KDD–cup 99 [11,12], NSL–KDD [13], ISCX [14] and CAIDA [15], have been specifically developed for conventional networks and extensively utilised in research related to IDSs for SDN. Notably, the existing datasets available for analysis have not been specifically designed or generated within the context of an SDN environment. Table 1 presents a comparative analysis between the previously mentioned dataset and the dataset proposed in this work.

In this work, several tools were employed in conducting a range of attack scenarios, such as DoS, DDoS, brute force, Prope, Mirai, exploitation and botnet attacks. Multiple attack scenarios have been established multiple times to address a range of attacks that target multiple hosts to encompass the inter- and intradomains. In addition, a range of widely used application services have been addressed, including HTTP, DNS, FTP, YouTube, Facebook and browsing.

3. Attack Vector

The introduction of a centralised design within the context of SDN architecture brings forward potential vulnerabilities capable of compromising the functionality and security of an SDN susceptible to a range of security attacks. This section presents a review of several attack types and vectors.

3.1. SDN Attack Vectors

Various types of attacks can inevitably exploit vulnerabilities present in all layers of an SDN [16]. Certain attacks are specifically targeted at SDN. These attacks have the potential to manifest within the SDN controller itself or via the communication routes connecting control and data plane devices. In addition, additional breaches are common in SDN standards and traditional networks. The perpetrator must increase their privileges or utilise the compromised workstation to initiate subsequent attacks on various devices or subnets [17].

Table 1.
Comprehensive overview of IDS datasets.

Dataset	Year	Attribute	Environment	Attacks types
BoT-IoT[6]	2018	42	IoT Network	DoS, DDoS, Theft, and Reconnaissance in addition to normal traffic data.
SDN-IoT[7]	2020	33	SDN-Based IoT	DDoS, DoS, OS fingerprinting, port scanning and fuzzing
InSDN[8]	2020	83	SDN Network	Botnet, DoS, DDoS, Brute-Forcing, Probe, Web attacks, exploitation using various tools (Ares, Slowhttptest, LOIC, HULK, Nping, torshammer, Metasploit, SQLMap, Hping3, Hydra, Burp Suite, NMap).
DDoS-SDN[9]	2021	16	SDN Network	DDoS
ToN-IoT[10]	2021	44	IoT Network	DDoS, Cross-site Scripting, ransomware, injection, and backdoor in addition to normal traffic data using Node-RED.
Proposed Dataset	2023	84	SDN-Based IoT	Botnet, Brute-Force, Dos, DDos (ICMP, SYN, UDP), Exploitation, Malware, MIRAI, Probe, R2L, UR2, Web-based, Spoofing, Recon using various tools in addition to normal traffic data.

Different mitigation strategies must be employed to address these challenges. The primary forms of attacks that target SDNs can be categorised into four distinct types (Fig. 1.A).

Data plane: Intruders may target the network elements themselves with the ability to obtain illegal entry into susceptible hosts within a system.

Control plane: The act of executing a flooding attack by using fake sources has the potential to induce congestion inside the channel links. Consequently, the disruption of communication between the SDN controller and the data plane components could isolate the SDN controller from the remaining network elements. In addition, the attacker can manipulate the trust that is formed

between OpenFlow switches with the controller to execute a man-in-the-middle attack, intercept valuable data or obtain complete control over the controller level [18].

Application layer: The potential adversary can execute a harmful software programme to breach the established security protocols or bypass the firewall and IDS applications.

Attacks on the control plane are reportedly exclusive to SDNs, arising from the separation of the data and the control plane, although attacks on data and application planes are prevalent across SDN and traditional networks.

3.2. IOT Attack Vectors

The Open Web Application Security Project [19] released an extensive preliminary document outlining the attack surfaces of IoT devices. These attack surfaces refer to the specific places within IoT systems and applications that are susceptible to vulnerabilities and potential security risks and can be categorised into three distinct types (Fig. 1.B).

Devices: These tools serve as the principal means by which attacks can be established. The primary components of the device, including firmware, memory, a web interface, network surfaces and a physical interface, exhibit vulnerabilities. The attackers can exploit the default configuration, vulnerable update mechanisms and outdated components.

The attackers can exploit the default configuration, vulnerable update mechanisms and outdated components.

Communication: The security of communication channels is a subject of concern. The channel serves as a means of connecting IoT devices with the external environment. The communication protocol utilised in IoT networks has security vulnerabilities that have the potential to impact the overall system.

Application software: This poses a significant threat to the security of online software and apps utilised on IoT devices. These vulnerabilities can lead to system compromise. The online application can illegally acquire user data and introduce harmful upgrades into the system.

