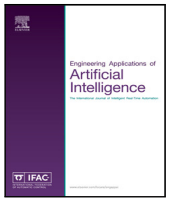
Engineering Applications of Artificial Intelligence 123 (2023) 106432

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks

Jalal Bhayo [a], Syed Attique Shah [b,*], Sufian Hameed [a], Awais Ahmed [c], Jamal Nasir [a], Dirk Draheim [d]

[a] Department of Computer Science, National University of Computer and Emerging Sciences (NUCES-FAST), 75160, Karachi, Pakistan
[b] School of Computing and Digital Technology, Birmingham City University, STEAMhouse, B47RQ, Birmingham, United Kingdom
[c] University of Electronic Science and Technology of China (UESTC), 610056, Sichuan, China
[d] Information Systems Group, Tallinn University of Technology, 12618 Tallinn, Estonia

ARTICLE INFO

Keywords:
Internet of things (IoT)
DDoS attacks
Software defined networks (SDN)
SDN-WISE
Intrusion detection system (IDS)
Machine learning

ABSTRACT

The Internet of Things (IoT) is a complex and diverse network consisting of resource-constrained sensors/devices/things that are vulnerable to various security threats, particularly Distributed Denial of Services (DDoS) attacks. Recently, the integration of Software Defined Networking (SDN) with IoT has emerged as a promising approach for improving security and access control mechanisms. However, DDoS attacks continue to pose a significant threat to IoT networks, as they can be executed through botnet or zombie attacks. Machine learning-based security frameworks offer a viable solution to scrutinize the behavior of IoT devices and compile a profile that enables the decision-making process to maintain the integrity of the IoT environment. In this paper, we present a machine learning-based approach to detect DDoS attacks in an SDN-WISE IoT controller. We have integrated a machine learning-based detection module into the controller and set up a testbed environment to simulate DDoS attack traffic generation. The traffic is captured by a logging mechanism added to the SDN-WISE controller, which writes network logs into a log file that is pre-processed and converted into a dataset. The machine learning DDoS detection module, integrated into the SDN-WISE controller, uses Naive Bayes (NB), Decision Tree (DT), and Support Vector Machine (SVM) algorithms to classify SDN-IoT network packets. We evaluate the performance of the proposed framework using different traffic simulation scenarios and compare the results generated by the machine learning DDoS detection module. The proposed framework achieved an accuracy rate of 97.4%, 96.1%, and 98.1% for NB, SVM, and DT, respectively. The attack detection module takes up to 30% usage of memory and CPU, and it saves about 70% memory while keeping the CPU free up to 70% to process the SD-IoT network traffic with an average throughput of 48 packets per second, achieving an accuracy of 97.2%. Our experimental results demonstrate the superiority of the proposed framework in detecting DDoS attacks in an SDN-WISE IoT environment. The proposed approach can be used to enhance the security of IoT networks and mitigate the risk of DDoS attacks.

1. Introduction

With the advancing Internet of Things (IoT) innovations, there is exponential growth in the inclusion of various kinds of "things"/ devices/ sensors/ objects into the Internet. These resource-constrained "things" can be an easy target for attackers to launch various types of attacks, including Denial-of-Service (DoS), Man-In-The-Middle (MITM), and malware attacks. In the last decade, the escalating usage of heterogeneous IoT devices has extended challenges related to security, performance, accessibility, and scalability. With this growing IoT dilemma, more connected devices mean more assault vectors and more conceivable outcomes for attackers to target (Wang et al., 2020; Ali et al., 2020). Therefore, there is a high demand to rapidly address these rising security concerns, or IoT applications will face inevitable threats. However, due to the heterogeneous nature of IoT devices, it is challenging to deploy security mechanisms (Yaqoob et al., 2019).

The number of Internet-connected devices in the IoT environment is expected to exceed 100 billion by the end of 2025 (Taylor et al., 2015). The proliferation of heterogeneous devices and objects in IoT environments has uncovered deficiencies in security protocols and mechanisms within IoT frameworks (Chernyshev et al., 2017). These security loopholes make IoT devices easy targets, and more ambitious attacks on IoT devices have been long predicted. For example, there are reported incidents of seizing access control of various IoT

* Corresponding author.
E-mail addresses: jalal.bhayo@nu.edu.pk (J. Bhayo), syed.shah2@bcu.ac.uk (S.A. Shah), sufian.hameed@nu.edu.pk (S. Hameed), 202014080105@std.uestc.edu.cn (A. Ahmed), dirk.draheim@taltech.ee (D. Draheim).

https://doi.org/10.1016/j.engappai.2023.106432
Received 8 July 2022; Received in revised form 9 April 2023; Accepted 5 May 2023
Available online 23 May 2023
0952-1976/© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

devices in the home or mechanical ecosystems to exfiltrate sensitive data of individuals. Moreover, IoT devices are dynamically recruited into botnet armed forces for multi-stage Distributed Denial of Services (DDoS) attacks (Hallman et al., 2017). Today, the definition of a DDoS attack gets more and more complicated as cybercriminals utilize combinations of high-volume attacks. It is becoming more challenging to detect infiltration that targets applications and existing network security infrastructures such as firewalls and intrusion prevention systems (IPS).

Attacks on IoT devices are increasing, as according to the report of a security vendor, 100 million IoT attacks have been detected in the first half of 2019. Cybersecurity and anti-virus provider Kaspersky spotted 105 million attacks from 276 thousand unique IP addresses in the first six months of 2019 (Over 100 Million, 2021). According to the same report, the most common malware types have been Mirai and Nyadrop and the majority of IoT devices have been affected in China (30%), Brazil (19%) and Egypt (12%) (Over 100 Million, 2021). Cicero, a network security company, has noticed a dramatic increase in the frequency of attack attempts against its customers (Corero, 2020). Dyn, a provider of Domain Name System (DNS) services, was attacked in October 2016 by two large and complex DDoS attacks against its DNS infrastructure (DDoS attack, 2016). Due to the attack, large numbers of Internet platforms and services – including well-known brands such as Spotify, Netflix, Reddit, and Twitter – experienced significant service outages. Another report indicates that DDoS attacks during the global pandemic year 2020 significantly increased in number as compared to the previous years (DDoS Attacks Spiked, 2021). The report also reveals different projections of DDoS attacks that depict more complex and high-frequency attacks as compared to prior years.

The Software-Defined Networking (SDN) paradigm opens tremendous opportunities to manage and secure IoT. SDN aims at creating network architectures that are more agile, flexible, and smart, making them different from the traditional networking architecture (Bhayo et al., 2022). An IoT network is different from a traditional network at various levels. Characteristic differences are in processing power, scalability, energy consumption, etc., which, however, also creates a significant management challenge for IoT networks. SDN has emerged as a promising network model with exponential growth in network management and configuration complexity (Siddiqui et al., 2022; Khalid et al., 2023). SDN seeks to effectively turn network design and operations to become more agile and to efficiently improve network functions (Hameed et al., 2021; Bawany and Shamsi, 2019). The SDN paradigm has unique features that provide a dynamic and programmable network with centralized management, where the network is abstracted from the upper-layer applications. SDN controller provides centralized network intelligence and helps network administrators to monitor, protect, and optimize network resources dynamically and programmatically configure network traffic patterns (Ahmad et al., 2015). The SDN controller cannot only manage the IoT heterogeneous system but can also monitor the incoming and outgoing traffic.

Recently, machine learning (ML) is widely used to aid in various aspects related to instruction detection and other security and threat analysis. ML supports diverse network traffic-generated datasets and a number of data features that can be helpful to get better insights once properly analyzed (Di Mauro et al., 2021). In integration with ML, the Software-Defined Internet of Things (SD-IoT) can offer various solutions to tackle the security challenges faced by IoT devices. Cui et al. (2019) utilized a support vector machine (SVM) algorithm to train the attack detection module in the SDN for classifying DDoS attack patterns. They highlight the importance of cognitive-inspired computing with entropy technique using entropy values as a feature vector. The implemented mechanism quickly detects and mitigates DDoS attacks and therefore restores back to normal communication in time. In this study, we use naive Bayes and Decision Tree (DT) along with an SVM classifier to get more improved and competitive results. In order to detect huge amounts of SDN malicious traffic and DDoS attacks, the sFlow and adaptive polling-based samples with Snort IDS were proposed on the data plane, along with the deep-learning Stacked Auto-encoders (SAE) on the control plane (Ujjan et al., 2020). The active mode of the proposed framework is significantly more effective than the passive mode with respect to preprocessing and DDoS detection in the traffic selector method. The framework seems to be useful in combination with conventional security mechanisms to optimize the results and detection time. However, the SAD-F framework is based on a traditional network and suffers from dynamic reprogramming and central management control. Our framework is also based on ML algorithms, but aided by SDN, i.e., it provides a unique solution to SD-IoT networks. Hameed and Ali (2018) propose the HADEC framework for live flooding-based DDoS attack detection using MapReduce and Hadoop. HADEC takes less than 5 min to process 1 GB of the log file having 15.83 GB generated live traffic. According to the results, HADEC takes low-time attack detection, near real-time, but it is more CPU intensive, and the capturing phase consumes 77% of the overall detection time. Instead, our framework is based on SD-IoT, which provides dynamic and programmable features. Additionally, with the help of supervised machine learning classifiers, it efficiently detects DDoS attacks in much less time. Similarly, Yin et al. (2018) proposed an SD-IoT framework for security against DDoS attacks. The framework consists of DDoS attack detection and mitigation algorithms based on the cosine similarity vector of incoming packet messages at the boundary of SD-IoT switches to determine the DDoS attack based on the threshold of the cosine similarity vector. The algorithms work only on packet-in messages; while, our proposed architecture focuses on a machine learning classifier that uses distinct network attributes to classify the DDoS attack traffic from the SD-IoT network traffic flow. In this regard, we investigated different IDS applications for IoT networks including a time-efficient IDS (Zhang et al., 2020) which is based on SD-IoT, Counter-based DDoS attack detection IDS (Bhayo et al., 2020), and other machine learning–based IDS for IoT (Verma and Ranga, 2020). The main objective of these IDS is to detect attacks in IoT networks and to ensure security in the IoT domain.