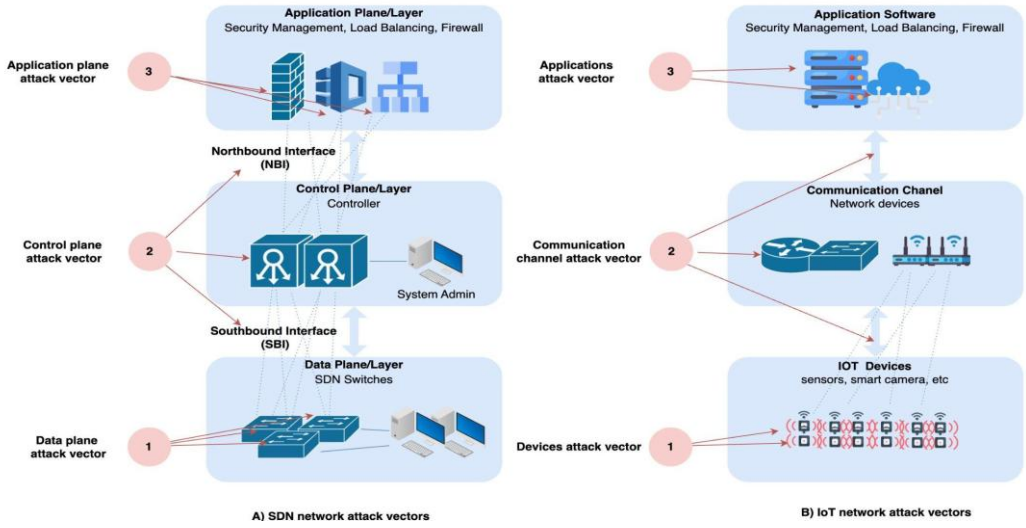

Fig. 1. A) SDN attack vectors, B) IoT network attack vectors.

3.3. Background

Attackers can use numerous methods to breach IT systems, and the majority of cyber-attacks employ similar strategies. The most common attack types are as follows:

Scanning: The initial phase of an attack involves the process of scanning, which entails the collection of information about a target system. This feature includes identifying accessible ports and accessible services on the target device or sensor.

DoS attack: The attacker also has the potential to rapidly deplete the resources of the SDN controller by overwhelming the targeted system with a substantial volume of fake packets that lack corresponding rules inside the flow tables of switches. Notably, the SDN controller serves as the central decision-making entity within the SDN; hence, the entire system becomes inaccessible to authorised users. This phenomenon results in the transformation of the network as a whole into an entity lacking cognitive capabilities.

DDoS attack: The dataset comprises many DDoS attack scenarios, including UDP flooding, ICMP flooding and TCP-SYN flooding attacks. The Hping3 programme is well recognised as one of the tools most often used for conducting DDoS attacks. Brute force attacks: Gaining unauthorised access to a victim's workstation by attempting to crack the login credential information by creating a comprehensive dictionary of all conceivable combinations of usernames and passwords. The attacker machine utilised in this scenario is Kali Linux, whereas the target is the server.

Web-based attack: According to the 2018 research by Symantec [20], approximately 10% of the URLs tested were susceptible to web application attacks, indicating a significant 56% increase compared with the previous year.

Probe attack: This preliminary stage is crucial for an attacker to gather important information before commencing their attack. The attacker conducts a systematic examination of the target system to obtain pertinent information that can facilitate the exploitation of the remote system, including details regarding the operating system's versions and open ports, amongst other factors. The use of the Metasploit platform is employed to identify the accessible ports and vulnerabilities inside the web applications present.

Botnet attack: The unauthorised individual can manipulate several compromised devices, known as a botnet, to execute various harmful actions, such as data theft, fraudulent attacks and distributed DDoS attacks against targeted servers or compromising web application servers. The Botnet attack is executed by utilising the Metasploit platform; in addition, a further instance of IoT attacks is the Mirai botnet. The botnet executed high-volume distributed DDoS attacks [21]. Therefore, an efficient and precises security mechanism is needed to safeguard IoT applications.

U2R attack: These malicious activities resemble regular network traffic and represent a significant risk to the network. Therefore, attacks need to be identified and detected promptly [22]. The utilisation of the Metasploit framework facilitates the acquisition of root privileges on the targeted machine.

4. Experimental Setup

4.1. Proposed Architecture

The construction of a substantial dataset requires a range of application services to be implemented within an experiment-generated dataset. In this manner, contemporary attacks over the Internet, which can be executed within today's SDNs, can be represented accurately. Furthermore, the attack scenarios must include existing attack routes across various SDN components. Moreover, a variety of attack scenarios are being investigated, originating from various sources intra- and interdomain to the SDN. The topology is represented by constructing four subdomains by utilising MiniNet on the Ubuntu 20.04 LTS operating system. Two Ryu controllers are charged for managing the four OpenFlow vSwitch (OVS) switches that connect to these subdomains. The Ryu controller is an open-source software that is based on the Python language. This controller is widely recognised for its ability to manage demands efficiently and effectively in modern networking settings whilst also being scalable; it is also compatible with multiple SDN protocols and features, making it suitable for a range of networking scenarios and interactions with numerous technologies. The first two subdomains pertain to a conventional network that comprises various services, such as HTTP and FTP servers. By contrast, the latter two subdomains correspond to an IoT network specifically designed for conducting experiments indoors and outdoors.

Two base station nodes serve as the root of the tree, which have IDs 1 and 11 (Fig. 2). Additionally, nine sensor nodes with IDs are in the ranges of 2–10 and 12–20. Each subdomain comprises an access point that serves three mobile stations, resulting in a total of 12 stations and 20 sensing nodes in the entire network. Each station and sensing node has been configured with a limited transmission range to facilitate indoor and outdoor traffic monitoring.

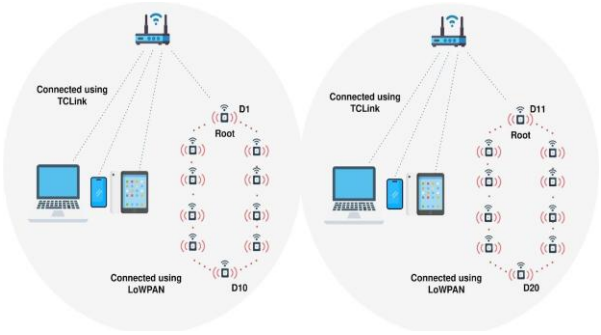

Fig. 2. Root sensing node.

Data transmission must occur through an access point. A mobility feature has been activated to enable mobility-based data collection. In addition to the utilisation of attack machines, which are Linux-based Metasploitable, this system is employed to offer vulnerable services that serve as a means to demonstrate prevalent vulnerabilities. The OVS

switch is configured to operate as a layer-3 switch by integrating the OVS software with the routing capabilities of the Linux kernel. In this scenario, the various virtual hosts can establish communication amongst themselves by utilising distinct subdomains.

The subsequent procedure outlines the process of topology construction and execution:

• A MiniNet topology consisting of eight virtual hosts (h1 to h8) is designed.

• The virtual hosts within the MiniNet network are connected to distinct OVS switches, namely, SA1, SA1, SB1 and SB2. These switches are further linked to two additional OVS switches, SA and SB, to establish a tree topology.

• The sensors are connected to the access point via TCLink and are interconnected with each other via LoWPAN.

• Ryu controllers are initiated for the established network nodes, including switches and access points.

Successful network connectivity can then be established by sending ICMP echo requests (pings) between hosts located in distinct subnetworks. Fig. 3 illustrates the proposed architecture.