SDN-IoT could face challenges in terms of performance, interoperability, scalability, dependability, and security (Xie et al., 2018). Furthermore, SDN poses a great challenge in network management (Ghaffar et al., 2021). Despite of rapid increase in research in the area of applications SDN-IoT using machine learning, it is facing multiple open and on-demand challenges including but not limited to (a) Unavailability of quality datasets for appropriate training, (b) Unawareness of distributed and scalable multi-controller platforms, (c) Continuous improvement of network security, and (d) Incremental deployment of SDN. The fusion of SDN and IoT brings several advantages, including intelligent routing, efficient data processing and analysis, centralized application and resource management, and dynamic network reconfiguration. These benefits are derived from SDN's programmable and centralized network infrastructure, which simplifies network management, enhances flexibility, and enables the implementation of innovative network services and applications in a scalable and cost-effective manner.

Developing and improving ML and SDIoT-based frameworks that address the potential security challenges associated with IoT devices is an overlooked aspect in the current literature. Our proposed framework is based on software-defined IoT and includes dynamic and re-programmable characteristics that enable the SD-IoT network to perform security services. The framework detects DDoS attacks with high accuracy and detection rates by incorporating machine learning methods. Our suggested framework for the SD-IoT network is based on machine learning and consists of three distinct independent components: (1) A data-plane module made up of Sensor OpenFlow Switches (SOFS) and IoT devices; (2) An IoT controller module comprised of adjusted SDN-WISE that manages and controls the SD-IoT network; and (3) A machine learning-based DDoS attack detection module consisting of various supervised learning-based classifiers for classifying malicious and legitimate traffic flow.

The proposed framework provides an innovative solution to one of the biggest challenges faced by SD-IoT networks i.e., DDoS attacks. Industries such as healthcare, finance, and transportation are heavily reliant on IoT networks to transmit sensitive data. Any security breach in these networks could have devastating consequences. With the integration of SDN and IoT, the proposed framework provides a promising solution for better security and access control mechanisms. By implementing a machine learning-based security framework to scrutinize IoT devices' movements and compile a profile, the framework can detect abnormal traffic and prevent DDoS attacks in SD-IoT networks. With the rapid development of various IoT domains such as Smart Cities, Smart vehicles, etc., security application development for IoT networks is one of the essential parts of these domains (Siddiqui et al., 2023). The proposed framework's applications extend beyond industries to smart cities and cloud computing data centers. With the rapid increase in IoT devices used in smart cities, the risk of cyber-attacks and DDoS attacks are a growing concern. By deploying the proposed framework, administrators can prevent service disruptions and improve the reliability of smart city applications.

The main contributions of this research are given as follows:

- A novel framework designed for an implemented system based on machine learning and Software-defined IoT with dynamic and re-programmable features is presented for the effective and timely detection of DDoS attacks.
- The ML-based module runs on top of the SDN-WISE controller and consists of different supervised learning classifiers such as Naive Bayes, SVMs, and Decision Trees (DT)s to efficiently determine the malicious traffic flow. DT has an accuracy ratio of 98.1%, whereas Naive Bayes and SVM have an accuracy rate of 97.4% and 96.1%, respectively.
- The proposed framework thoroughly analyzes the given parameters such as IoT nodes, attack nodes, payload size, and packet frequency with the selected classifiers to measure the performance of an SD-IoT network through outcome factors such as CPU usage, attack detection time, and memory usage. According to the results, the early detection of malicious traffic within the SD IoT network is a major advantage in the prevention of high levels of exploitations and the isolation of IoT devices from malicious nodes.

Table 1 presents the most used acronyms in this paper. The rest of the paper has the following structure. Section 2 presents the related work. Section 3 discusses the proposed ML- and SD-IoT-based framework and explains each component of the framework in detail. Section 4 discusses the testbed experimental setup and explains the results gathered during the experiments and finally Section 5 presents the conclusion.

Table 1
List of most common abbreviations.

Acronym	Full term(s)
AI	Artificial Intelligence
ANN	Artificial Neural Network
BNE	Bayesian–Nash Equilibrium
CAIDA	Center for Applied Internet Data Analysis
C-DAD	Counter-based DDoS detection
DT	Decision Tree
DNN	Deep Neural Network
DoS	Denial-of-Service
DDoS	Distributed Denial of Services
DNS	Domain Name System
IoT	Internet of Things
IDPS	Intrusion Detection and Prevention System
k-NN	k-Nearest Neighbor
ML	Machine Learning
MITM	Man-in-the-Middle
NB	Naive Bayes
RF	Random Forest
SOFS	Sensor OpenFlow Switches
SDN	Software Defined Networking
SDN-WISE	Software Defined Networking solution for Wireless Sensor Networks
SD-IoT	Software-Defined Internet of Things
SVM	Support Vector Machine
WSN	Wireless Sensor Networks

2. Related work

Several studies in the existing literature have analyzed DDoS attacks and contributed various protection mechanisms (Tayyab et al., 2020; Snehi and Bhandari, 2021; Alamri and Thayananthan, 2020). The most broadly utilized defense methods are identifying and mitigating DDoS attacks, traffic separation, and trace-back the DDoS source. DDoS detection solutions are effectively separating typical streams of activity from unusual streams of activity. Traffic separation solutions obstruct substantial movement, while trace-back mechanisms must be compelling under sponsored activity performed for the most part after the assault. A large portion of current DDoS identification systems has constrained achievements considering the accompanying difficulties: (a) the attack frequently uses legit requests to overload the target itself, making it difficult to distinguish an attack movement from normal activity, (b) quick ongoing recognition is troublesome due to the enormous measure of information associated with the current network (Suresh and Anitha, 2011).

Two critical and challenging research concerns in identifying DDoS attacks are as follows:

- Distinguishing a genuine and sufficient selection of features that can be used to construct efficient models for differentiating DDoS attacks from normal traffic.
- Assessing the viability of the various machine-learning approaches employed in the discovery process.

Statistical approaches can be used to detect suspicious patterns in resource utilization in response to DDoS attacks. The issue with statistics-based identification is that it is not conceivable to discover the typical network packet distribution. Or maybe, it must be reproduced as a uniform distribution (Lee et al., 2008). A few strategies which apply data mining methods can acquire a high success rate in recognizing the attacks. In any case, these techniques generally cannot be utilized as a part of real-time computing (Xu et al., 2007). One advantage of clustering over statistical methods is that they are not dependent on any prior knowledge about the data distribution. Numerous factors can be utilized to recognize common network patterns. However, obtaining fundamental characteristics from a massive network is critical for modeling network behaviors that are distinct from normal traffic.

Numerous studies have been conducted on the problem of feature extraction. For example, Chhabra et al. (2013) selected eight relative values as features independent of the network stream. Haddadi et al. (2010) suggest and investigate recognizable evidence of successful network features for attack detection testing, applying the principal component analysis (PCA) technique to determine an optimal set of capabilities. Software-Defined Networking (2020) examined the application of multivariate relationship analysis to DDoS discovery and developed a strategy for recognizing flooding attacks using co-variance analysis. They used the majority of the flag bits in the TCP header's flag field as highlights in the co-variance investigation presentation. The researchers demonstrated the effectiveness of the proposed technique in detecting SYN flooding attacks, a critical sort of DDoS attack, however, the technique faces a severe hurdle because there is no guarantee that the six flags are substantial or sufficient attributes for reliably distinguishing all sorts of DDoS attacks.

A variety of statistics and machine learning techniques can be employed to detect the unusual changes in resource use associated with DDoS attacks. Both techniques, however, have their limitations. For example, one obvious limitation of statistics-based detection is the inability to determine the usual network packet circulation. This issue can be resolved by employing clustering methodology to construct the standard examples, as one of the advantages of clustering tactics

over measurement procedures is that they are not dependent on any previously known information transmission. While machine learning algorithms, which are frequently derived from the overlapping field of information mining, have been shown to be quite accurate at identifying DDoS attacks, they also have their own limits. For example, these systems demand a significant learning period, and as a result, these techniques cannot be used progressively at the moment. Regardless of these constraints, solutions to the DDoS recognition problem will emerge from either or both of these domains and major research effort is being directed in this direction. Lee et al. (2008), for example, used a multiclass SVM characterization model to detect DDoS threats. In the solution proposed by Xu et al. (2007), a collection of new features was also presented, including the establishment of relative values as a critical component of an extended arrangement of discovery data. Additionally, they presented another method of detecting DDoS attacks through the use of attack force. In Ahmed et al. (2020), Ahmed et al. presented a scalable Spark-based live DDoS detection framework, termed SAD-F, which is capable of analyzing potential DDoS attacks without any time delays, as the framework's performance has been tested against both live and passive traffic. SAD-F first captures live netflow traffic by using (Wireshark) live traffic capturing feature, then preprocesses it to extract required information, and finally uses ML-Spark algorithms to run detection algorithms for DDoS flooding attacks. SAD-F tackles the difficult problems of the traditional approach in terms of scalability, memory inefficiencies, and processes by parallel data processing with better efficiency and low latency. Bhayo et al. (2020) explored various research gaps and security challenges associated with the IoT and proposed a solution for counter-based DDoS detection (C-DAD) in SD-IoT networks. However, this architecture is built on a counter-based approach, whereas our research relies on machine learning algorithms that efficiently detect DDoS attacks against trained malicious patterns.