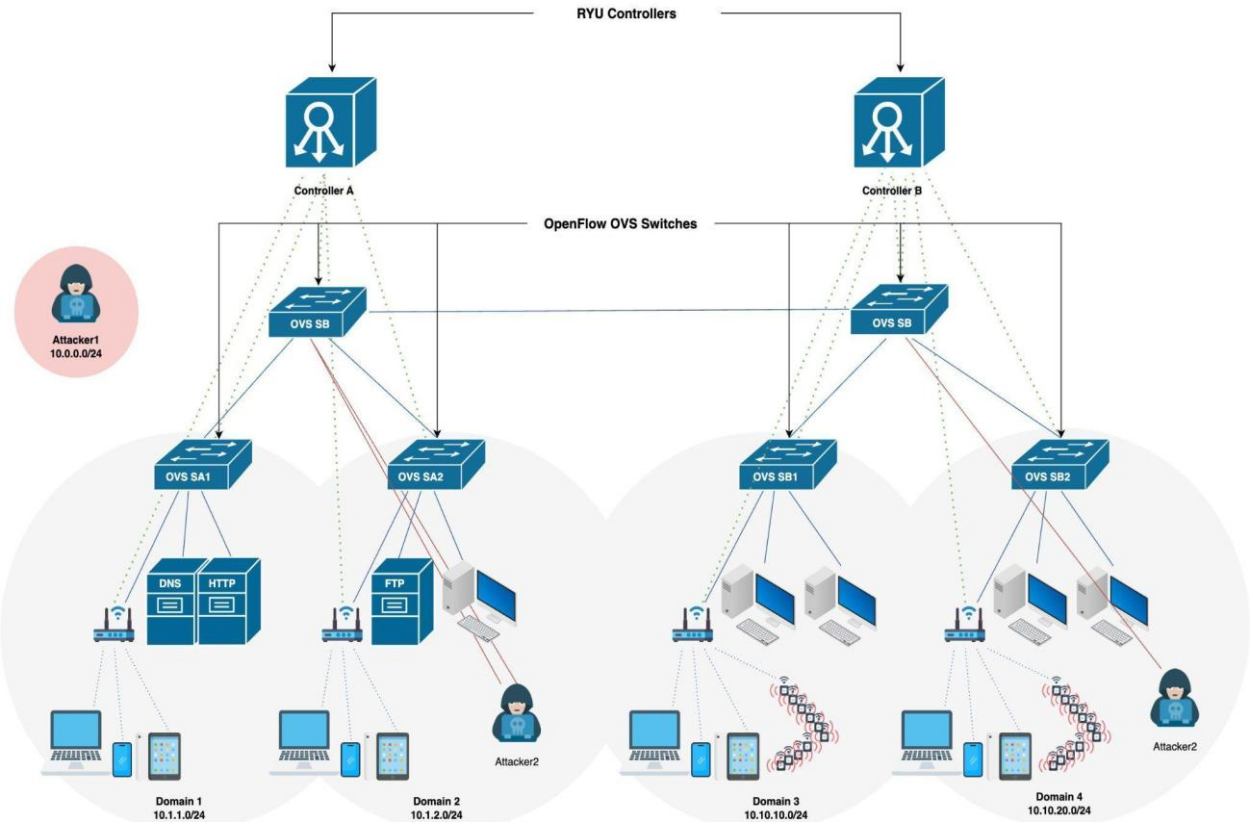

Fig. 3. Proposed network architecture.

4.2. Generation Methods

As an illustration, the attacker possesses the ability to produce harmful data flows to attack SDN controllers or potentially compromise communication channels connecting the SDN controller and OpenFlow switches. Once the flow of traffic is established, compromised individuals can be utilised to initiate subsequent attacks. Notably, SDN applications may exhibit many vulnerabilities, including but not limited to command injection, buffer overflow and SQL injection.

These vulnerabilities have the potential to generate chances for attacks, enabling the attacker to circumvent the authentication process and gain unauthorised access to the controller by executing a malicious script. If the attacker can obtain unauthorised entry to the controller, then they can initiate additional offensive actions, including the modification of flow rules, the execution of a DoS attack and the interception of data and control traffic for surveillance purposes. Table 2 lists the attack categories employed within the virtual environment.

Table 2.
Dataset with attack classes produced within a virtual environment.

Attack	Description of Activities
DoS	TCP-SYN, UDP, ICMP Flooding
DDoS	TCP-SYN, UDP, ICMP Flooding
Web Attack	XSS, SQL Injection
R2L	IMAP, Brute-Force, Mirai
Malware	Trojan, Botnet
Probe	Probe scan, Discover service, Vulnerability scan, Spoofing, Packet Sniffing, Ping Sweep
U2R	VSFTPD, Samba, Passive Exploitation

The entirety of the generated network traffic is gathered and examined via Wireshark. The data are subsequently labelled according to their profile and attack type, and its key features are extracted via CICFlowMeter, which was created by the Canadian Institute of Cyber Security and uses Java for network traffic flow generation from PCAP files. Fig. 4 illustrates the key stages involved in the traffic generation process.

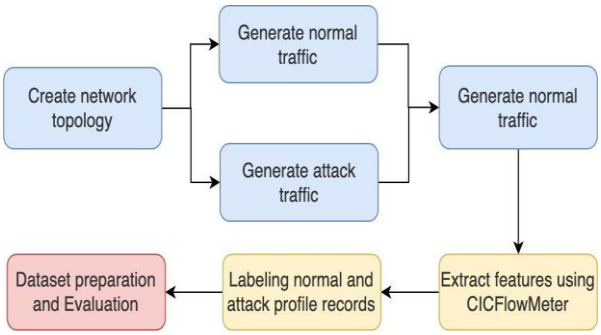

Fig. 4. Dataset generation process.

5. Results and Analysis

The dataset was partitioned into three distinct groups according to the traffic type and target machine. The first category comprises exclusively regular traffic; the second category comprises attack traffic specifically directed from the Metasploit server; and the third category comprises the attack on the IoT device. The overall number of 39,608,114 occurrences for regular and attack traffic. The regular traffic yields a cumulative count of 11,638,252, whereas the attack traffic has 27,969,892 occurrences. Table 3 shows the data records and their respective sizes. Wireshark is utilised to gather traffic flows in each class on the victim computer and interface of the SDN controller.

As mentioned, CICFlowMeter is utilised to obtain the flow features of the dataset, and a separate dataset is deployed for each attack type and combined to produce the final dataset. The primary motivation for utilising the CICFlowMeter is that none of the other tools specifically focus on time-based aspects [23]. Nevertheless, various applications are subjected to distinct time limitations. Consequently, the calculation of statistical time-related properties for flow traffic becomes incredibly important.

The generated flows are computed bidirectionally, with the flow path (forward or backward) determined by the first packet in the flow. CICFlowMeter generates over 80 metrics within a CSV file, including protocol, byte number, duration, packet number and others. In the process of labelling, various aspects are utilised, including information about the source and destination IP addresses. Table 4 shows some of the extracted features and corresponding details.

Table 3.
Total number of data records with their respective sises.

Group	Description	Instance number	.PCAP size
Normal Group	HTTPS, HTTP, FTP, DNS, mail, browsing, YouTube,.etc.	11,638,252	485 MB

Attack Group	DoS DDoS R2L (IMAP, Brute-Force Attack) Web-Based Attack	26,678,409	4.58 GB
	Botnet Probe (scan, Discover service, Vulnerability scan, Spoofing	765,522	97.82 MB
IoT Group	Mirai	525,961	286 MB

Table 4.
Some of the extracted features and corresponding details.

Feature	Description	Attack	Description	Attack	Description
Flow ID	Flow unique Identifier	Fwd IAT Mean	The average duration between two consecutive packets in the forward direction.	down/Up Ratio	Upload and download ratio
Source IP	IP address of the source host	Fwd IAT Std	Duration elapsed between the transmission of two packets in the forward direction.	Average Packet Size	The average size of packet
Destination IP	IP address of the destination host	Fwd IAT Total	Total time between two packets sent in the forward direction	Fwd Segment Size Avg	Average size observed in the forward direction
Flow, IAT Max,	The maximum duration between the transmission of two packets within a given flow.	PSH Flag Count	Amount of packets containing the PUSH flag.	Idle Max	The maximum duration of inactivity observed before the resumption of activity in a flow.
Flow, IAT Min,	Minimum interval duration between the transmission of two packets inside a given flow.	ACK Flag Count	Amount of packets containing the ACK flag.	Idle Std	The standard deviation of time duration of that flow remains idle before becoming active.
Fwd IAT Min	Minimum time between two packets sent in the forward direction	URG Flag Count	Amount of packets containing the URG flag.	Attack Type	Specify the Attack Type
Flow ID	Flow unique Identifier	Fwd IAT Mean	Average duration between two consecutive packets in the forward direction.	down/Up Ratio	Upload and download ratio
Source IP	IP address of the source host	Fwd IAT Std	Duration elapsed between the transmission of two packets in the forward direction.	Average Packet Size	The average size of packet

1.1.Evaluation

Widely recognised ML and DL techniques have been used to evaluate dataset quality. The first step is the preprocessing phase, which involves removing duplicated and nonvaluable values, encoding and scaling the dataset to mitigate significant variations in values and then utilising several algorithms, such as the following:

The logistic regression (LGR) model calculates the probability of an event on a dataset of independent variables, such as voting or not voting. The dependent variable is confined between 0 and 1 because the outcome is probabilistic. These formulas represent this logistic function, often known as log odds or the natural logarithm of odds:

$$h_\Theta(x) = g(\Theta^T x) = \frac{1}{1 + e^{-\theta^T x}} \quad \ldots(1)$$

$$where\ g(z) = \frac{1}{1 + e^{-z}}$$

K-nearest neighbours (KNNs) are nonparametric, supervised learning classifiers that classify or predict data point groupings on the basis of nearness. KNNs can be used for regression or classification. However, KNNs are usually used for classification, assuming that similar points are nearby.

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i| \quad \ldots(2)$$

Naive Bayes classifiers use Bayes' theorem for classification. The algorithms in this family have a similar premise. Each pair of classed features is independent.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \quad \ldots(3)$$

The most powerful and popular categorisation and prediction technique is decision tree classifiers (DTCs). A decision tree (DT) is a flowchart-like tree structure with internal nodes representing attribute tests, branches representing test outcomes and leaf nodes (terminal nodes) representing class labels.

$$Gini(P) = \sum_{i=1}^{n} p_i(1 - p_i) \quad \ldots(4)$$

$$1 - \sum_{i=1}^{n} (p_i)^2$$

$$where\ P = (p_1, \ldots, p_n)$$

In supervised learning algorithm RFs, its 'forest' is a DT ensemble trained with a 'bagging' method. The bagging method combines learning models to improve the results. RFs are useful for classification and regression, which are common in ML systems; it can also resist DT overfitting.

$$Gini(P) = 1 - \sum_{i=1}^{n} (P_i)^2 \quad \ldots(5)$$

The deep neural network (DNN) model utilised in this work is shown in Table 5.

Table 5.
Neural network training model.

Algorithm Family	Layers	Neuron
DNN	5 Dense	250,150,64,32,16
Activation function	Relu, Sigmoid	
Loss function	Categorical cross-entropy	
Optimizer	Adam	
Batch-size	64	
Epochs	10	

The studies were conducted via the Python programming language, utilising a range of packages, including sklearn, Keras and TensorFlow, and the dataset was evaluated on the basis of the following metrics: accuracy, F1 score, precision and recall, which are described via the following equations:

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Instance} \quad \ldots(6)$$

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad \ldots(7)$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad \ldots(8)$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad \ldots(9)$$

The metrics of the DTC and RF classifier (RFC) algorithms yield satisfactory results across all the other models. In general, the aforementioned algorithms achieved a high level of success in identifying the majority of attacks. Figs. 5 and 6 show the results related to the diverse attack classes and model performance.