In Suresh and Anitha (2011), the authors introduced a new probabilistic packet inspection (PPM) model called TTL-based PPM plot, in which each bundle is separated with a probability inversely proportional to the separation traversed by the parcel up to this point, enabling a casualty source to track back the attack source. Nguyen and Choi (2010) have developed an Anti-DDoS structure based on k-NN (k-nearest neighbor) classification for proactively identifying DDoS attacks. They used the k-NN approach to categorize the system's state during each DDoS attack session. While the k-NN strategy is superior for assault discovery, it is computationally expensive for continuous use as the number of concurrent operations increases. Eskin et al. (2002) performed anomaly detection using an SVM classifier in which the feature space is mapped into another component space. Similarly, Yuan and Mills (2005), the author catches the traffic pattern of a DDoS attack using cross-relationship analysis. Nagtilak et al. (2020) propose to enhance the DDoS attack detection model based on deep learning for the IoT system. The detection model detects the attack in less time and provides better future extraction and good performance than conventional algorithms. The model takes advantage of deep learning to train massive generated data and efficiently detect DDoS attacks in IoT systems.

Du and Wang (2019) proposed a honeypot strategy for DDoS attacks in the industrial Internet of things using SDN. SDN provides dynamic protection through a honeypot strategy to efficiently control the malicious attacker. They also propose a pseudo-honeypot game (PHG) strategy that protects from anti-honeypot-based attacks and proves several Bayesian–Nash Equilibrium (BNE) groups in the PHG strategy. This strategy-based method improves energy consumption and IoT security. Idhammad et al. (2018) present a semi-supervised Machine learning-based approach to detect DDoS attacks. The author uses co-clustering, information gain ratio, network entropy estimation, and extra trees classifier to classify DDoS traffic accurately. The supervised algorithms are used to reduce the false-positive rates and unsupervised models for classifying malicious traffic. Different experiments have been conducted to benchmark the datasets, including NSL-KDD, UNSW-NB15, and UNB ISCX 12, to get accuracy with satisfactory false-positive rates. However, this research is based on a traditional network, while our research is based on a supervised machine learning approach using SD-IoT for IoT networks for getting more efficient performance.

In da Costa et al. (2019), Kelton et al. investigate the contemporary techniques used for intrusion detection based on machine learning for the IoT. The research reveals valuable techniques for achieving a better recognition rate for malicious traffic. Some methods can reduce the false-positive rates but with increased classification and training time. Therefore, it is perceived that false-positive rates are still a problem for the researcher for further future work to minimize false-positive rates. The authors surveyed the machine learning-based work, which summarizes the different papers mostly based on TCP/IP and few are related to IPv6 over Low-Power Wireless Personal Area Networks (6LowPAN). According to this study, further research is needed to improve the false-positive rate for DDoS detection in IoT networks. In this context, we will provide a methodology for detecting DDoS attacks over the SD-IoT network that is based on supervised machine learning. Pande et al. (2020) generates DDoS attacks through the ping of death technique and detects DDoS attacks with machine learning techniques. The experiment is conducted with the random forest (RF) machine learning classifier to classify DDoS traffic using the WEKA tool. The model is trained with a supervised learning algorithm and gets 97.76% results for classification using the NSL-KDD dataset. We also use the WEKA tool and RF classifier; however, we additionally added other supervised learning classifiers such as SVM and DT and Naive Bayes. The testing conducted in our research is distinctive because it is based on the IEEE 802.15.4 protocol with the help of SD-IoT.

SDN despite being a potential network architecture that gives operators more control over a network infrastructure, its architectural entities pose several security risks and targets which makes it vulnerable to DDoS attacks. To tackle this problem, authors in Sahoo et al. (2020) use SDN's centralized control to identify DDoS attacks on the control layer. They proposed an evolutionary SVM model from machine learning to detect malicious traffic. Further, this article integrated the Genetic Algorithm(GA), "KPCA: Kernel Principal Component Analysis" to improve SVM identification accuracy (GA). The experimental findings demonstrate that the proposed model provides more accurate classification and greater generalization than single-SVM. In addition, the proposed model can be implemented within the controller in order to build security rules that block potential attacks. Additionally, Radial Basis Function N-RBF is employed to speed up the learning process. Experimental results also show that KPCA outperforms Principal Component Analysis (PCA) on the DDoS dataset. Their model outperforms the baseline model in accuracy by 0.9897%.

DDoS attacks have always threatened network security. Since its inception, both industry and academia have been exploring DDoS detection and defense. DDoS detection and mitigation methods have been developed so far. Most methods cannot efficiently detect a small number of attacks and fail to minimize false alarms. In Agarwal et al. (2022), the authors present a novel approach to DDoS mitigation using a deep neural network (FS-WOA-DNN) — a new feature selection-whale optimization technique. The input dataset undergoes a min–max normalization approach in the pre-processing phase to replace all of the input within a predetermined range. Following normalization, the data is sent into the proposed FS-WOA to help pick the finest features for classification. A deep neural network classifier is applied to the data to determine whether it is "normal" or "attacked," based on the selected features. The normal data is encrypted using homomorphic methods and safely stored in the cloud, thus strengthening the security of the proposed architecture. The proposed algorithm was simulated and validated using the MATLAB tool and the results indicate that it can find DDoS attacks with a 95.35% accuracy rate.

In a recent study (Agrawal et al., 2022), authors proposed a novel approach towards DDoS detection. They suggested a Modified version

Table 2
Comparison of existing literature on DDoS attack detection techniques.

Study	Traditional DDoS detection techniques				Machine learning DDoS detection techniques			
	CPU utilization	Memory utilization	Detection time	Throughput	CPU utilization	Memory utilization	Detection time	Throughput
Bhayo et al. (2020)	Yes	Yes	Yes	Yes	No	No	No	No
Gillani et al. (2018)	No	No	Yes	Yes	No	No	No	No
Van Adrichem et al. (2014)	Yes	No	Yes	Yes	No	No	No	No
Cui et al. (2016)	Yes	No	No	No	No	No	No	No
Ahmed and Kim (2017)	Yes	Yes	No	No	Yes	Yes	Yes	Yes
Patil et al. (2019)	No	Yes	Yes	No	No	No	No	No
Chen et al. (2019)	No	No	No	No	No	No	Yes	No
Mohammadi et al. (2019)	Yes	Yes	Yes	No	No	No	No	No
Ahmed et al. (2020)	No	No	No	No	Yes	Yes	Yes	Yes
Hameed and Ali (2018)	Yes	Yes	Yes	Yes	No	No	No	No

Table 3
Threat model analysis for SD-IoT.

Vulnerability	Attack type	Detection approaches
Unencrypted data stored	Ransomware attack	Differential area analysis (Davies et al., 2021)
Targeted interference	Jamming attack	Unsupervised learning with clustering (Karagiannis and Argyriou, 2018)
Default, weak, or guessable passwords	Bruteforce attack	IDS using attack patterns (Raikar and Meena, 2021)
Firmware access	Privilege escalation attack	Multi-feature-based behavior of privilege escalation attack detection method (Shen et al., 2020)
Memory corruption	Buffer overflow exploitation	Bio-inspired based approach (Hamidouche et al., 2019)
XSS IoT sensor device	Botnet	N-BaIoT (Meidan et al., 2018)
Directory traversal	HTTP attack	Edge Intelligence (EI)-enabled HTTP anomaly detection framework (An et al., 2021)
Lack of device management	DoS, DDoS	C-DAD (Bhayo et al., 2020)
Insecure network services	Malware	Fuzzy pattern tree methods for malware detection (Dovom et al., 2019)
Security and privacy	Eavesdropping attacks	ML-based detection technique (Xiao et al., 2018)

of a Deep Belief Neural Network (M-DBNN) to achieve low false-positive rates and high prediction accuracy. The Center for Applied Internet Data Analysis (CAIDA) "DDoS Attack 2007" dataset is used to test the proposed model. The method achieves an accuracy of 87%, and its results are compared to those obtained by using a deep neural network (DNN), SVM, an artificial neural network (ANN), and a neural network (NN). High detection accuracy with minimal false positives is a key feature of the suggested approach.

SDN is an approach that utilizes software programs to centrally and intelligently control network design. Separating the control plane of network devices from the data plan simplifies network management. In Adeniji et al. (2023), the authors use SVM to detect DDoS attacks in IPv6-enabled SDNs. The 20-min test generated 500,000 normal and attack traffic packets. The packet data was re-processed and 25% of the data was trained on SVM. The SVM detected 100% potential attacks with 99.69% accuracy.

These solutions defined in the existing literature, detect and prevent DDoS attacks through algorithm-based approaches. Simulating them through a programmable and open-source network can help understand and solve DDoS attacks. The abstract ideas from the relevant research work can assist a great deal in developing machine learning and SD-IoT-based environments. After an extensive literature review and to the best of our knowledge, we concluded that the SDN-IoT had been repeatedly explored for DDoS detection in two different ways, which we defined as, (1) Traditional DDoS detection techniques and (2) Machine learning DDoS detection techniques. Further, both techniques are explored according to the related existing literature with different parameters, such as CPU utilization, memory-utilization, detection-time, and throughput as shown in Table 2.