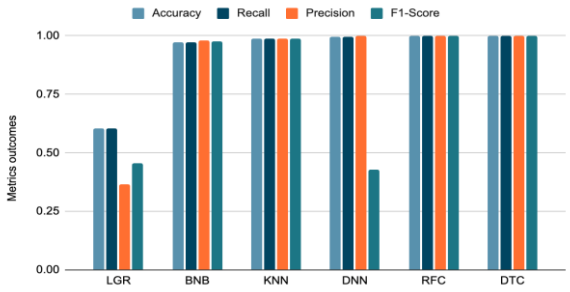

Fig. 5. Performance metrics correspond to the model.

The combined metrics exhibit a high level of performance across all classes, with a notable exception being the U2R category, which demonstrates subpar performance metrics. This phenomenon can be attributed to other categories often exhibiting greater dissimilarity than typical traffic patterns do. However, the U2R attack class significantly resembles regular data traffic. Furthermore, the U2R flow records are smaller than the regular flow records within the same dataset. The system had a high degree of success in identifying the majority of attacks, but it exhibited relatively low performance in detecting U2R attacks. Another noteworthy observation pertains to the potential masking of subpar performance on less common attack classes by the favourable outcomes achieved on the combined dataset. This phenomenon arises from the predominance of samples associated with DoS/DDoS and probing attacks. Moreover, the duration of training is directly proportional to the number of data records. In other words, as the size of the records increases, the training time also increases.

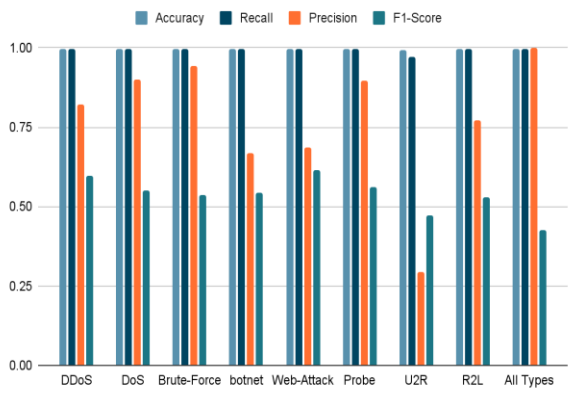

Fig. 6. Performance metrics correspond to the attack class.

1.2. Limitations

The historical record of SDN attacks remains undisclosed. Hence, in this work, possible weaknesses were identified that may be exploited by taking the attacker's viewpoint.

• The testbed was constructed utilising a pair of SDN controllers that functioned equally with EQUAL roles. The categories of security analysis functionalities for other controllers are disregarded. However, controllers may exhibit varying security modelling [24,25], which would require the implementation of different countermeasures.

• The network topology may be constructed via physical devices to increase the availability of intrinsic information in SDNs. A series of experiments were conducted to evaluate different attack scenarios and analyse their effects on the layers of an SDN architecture.

• One of the primary constraints of the dataset under consideration is a significant imbalance in class distribution. This issue has the potential to introduce bias in the IDS against the majority class, resulting in elevated false alarm rates and reduced evaluation accuracy. Nevertheless, several methodologies exist for addressing the issue of imbalanced samples, as shown in prior research [25]–[29]. Two distinct approaches can be employed: (a) the higher classes can be subdivided to generate a greater number of classes, and (b) the merging of multiple minority classes that exhibit similar traits can result in the creation of a singular new class. Consequently, the problem of imbalance can be mitigated, leading to a notable increase in the prevalence ratio.

2. Conclusion

This work examined the lack of an openly accessible dataset obtained from SDN-based IoT networks and produced an extensive dataset consisting of 86 features extracted from approximately 40 million records obtained from simulated SDN-based IoT networks captured within two flow profiles, representing normal behaviour and 15 different attack types in two scenarios, interdomain and intradomain. In addition, the dataset was evaluated via several ML and DL methodologies, such as DTC, RFC, DNN, KNN, Bernoulli naive Bayes and LGR. The outcome reveals high accuracy in successfully recognising the majority of attacks when the DTC and RFC models are employed and a low accuracy level in detecting the U2R attack category. Nevertheless, an SDN is more vulnerable to malicious network traffic than traditional networking configurations are. Within the traditional network architecture, an attack can affect only a specific segment of the network, which is usually limited to the components, without causing complete network failure. However, in the context of SDN, the compromised users and switches have the ability to overpower the SDN controller, leading to harmful consequences for the entire network.

References

[1] W. Meng, “Intrusion detection in the era of IoT: building trust via traffic filtering and sampling”, IEEE Comput. 51 (7) (2018) 36–43. IEEE, DOI: 10.1109/MC.2018.3011034.

[2] [2] F.Y. Okay, S. Ozdemir, “Routing in fog-enabled IoT platforms: a survey and an SDN-based solution”, IEEE Internet Things J. 5 (6) (2018) pp. 4871–4889.DOI: 10.1109/JIOT.2018.2882781.

[3] W. Li, W. Meng, Z. Liu, M.H. Au, “Towards blockchain-based software-defined networking: security challenges and solutions”, IEICE Trans. Info Syst. E103CD (2) (2020), pp. 196–203. DOI: 10.1587/transinf.2019INI0002.

[4] T. Das, V. Sridharan, M. Gurusamy, “A survey on controller placement in SDN”, IEEE Commun. SurvTutorials 22 (1) (2020), pp. 472–503. DOI: 10.1109/COMST.2019.2935453.

[5] SDN-Based Intrusion Detection System: A Survey, Ying-Jun Zhang, Hong Li, Zhong-Ru Yang, and Jun-Ping Du,IEEE Access, Volume 6, 2018.

[6] N. KoronIoTis, N. Moustafa, E. Sitnikova, and B. Turnbull, “Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: BoT-IoT Dataset”, Future Generation Computer Systems, vol. 100, pp. 779–796, 2019, DOI: 10.1016/j.future.2019.05.041.

[7] A. Kaan Sarica and P. Angin, “A Novel SDN Dataset for Intrusion Detection in IoT Networks”, 2020 16th International Conference on Network and Service Management (CNSM), Izmir, Turkey, 2020, pp. 1-5, doi: 10.23919/CNSM50824.2020.9269042.

[8] Mahmoud Dais ELsayed, Nhien-An Le-khac, and Anca D. Jurcut, InSDN: “A Novel SDN Intrusion Dataset”, IEEE 2020. DOI: 10.1109/ACCESS.2020.3022633.

[9] Ahuja, N., Singal, G., Mukhopadhyay, D., & Kumar, N. (2021). “Automated DDOS attack detection in software defined networking. Journal of Network and Computer Applications”, 187, 103108. doi: 10.1016/j.jnca.2021.103108.

[10] A. R. Gad, A. A. Nashat and T. M. Barkat, "Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on ToN-IoT Dataset," in IEEE Access, vol. 9, pp. 142206-142217, 2021, doi: 10.1109/ACCESS.2021.3120626.

[11] A. Divekar, M. Parekh, V. Savla, R. Mishra, and M. Shirole, ‘‘Benchmark- ing datasets for anomaly-based network intrusion detection: KDD CUP 99 alternatives,’’ in Proc. IEEE 3rd Int. Conf. Comput., Commun. Secur. (ICCCS), Oct. 2018, pp. 1–8. DOI: 10.1109/CCCS.2018.8586840.

[12] L. Bontemps, V. Cao, J. McDermott, and N.-A. Le-Khac, ‘‘Collective anomaly detection based on long short-term memory recurrent neural networks,’’ in Future Data and Security Engineering FDSE (Lecture Notes in Computer Science), vol. 10018, T. Dang, R. Wagner, J. Küng, N. Thoai, M. Takizawa, and E. Neuhold, Eds. Cham, Switzerland: Springer, 2016. DOI: 10.1007/978-3-319-48057-2_9.

[13] M. Tavallaee, E. Bagheri, W. Lu, and A.A. Ghorbani, ‘‘A Detailed Analysis of the KDD CUP 99 data set,’’ in Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl., Jul. 2009, pp. 1–6. DOI: 10.1109/CISDA.2009.5356528.

[14] for Applied Internet Data Analysis (CAIDA), T. C. (2016). Caida anonymized internet traces 2016 dataset. DOI: 10.5220/0006639801080116.

[15] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, ‘‘Toward developing a systematic approach to generate benchmark datasets for intrusion detection,’’ Comput. Secur., vol. 31, no. 3, pp. 357–374, May 2012. Doi: 10.1016/j.cose.2011.12.012.

[16] Symantec. Internet Security Threat Report, 2018. [Online]. Available: https://symantec-enterprise-blogs.security.com/

[17] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis et al.,”Understanding the Mirai Botnet”, in 26th { USENIX } Security Symposium ({ USENIX } Security 17), 2017, pp. 1093–1110.doi: 10.1016/j.fsidi.2020.300926.

[18] N. Sharma and S. Mukherjee,”A novel multi-classifier layered approach to improve minority attack detection in IDS”, Procedia Technol., vol. 6, pp. 913–921, Jan. 2012. Doi: 10.1016/j.protcy.2012.10.111.