3. Machine learning-based proposed framework for secure SD-IoT

3.1. Security analysis for the proposed framework

Security analytics is an approach that focuses on data analysis to produce proactive security solutions. Security analytic-based frameworks are frequently designed to detect threats over models or applications. Various solutions can be found focusing on security threats to IoTs on different levels, including network-level, devices-level, and application-level. However, in this research, we have analyzed solutions that mostly investigate communication environment-level threats. IoT devices are used in different applications in large numbers and are vulnerable to various threats, as shown in Table 3. From the literature, we have found several techniques to detect intrusion detection. Table 4 presents a comprehensive list of available techniques, which were evaluated through security analyses to showcase the existing approaches and facilitate comprehension of the research problem.

For the test-bed we designed an SD-IoT network topology that consists of three main components, including the SDNWISE Controller, IoT Controller, and IoT nodes as shown in Fig. 1. The IoT network is divided into different clusters, consisting of SOFS and IoT nodes. The SD-IoT network is designed with normal malicious nodes in each cluster. In this regard, malicious traffic is generated via malicious nodes to detect the DDoS attack. The malicious nodes are programmed to generate flooding traffic for DDoS attacks towards the target node.

This research only focuses on flooding-based network traffic to generate huge network traffic for DDoS attacks. The SD-IoT network consists of an SDNWISE controller and IoT nodes that generate n-1 message traffic in the SD-IoT network. The normal nodes forward legitimate traffic in the SD-IoT network as per the network behavior

Table 4
Comparative analysis different approaches for attack detection.

Detection approach	Short description	Reasoning (Pros/Cons)
Statistics-based	It examines network traffic and processes the data using complicated statistical techniques.	–Requires extensive statistical understanding. –Simple but less precise.
Pattern-based	It tries to recognize the data's characteristics, shapes, and patterns.	–Easy to implement. –A hash function could be used for identification.
Rule-based	To detect a potential attack on suspicious network traffic, it employs an attack "signature".	–Rules require pattern matching and rule-based systems can be computationally costly. –A huge number of rules are required to determine all potential threats. –Low rate of false positives. –High rate of detection.
State-based	It analyzes a series of events to detect any potential attack.	–Probabilistic and self-learning. –Low false positive rate.
Heuristic-based	Identifies any abnormal activity that is out of the ordinary.	–Exploratory and evolutionary learning is required.
ML-based approach	Machine learning models are composed of a collection of rules, procedures, or sophisticated "transfer functions" that may be used to discover significant data patterns or forecast behaviors.	–The implementation of ML models is typically straightforward, but the pros and cons of using such models depend on the specific characteristics of the algorithm in question.

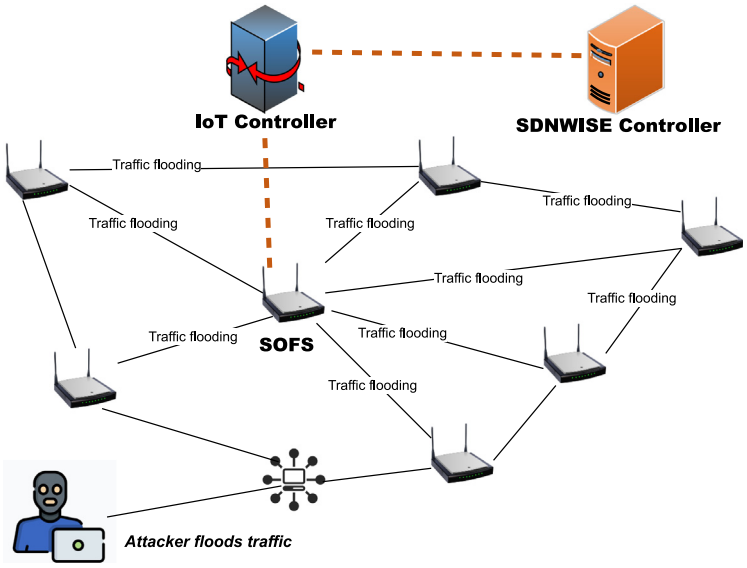


Fig. 1. A general illustration of SD-IoT network topology.

of the particular application. Furthermore, we can also customize the normal node traffic pattern as per application requirements. We conducted different experiments to evaluate results with other parameters, including attack node, packet frequency, and simulation. We vary the packet flooding ratio with varying numbers in the packet frequency parameter to detect the DDoS attack. We also conducted experiments with attack node parameters to compromised IoT nodes with different flooding packet rates. The main objective of this research is to detect the DDoS attack at an early stage. The main advantage of this method is to evaluate and analyze the experiment's result more deeply with different outcome parameters, including CPU and Memory utilization, network throughput, and attack detection time.

As illustrated in Fig. 2, the proposed framework is composed of three modules: (1) A dataplane module, composed of an SD-IoT network, Sensor OpenFlow Switch (SOFS), and IoT devices; (2) An IoT controller module composed of an adjusted SDN-WISE; and (3) a machine learning-based DDoS attack detection module. The SD-IoT network module is composed of IoT nodes and is responsible for managing incoming IoT traffic and serving as a gateway between the source IoT node and the controller. The SDN-WISE controller module is responsible for managing traffic and instructs switches where to send packets. The machine learning detection module classifies IoT node traffic to detect DDoS packets. A detailed explanation of these modules is discussed in the subsequent sections.

3.2. SD-IoT and IoT nodes

IoT involves several heterogeneous devices, which require a unique set of access systems and safety mechanisms. Traditional security approaches such as intrusion detection and prevention systems (IDPS) and Firewalls are deployed at the web edge devices to shield the network from outside attacks. SDN, an intelligent networking paradigm, offers new solutions to understand and solve issues identified with IoT. By applying SDN, network configuration and management can be simplified significantly. Wide acknowledgment for SDN demonstrates that SDN can build a tighter association among the objects in an IoT network. Each IoT device has an IoT specialist that interfaces with the IoT controller. SDN separates the network management operation into network management and packet forwarding at the data plan. SDN has OpenFlow-based switches that forward the packets according to the flow table; however, the unknown packet or switch with no flow information about the received packet forwards to the controller for further assistance. The controller forwards rules about unknown

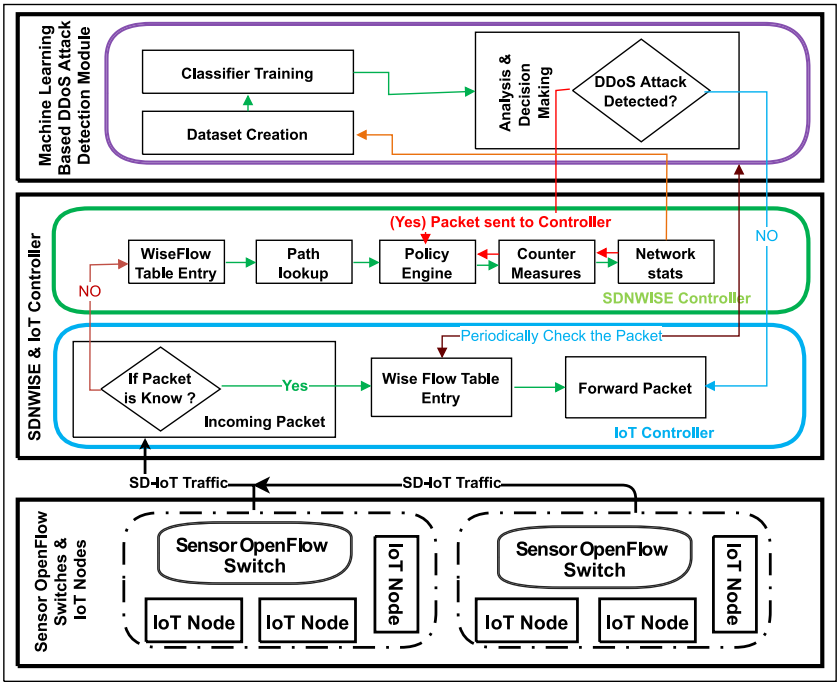


Fig. 2. Proposed ML-based DDoS attack detection framework for SD-IoT networks.

Table 5
Implementation of the adjusted WISE flow table.

Matching rule					Action					Statistics	
Op.	Size	S	Addr.	Value	Type	M	S	Addr.	Value	TTL	Counter
=	1	1	0	0	Modify	1	1	0	1	122	23
=	1	1	0	1	Modify	1	1	0	0	122	120
–	0	–	–	–	Forward	0	0	0	D	122	143
–	0	–	–	–	Drop	0	0	–	–	100	42
–	0	–	–	–	Forward	0	0	0	D	100	32

packets; it also modifies the policies and reprograms the network by installing new software or packages.

The SD-IoT network consists of IoT nodes and a customized Sensor Openflow Switch (SOFS). It can detect, break down and gather information to accomplish the client targets. All of the IoT nodes should be enlisted with the IoT controller along with the points of interest such as their question identifiers, addresses, imparting system conventions, and basic systems. SOFS is a customized OpenFlow flow switch that performs packet forwarding according to the flow table. A request is issued to the Control plane if no entry in the WIreless SEnsor Networks (WISE) Flow Table matches the current packet. Each node must know its optimal next hop towards a node in order to contact the Control plane. Through beaconing, this value is determined in a distributed manner utilizing the Topology Discovery (TD) layer. The SDNWISE Flow table has similar functionalities to the OpenFlow table used in the traditional SDN network. Table 5 shows the three components of the SDNWISE table: Matching Rule, Action, and Statistics (Galluccio et al., 2015). There are three matching conditions against each flow entry, and each matching condition has five fields: Operator, Size, State (S), Offset (Addr), and value. These matching conditions are matched against the rule until the end; if the WISE table does not find the matching rule, it will build the Request Packet and forward it to the controller via the sink node. The packet will only be forwarded to the outgoing interface if the condition is matched.

3.3. SDN-WISE and IoT controller

While existing frameworks provide an efficient network topology for nodes linked to an IoT network, they lack a mechanism for logging communication between nodes. But, packets generated by the nodes are not being recorded, which means that they are needed for various security measures to be implemented. We adjusted the existing framework of the SDN-WISE network by customizing the sink module to the IoT controller and adding SOFS in the SD-IoT network. The proposed framework overcomes the above-discussed issues and integrates machine learning-based security services into the SD-IoT network. The framework contains a logging module that logs all incoming packets in the forwarding layer. These logs are recorded in the controller's directory. To implement the detection module, it is necessary to have information about the communication between nodes as well as the communication's frequency.

The IoT controller receives the traffic from the SD-IoT network and checks if the traffic is known and reliable, then forwards it to the WISE Flow entry component, which further forwards the packets according to flow entry. The decision making for these choices are reflected in the remote physical system utilizing the SDN controller. IoT controller on getting the association from its IoT operator will fabricate the sending rules depending on the systems administration conventions deployed and convey these guidelines to the SDN controller. Once the IoT controller gets the address or identifier of the destination, it needs to identify its source in the network. This is achieved through IoT agents registered with the IoT controller by comparing their identifier or address. As illustrated in Fig. 3, the SDN-WISE framework is composed of three modules: the SDN-WISE controller module, the Sink module, and the IoT Nodes module.

3.3.1. SDN-WISE controller module

The SDN controller controls the switches via OpenFlow protocol for traffic forwarding. It also pushes rules into Openflow-based switches

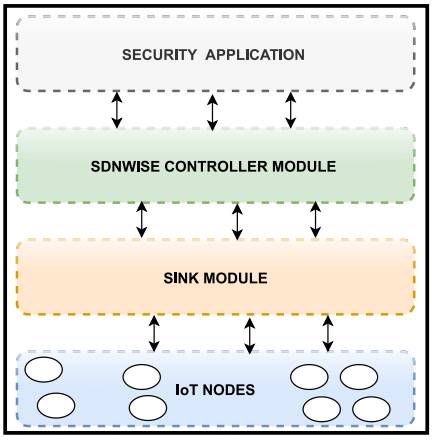


Fig. 3. Existing SDN-WISE framework for SD-IoT network.

to make a decision when network traffic hits them. Switches need to maintain such rules in the flow table. As per flow, such rules are called 'flows', and they are stored in WISE Flow tables. The communication between IoT devices and security applications must be done via the controller. The controller has the OpenFlow protocol used for network configuration, and it is also used to find the best optimal network path for applications.

In the proposed SDN-WISE framework, the nodes are determined by the data structures, i.e., WISE *States Array*, *Accepted Array IDs*, and the WISE *Flow Table*. The controller sends the information contained in these structures; through sending this information, the controller defines networking policies that the nodes in the network must implement. The nodes use the wireless medium for connectivity and the wireless medium has a broadcasting nature; therefore, nodes will also receive packets that are unrelated to them. The information in *Accepted IDs Array* lets the node decide to select the packets that it should process further. If there is no such information in *Accepted IDs Array*, the node will drop the packet. For further processing of the packet, the node will check the entries of the WISE *Flow Table* and check the matching rule. If the matching rule is satisfied, the related action will be performed; however, if not, the packet will be sent to the controller via the sink with a flag indicating a request for further instructions to handle the packet. The SDN-WISE controller module consists of different components that cooperate to perform network operations and control the SD-IoT network. The controller performs network operation at the central point with the help of set components as shown in Fig. 4.

WISE flow table entry. If the received packet has no flow entry in the WISE Flow table in the data plane, a request packet is sent to the controller. Each node in the SD-IoT network must know the path towards the controller via the best next hope path to sink. The best path value is calculated through a beacon packet with the help of the Topology Discovery layer (Abdolmaleki et al., 2017; Shalimov et al., 2013).

Path lookup. Two types of messages are sent from the sensor nodes to the controller, i.e., REPORT and REQUEST. A REPORT message contains an array of local topology information, whereas a REQUEST message contains the source and destination address of the path to be established. The SDN controller handles these messages to build the topology from the REPORT messages and responds to REQUEST messages using the shortest path from the topology. The SDN controller keeps reading messages from the serial interface. If the message is a REPORT message, it extracts the information from the message and uses it to construct the whole topology. Similarly, if the message is a flow REQUEST, it extracts the source and destination of the requested path. Then, it computes the shortest path from the source to destination using Dijkstra's algorithm (Barbehenn, 1998; Jiang et al., 2014) and replies with a RESPONSE message in case a path exists.

Policy engine. The policy engine ensures that packets meet particular requirements. It helps to reduce the complexity of SDN management. It sustains the Quality of Service(QoS) of a specific flow to enforce design constrain. The policy engine handles efficiently in the SDN paradigm.

Network stats. Network administrators must monitor network status for security audit, including network problem tracking, troubleshooting, future planning, network scaling, etc. In this regard, the SDN controller provides flow information and derived network statistics to high-level SDN applications. Either gathered using the southbound protocol or from an external source, such as an IPFIX (Internet Protocol Flow Information Export) (Hofstede et al., 2014) probe. The SDN controllers should derive statistics for high-level traffic identifiers via linking with low-level traffic identifier statistics.

3.3.2. Sink module

The sink is a gateway between the sensor nodes and the controller. All control packets should pass through the sink to reach the controller. The sink module consists of three main components, which are used to communicate with controller and IoT nodes as shown in Fig. 5. The sink implements three layers on top of the MAC layer as part of the data plane protocol stack.

Incoming packet handle. This module is responsible for handling upcoming packets to check whether the packet is known and has a valid entry in the WISE flow table. If the WISE flow table does not have an entry, then the packet handler would send this packet to the controller, as depicted in Fig. 6.

Adjusted WISE flow table. Arriving packets are matched against the WISE flow table. The flow table is composed of three sections, i.e., on matching rules, actions, and statistics (Galluccio et al., 2015), as shown in Table 5.

Each entry in the flow table can have up to three matching conditions as part of the matching rule. Each matching condition has five fields, i.e., *operator*, *size*, *state* (S), *offset* (Addr), and *value*. The relational operator are specified to be used against the value. Offset and size fields specify the starting byte and the number of bytes that needs to be considered starting from the offset. For example, if the size is 2 and the offset is 5, then two bytes starting from byte 5 are used to compare the relational operator with the value. Each SDN-WISE network has a WISE state array which contains the current state for each active controller. The state (S) indicates whether the matching must be done against the current packet or the state. If $S = 0$, the current packet is matched against the value. Whereas, if $S = 1$, the state of IoT node or status of controller is compared against incoming packet with WISE table entry. If all matching conditions are satisfied, the operation in the action part is carried out and the statistics are updated. The action part of the flow

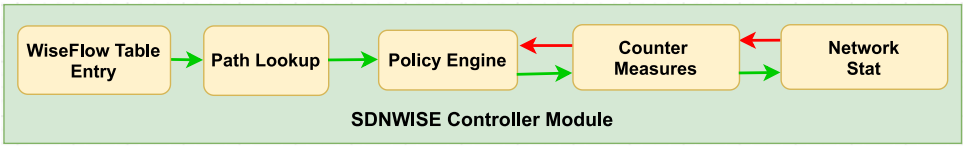


Fig. 4. SDN-WISE controller module.

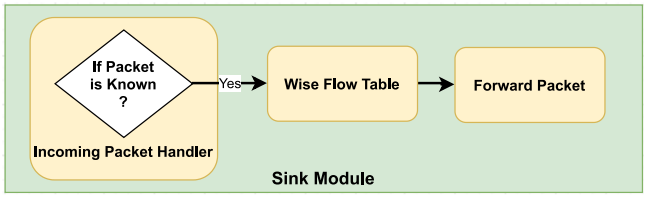


Fig. 5. Sink module.

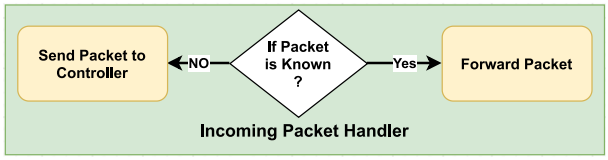


Fig. 6. Incoming packet handler.

table entry is composed of five fields, i.e., *type*, M, S, *offset*, and *value*. The action types are *Forward*, *Drop*, *Modify*, *ASK*, etc. When the action is *Modify*, the S field specifies whether to modify the WISE state array or the current packet. The M field specifies whether only one matching entry must be executed or not. After one successful flow table entry is matched and the corresponding action is executed; if $M = 0$, SDN-WISE stops browsing the flow table. However, it keeps searching for other matching rules if $M = 1$. If no matching rule is found for the incoming packet, a request packet for a flow table entry is sent to the controller via the sink.