[19] K. Benzekki, A. El Fergougui, and A. Elbelrhiti Elalaoui, “Software-defined networking (SDN): A survey”, Secur. Commun. Netw., vol. 9, no. 18, pp. 5803–5833, Dec. 2016. Doi: 10.1002/sec.1737.

[20] A. Dawoud, S. Shahristani, and C. Raun, “Software-defined network security: Breaks and obstacles”, in Networks of the Future:

Architectures, Technologies, and Implementations. Boca Raton, FL, USA: CRC Press, 2017, pp. 89–100, DOI:10.1201/9781315155517-5.

[21] M. Liyanage, A. Braeken, A. D. Jurcut, M. Ylianttila, and A. Gurtov, "Secure communication channel architecture for software defined mobile networks", Comput. Netw., vol. 114, pp. 32–50, Feb. 2017. Doi: 10.1016/j.comnet.2017.01.007.

[22] The IoT attack surface: Threats and security solutions 2019. [Online]. Available: https://www.trendmicro.com/vinfo/gb/security/news/internet-of-things/the-IoT-attack-surface-threats-and-security-solutions.

[23] A. H. Lashkari, G. Draper-Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," in Proc. ICISSP, 2017, pp. 253–262. DOI: 10.5220/0006105602530262.

[24] R. Khondoker, A. Zaalouk, R. Marx, and K. Bayrou," Feature-basedcom- parison and selection of software defined networking (SDN) controllers", in Proc. World Congr. Comput. Appl. Inf. Syst. (WCCAIS), Jan. 2014, pp. 1–7. DOI: 10.1109/WCCAIS.2014.6916572.

[25] K. Phemius, M. Bouet, and J. Leguay, "DISCO: Distributed multi-domain SDN controllers", in Proc. IEEE Netw. Oper. Manage. Symp. (NOMS), May 2014, pp. 1–4. DOI: 10.1109/NOMS.2014.6838330.

[26] C. Mera and J. W. Branch,"A survey on class imbalance learning on automatic visual inspection", IEEE Latin Amer. Trans., vol. 12, no. 4, pp. 657–667, Jul. 2014. DOI: 10.1109/TLA.2014.6868867.

[27] S. Wang and X. Yao, "Multiclass imbalance problems: Analysis and potential solutions", IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 42, no. 4, pp. 1119–1130, Aug. 2012. DOI: 10.1109/TSMCB.2012.2187280.

[28] R. Longadge and S. Dongre, "Class imbalance problem in data mining review", 2013, arXiv:1305.1707. [Online]. Available: http://arxiv.org/abs/1305.1707.

[29] S. M. Abd Elrahman and A. Abraham, "A review of class imbalance problem", J. Netw. Innov. Comput., vol. 1, no. 2013, pp. 332–340, 2013. Doi: 10.1016/j.jmse.2022.06.002.

مجموعة بيانات اكتشاف التطفل في شبكة إنترنت الأشياء المُعرّفة بالبرمجيات: بين وداخل النطاقات

هبه ضرار[1*]، علي حسين حمد[2]

[1،2] *قسم هندسة المعلومات والاتصالات، كلية الهندسة الخوارزمي، جامعة بغداد، بغداد، العراق*

*البريد الالكتروني: heba.d@kecbu.uobaghdad.edu.iq

المستخلص

يتم نشر الشبكات المعرفة بالبرمجيات (SDN) على نطاق واسع في العديد من تكوينات الشبكة. ومع ذلك، فإن تطوير التكنولوجيا الجديدة يمثل العديد من نقاط الضعف والمخاطر التي لا تزال تشكل تحديات أمام الشركات المصنعة في معالجتها. تتمثل إحدى العوائق الرئيسية التي تمت مواجهتها في نشر نظام كشف التسلل (IDS) في عدم وجود مجموعة بيانات يمكن الوصول إليها بشكل مفتوح، خاصة التي يتم الحصول عليها من شبكات SDN، وشبكات إنترنت الأشياء المستندة إلى SDN. ينتج هذا العمل مجموعة بيانات شاملة لتقييم فعالية IDS القائمة على الشذوذ في اكتشاف الهجمات داخل المجال. تشتمل مجموعة البيانات على ست وثمانين ميزة مستخرجة ما يقرب من أربعين مليون سجل تم الحصول عليها من محاكاة شبكات إنترنت الأشياء المستندة إلى SDN، والتي تم التقاطها ضمن ملفي تعريف تدفق يمثلان أنواعًا عادية وخمسة عشر نوعًا مختلفًا من الهجمات. بالإضافة إلى ذلك، نعرض التقييم الذي يستخدم ستة أساليب للتعلم الآلي، والتعلم العميق المستخدمة على نطاق واسع لمعرفات كشف المعلوماتDTC :، RFC، DNN، KNN، BNB، LGR.