Forwarding layer (FWD). The sink implements a forwarding layer that is responsible for handling incoming packets according to the rules specified in the WISE flow table, which is analogous to the flow table in OpenFlow. It also keeps updating the WISE flow table according to the flow instructions sent from the controller. The FWD is responsible for handling incoming SDN-WISE packets. The header of SDN-WISE packets has a fixed length of 10 bytes and is made of seven fields as shown in Table 6. SDN-WISE defines eight packet types (SDN, 2021), as follows:

- Data Packet: It consists of the data packet having variable payload size.
- Beacon Packet: It is a broadcast packet that reports the distance of a source node from the sink and its battery level information.
- Report Packet: Reports the list of neighbors and is route to the sink.
- Request Packet: Request forwarded to the controller to encapsulate the unknown flow via the sink node.
- Response Packet: The controller forwards the rule of a requested unknown packet via response packet to the Openflow switch for the flow entry of the requested packet.
- Open-path Packet: It is used to create a path between two nodes in the network.
- Config Packet: The controller uses this packet to read/write the configuration information of any node or to forward the configuration information required to the node from the controller.
- Reg Proxy Packet: This packet is used to inform the control plane about a sink and the SD-IoT network or further information about it.

3.3.3. IoT nodes

In the SDN-based IoT network, every node is directly connected to the sink via nodes maintaining the flow table having the best next hop towards the sink. Each node itself sets this path by running a protocol in which some of the information packets are exchanged between the nodes containing information related to battery level or IoT node's power status and hop count to the sink, as there could be multiple sinks in a network. Therefore, the route to the nearest sink will be preferred by the node.

Table 6
SDN-WISE packet header fields.

Byte(s)	Name	Description
0	NET	Identifier of network
1	LEN	Total length of packet
2–3	DST	Destination address
4–5	SRC	Source destination
6	TYP	Packet type
7	TTL	Number of hopes remaining
8–9	NXH	Next hop address

3.4. Machine learning based ddos attack detection module

The machine learning detection module is a completely independent module that has been developed by us as a feature to be used inside the SDN-Wise controller. Whenever the IoT controller receives a packet via SOFS from the SD-IoT network, it first checks whether the packet is legitimate or unknown. If the packet is unknown, the ML detection module forwards it to the SDN-WISE controller for further inspection. We have used ML as a black box, our main concern for this research is to detect DDoS in SD-IoT and we trained models concerning the standard requirements of the particular model. The features that were used for training are: (i) IoT Nodes (ii) Simulation Time (iii) Packet Frequency and (iv) Detection Time in ms.

3.4.1. Selection of classifier

DDoS is a persistent problem due to variances in its attack strengths and types. Researchers are continuously working on detecting and mitigating DDoS using various state-of-the-art solutions and algorithms, including machine learning techniques. In this step, our framework is designed to select the classifiers as per the defined workflow. In our case, we have selected three supervised machine learning algorithms, i.e., Naive Bayes, DT, and support vector machines (SVM), to analyze the data sets. The *pros and cons* of each classifier is summarized in Table 7.

3.4.2. Configuration of machine learning module

The module has been converted into Java JAR file so that it can be used with any framework. A JAR file extension needs to be included in the SDN-WISE controller which will facilitate passing packets as a command-line argument along with a classifier name (i.e., Naive Bayes, DT, or SVM), and as a result, the JAR file will contain the details of a packet accordingly. Based on this classification result, a network engineer can decide to put a specific rule in the controller to forward or drop certain packets.

4. Testbed and experimental setup

This section explains the details of the implemented testbed and evaluates the generated results acquired from different sets of experiments. The performance of the machine learning-based DDoS detection module integrated with the SDN-WISE controller will be discussed in detail as well.

4.1. Testbed setup

For performing the experiments, the specification of the deployed testbed consists of Ubuntu v16.0.2, Intel® Core™ i7-3540M 3.00 GHz processor, and 4.0 GB RAM. SDN-WISE has been integrated as a working environment on the testbed setup, which has been used to simulate an SDN-IoT traffic generation. The machine learning library WEKA

Table 7
Pros and Cons of selected machine learning classifiers.

Classifier	Class nature	Pros	Cons
SVM	Decision boundary	– SVM effectively learns from a small training set – Works well with binary classification – Can model complex and nonlinear relationships – Performs well when classes are separable – Outliers have less impact – Robust to noise (because it maximizes margins)	– Does not perform well with large datasets – Results on multiple classification tasks are not satisfactory – Selecting appropriate hyperparameters is important – Requires significant processing power and memory – Sensitive to kernel function parameters
NB	Probabilistic	– NB is very fast in real-time predictions – Very flexible with larger datasets – Fast to execute high-performance – Performs effectively with multi-class – Works well with higher dimensions – Robust to noise – Capability to learn incrementally	– Slow at training – Not a good estimator – Fails when one of the certain features has zero occurrences, the posterior probability will be zero, so training data should represent the population effectively – Does not perform well on attribute-related data sets
DT	Tree	– In DT normalization/ scaling of data is not needed – Effectively handles missing values – Easy to explain/visualize due to graphical representation – Automatically selects features – Produces a strong interpretation	– DT is more prone to overfitting – Sensitive to changes in data – Requires more time for training – Results are biased to the majority class – Ignores the correlation of data

("Waikato Environment for Knowledge Analysis") (Witten et al., 2005) has been used for the application of the classification techniques. To measure the efficiency of the algorithms, each classifier has been trained on our dataset using 20% of the collected data as training data and 80% of the collected data as test data. The Cooja simulator (Österlind et al., 2006) is highly recommended and mostly used for IoT and wireless sensor networks as the Cooja simulator and Contiki focuses on low power consuming devices.

4.1.1. Proposed SD-IoT network architecture

In this research, the SD-IoT-based network model consists of four main components, including ML-based DDoS Attack detection Module, SDNWISE controller, IoT controller, and SD-IoT network. The SD-IoT network components have IoT devices, including sensors, smart devices, and others that communicate via SOFS. We designed a network that has malicious as well as normal nodes. These IoT devices are generating malicious and normal traffic in the SD-IoT network. The SOFS acts as a forwarding device to forward the SD-IoT network traffic according to the flow table. IoT controller is used as a mediator between SD-IoT network and Machine learning-based security applications, and these applications are running at the top SDNWISE controller. SDNWISE controller provides the network management functionalities and exposes the north-bound APIs for security applications. The ML-based attack detection module runs at the top of SDNWISE and detects the DDoS attack using machine learning algorithms.

The experiments carried out in this research are based on variations of the following attributes:

- Number of IoT nodes
- Simulation time
- Packet frequency (normal and burst mode) (packets/min)
- Number of attack nodes

The sizes of the above-mentioned parameters change between experiments used to evaluate the model. We performed numerous experiments to determine resource utilization, including CPU, memory, and detection time, while adjusting the size of each parameter to obtain the optimal result.

4.2. Results and evaluations

In this research, we conducted different experiments by generating malicious and normal traffic through IoT devices to observe the utilization of resources. Moreover, we also observed controller resources through these experiments with varying parameters. The details of each experiment and their respective results are presented as follows:

Table 8
Experiment A1.

Algorithm name	IoT nodes (Vary)	Simulation time	Packet frequency	Detection time in ms
Naive Bayes	5	15 min	20 packet/min	578
Decision Tree	5	15 min	20 packet/min	481
SVM	5	15 min	20 packet/min	602

Table 9
Experiment A2.

Algorithm name	IoT nodes (Vary)	Simulation time	Packet frequency	Detection time in ms
Naive Bayes	15	15 min	20 packet/min	576
Decision Tree	15	15 min	20 packet/min	482
SVM	15	15 min	20 packet/min	609

Table 10
Experiment A3.

Algorithm name	IoT nodes (Vary)	Simulation time	Packet frequency	Detection time in ms
Naive Bayes	30	15 min	20 packet/min	576
Decision Tree	30	15 min	20 packet/min	502
SVM	30	15 min	20 packet/min	622

4.2.1. Experiment-A

To analyze the attack detection times for different algorithms in the SD-IoT network, we conducted five separate experiments as shown in Tables 8 to 12. In each experiment, we changed the IoT nodes with other fix parameters. Experiment A is based on the following parameters,

- Varying number of IoT nodes
- Fix simulation time
- Fix packet frequency

Experiment A has been performed with five different variations. Three different machine learning classifiers have been utilized, i.e., Naive Bayes, a Decision Tree (DT) classifier, and a support vector machine classifier. It has been revealed that the three selected classifiers have taken early the same time for detection. Furthermore, it has been observed that an SD-IoT network with 5 to 45 nodes can effectively utilize a machine learning detection module for very favorable outcomes, as shown in Fig. 7. The DT classifier proved efficient in average classification time, in comparison to Naive Bayes and support vector machine.

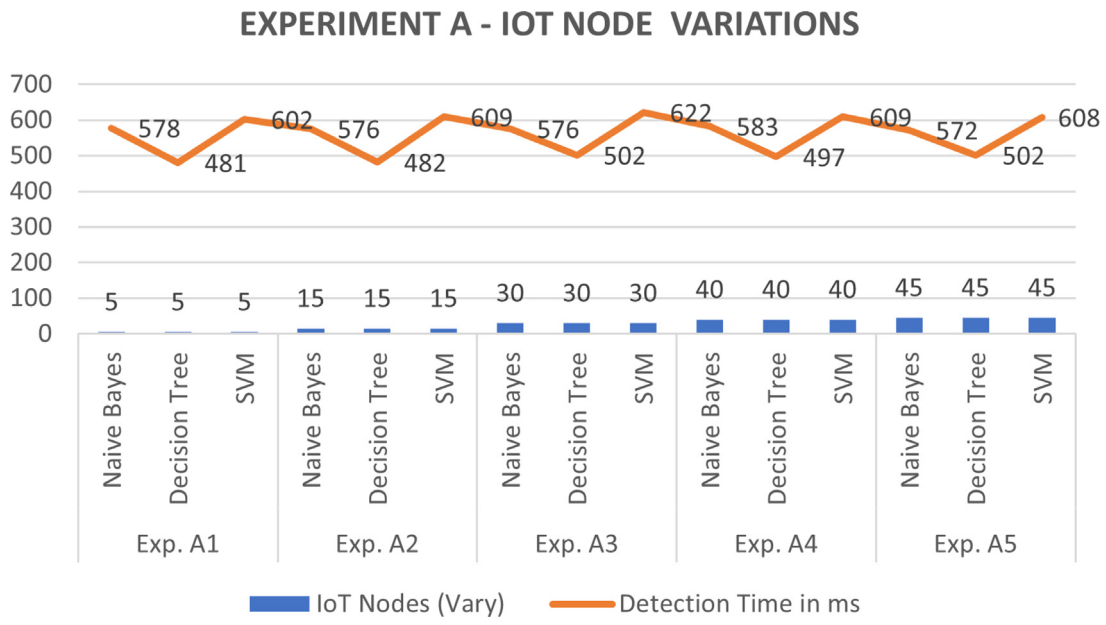


Fig. 7. Experiment A with IoT node variations.

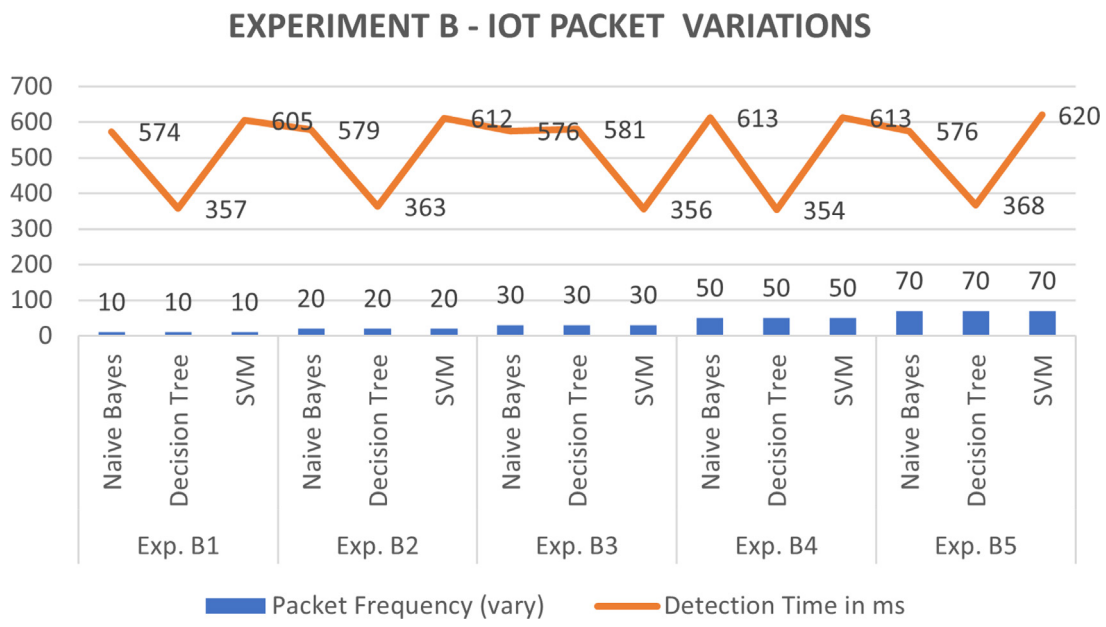


Fig. 8. Experiment B with IoT packet variations.

Table 11
Experiment A4.

Algorithm name	IoT nodes (Vary)	Simulation time	Packet frequency	Detection time in ms
Naive Bayes	40	15 min	20 packet/min	583
Decision Tree	40	15 min	20 packet/min	497
SVM	40	15 min	20 packet/min	609

Table 12
Experiment A5.

Algorithm name	IoT nodes (Vary)	Simulation time	Packet frequency	Detection time in ms
Naive Bayes	45	15 min	20 packet/min	572
Decision Tree	45	15 min	20 packet/min	502
SVM	45	15 min	20 packet/min	608

4.2.2. Experiment-B

In this experiment, we use the packet frequency as the variable parameter, whereas the other parameters have constant values, as shown in Tables 13 to 17. Experiment B also has been performed with five different variations. Similar to experiment A, it has been observed that all of the selected classifiers have taken similar detection time, the selected classifiers include Naive Bayes, DT, and Support Vector Machine. The results have shown that the detection module needs a maximum of 70 packets/min, without affecting detection time. However, the average classification time for DT is less in comparison with Naive Bayes and support vector machine, as shown in Fig. 8. Experiment B is based on the following parameters,

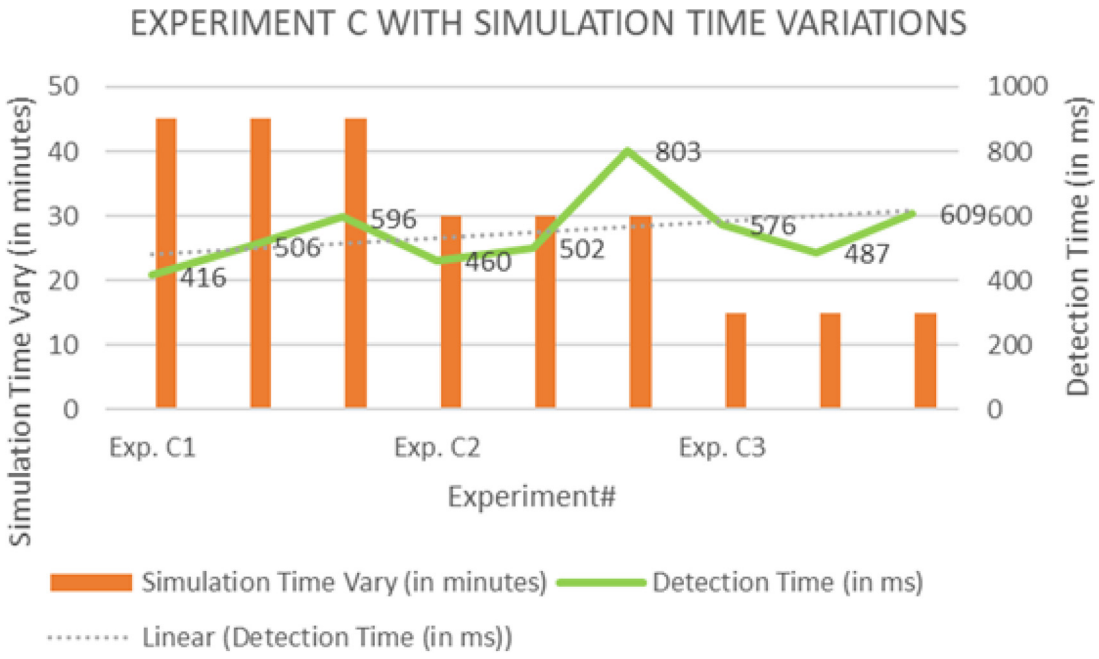


Fig. 9. Experiment C with simulation time variations.

Table 13
Experiment B1.

Algorithm name	IoT nodes	Simulation time	Packet frequency (Vary)	Detection time in ms
Naive Bayes	15	15 min	10 packet/min	574
Decision Tree	15	15 min	10 packet/min	357
SVM	15	15 min	10 packet/min	605

Table 14
Experiment B2.

Algorithm name	IoT nodes	Simulation time	Packet frequency (Vary)	Detection time in ms
Naive Bayes	15	15 min	20 packet/min	579
Decision Tree	15	15 min	20 packet/min	363
SVM	15	15 min	20 packet/min	612

Table 15
Experiment B3.

Algorithm name	IoT nodes	Simulation time	Packet frequency (Vary)	Detection time in ms
Naive Bayes	15	15 min	30 packet/min	581
Decision Tree	15	15 min	30 packet/min	356
SVM	15	15 min	30 packet/min	613

Table 16
Experiment B4.

Algorithm name	IoT nodes	Simulation time	Packet frequency (Vary)	Detection time in ms
Naive Bayes	15	15 min	50 packet/min	578
Decision Tree	15	15 min	50 packet/min	354
SVM	15	15 min	50 packet/min	613

Table 17
Experiment B5.

Algorithm name	IoT nodes	Simulation time	Packet frequency (Vary)	Detection time in ms
Naive Bayes	15	15 min	70 packet/min	576
Decision Tree	15	15 min	70 packet/min	368
SVM	15	15 min	70 packet/min	620

Table 18
Experiment C1.

Algorithm name	IoT nodes	Simulation time (Vary)	Packet frequency	Detection time in ms
Naive Bayes	15	45 min	20 packet/min	416
Decision Tree	15	45 min	20 packet/min	506
SVM	15	45 min	20 packet/min	596

Table 19
Experiment C2.

Algorithm name	IoT nodes	Simulation time (Vary)	Packet frequency	Detection time in ms
Naive Bayes	15	30 min	20 packet/min	460
Decision Tree	15	30 min	20 packet/min	502
SVM	15	30 min	20 packet/min	803

Table 20
Experiment C3.

Algorithm name	IoT nodes	Simulation time (Vary)	Packet frequency	Detection time in ms
Naive Bayes	15	15 min	20 packet/min	576
Decision Tree	15	15 min	20 packet/min	487
SVM	15	15 min	20 packet/min	609

- Varying frequency of packet
- Fix simulation time
- Fix number of IoT nodes

4.2.3. Experiment-C

Experiment C has been performed with five different variations, as shown in Tables 18 to 20. In the experiment, the interval of simulation time has been varied from 45 to 15 min, and it is revealed that there was no effect on the detection time of the machine learning module, as shown in Fig. 9. However, in the case of average classification time, the performance of the Decision Tree (DT) classifier was better in comparison with the remaining two classifiers. Experiment C is based on the following parameters,

- Varying simulation time
- Fix number of IoT-nodes
- Fix frequency of packet

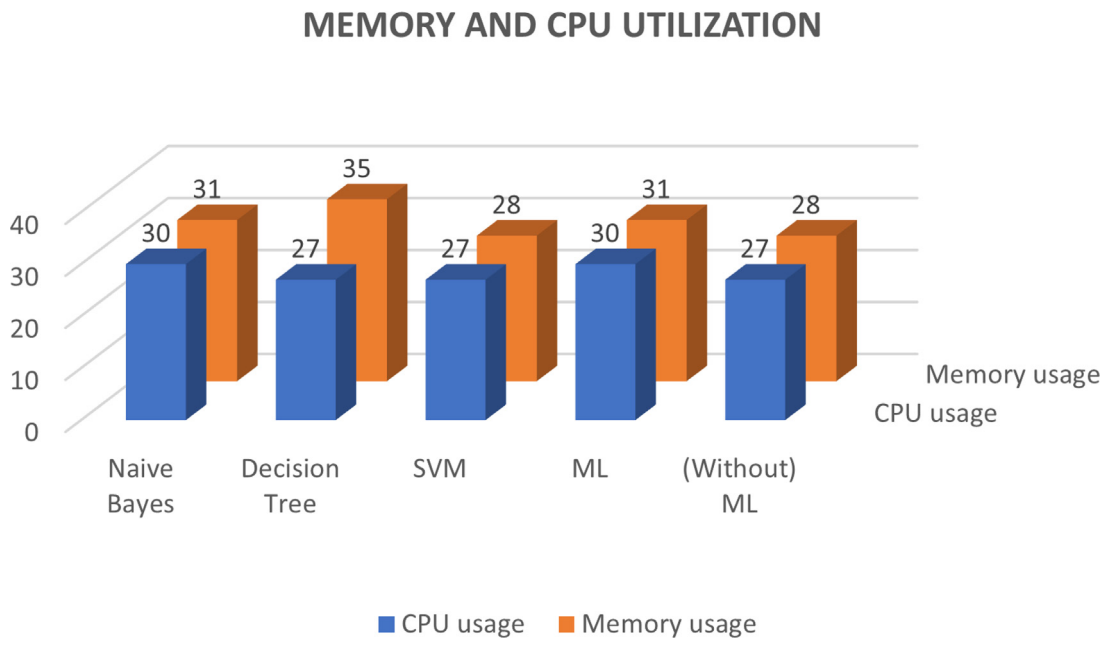


Fig. 10. Experiment D1: CPU and memory usage — Machine Learning (ML) vs. Non-Machine Learning module.

Table 21
Experiment-E: SDN-WISE controller throughput.

Throughput (No. of packets)	Packets
Total throughput in 6 h	1 049 756
AVG throughput/h	174 959
AVG throughput/min	2915
AVG throughput/s	48

Table 22
Average summary of accuracy and detection rate.

Classifier	Accuracy %	Detection rate
Naive Bayes	97.4	554
Decision Tree	98.1	432
SVM	96.1	625
Overall Average	97.2	537

4.2.4. Experiment-D1: CPU and memory usage comparison of different machine learning algorithms and without machine learning module

Usage of CPU and Memory consumption has been observed by performing experiment D1. The experiment proved that the machine learning module put less burden on the CPU, and it only uses an additional 3% of CPU for its functionality. Moreover, similar to CPU usage, the machine learning module also utilized 3% of additional memory usage for its functionality as shown in Fig. 10.

This experiment has been performed to compare the utilization of CPU and memory for different classifiers. The results of the experiment have shown that the support vector machine and DT have utilized almost equal quantities of CPU usage. However, the DT needs more memory than the support vector machine and Naive Bayes as shown in Fig. 10.

4.2.5. Experiment-D2: Memory and CPU utilization: Periodic check vs. all packets

Experiment D2 has been performed to measure the utilization of CPU and memory, in a scenario when each packet is sent to a machine learning-based classifier, instead of sending it to the SD-IoT controller. It has been observed that additional CPU and memory have been utilized by all of the selected machine learning classifiers when they acquire packets in this manner. The results are shown in Fig. 11. For the selected scenario, the DT classifier proved to be most effective in CPU utilization, but least effective in memory usage.

4.2.6. Experiment-E

This experiment is used to calculate the throughput of the detection module. The simulation for this experiment has been executed for a total of 6 hours, and the outcome has shown that the machine-learning detection module can process the quantity of 48 packets per second, as shown in Table 21. According to the results shown in the above subsections, we can conclude that our machine learning-based DDoS attack detection module efficiently detects the attack with additional usage of CPU 3% and memory 3% with the machine learning module.

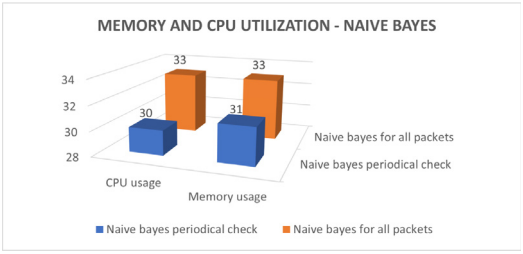
4.2.7. Experiment-F

In our framework, we considered accuracy as an average result of classified packets while the detection rate is the total packet detection rate. Results shown in Table 22 are the average accuracy and detection rate of all the experiments performed in this research. We achieved an accuracy of 97.4% for Naive Bayes, 98.1% for Decision Tree, and 96.1 for the SVM classifier. It is concluded from Tables 8 to 20 that on average the Decision Tree model outperforms in both experiments (A & B), while Naive Bayes performed better for experiment C. The proposed framework has an overall average score of 97.2%.

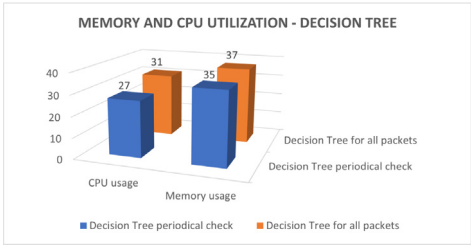
4.3. Discussion

Numerous experiments are carried out in this research to evaluate the performance of a machine learning-based DDoS attack detection application in SD-IoT networks. This section presents the conclusions from the experiments carried out in this study.

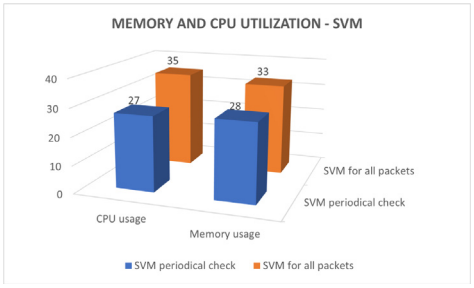
IDS detects DDoS attacks in traditional networks using different techniques with regard to performance on factors including CPU, memory, throughput, and attack detection time. In the attack response, the framework notifies about the malicious flow to the controller module whenever the DDoS attack is detected by a Machine learning-based DDoS Attack detection module. The SDNWISE controller will take different attack mitigation actions; the attack countermeasures might be a flow of attacking nodes being removed from the flow table and nodes marked as malicious. However, the proposed framework detects the DDoS attack in the SD-IoT network with very low time and saves and



(a) Naive Bayes



(b) Decision Tree



(c) SVM

Fig. 11. Experiment D2: CPU and memory usage of ML classifiers — all-packets' check vs. periodic checks.

optimizes the resources as compared to traditional IDS. Furthermore, these mitigation approaches are based on the proper OSI model layer, such as layer 3, layer 4, and layer 7. These mitigating approaches include Random Port Hopping, network filtering (Egress and Ingress), and a technique in which the flow is branded as malicious, and the packets of the labeled flow are dropped (Rai and Challa, 2016; Sahay et al., 2015).

The experiment demonstrates that augmenting the number of IoT nodes through standard node parameters does not alter CPU or memory utilization. However, elevating IoT nodes' burst mode, attack node, or packet payload parameters has an impact on CPU and memory consumption, as well as the SD-IoT network throughput and controller workload parameters. As the algorithm utilizes counters to store integer or floating-point data, changes to these parameters do not affect the algorithm. The IoT nodes in the study are limited in number and transmit messages at a rate of one per second or, in burst mode, up to a maximum of 1000 messages per second. The storage range of the counter variables accommodates a broad spectrum of counter values and, therefore, does not have any impact on the algorithm's counter variables. Furthermore, counter-based algorithms rely on counter values that are reprogrammed with threshold values. The algorithm creates the DDoS attack warming message once the counter hits the threshold number.

4.3.1. Attack detection time

According to our results, the Decision Tree (DT) classifier efficiently classifies the malicious traffic in 480 to 500 ms with a minimum of 5 and a maximum of 45 IoT nodes as compared to Naive Bayes and SVM with other fixed parameters as shown in Tables 8 to 12. With a varying packet frequency, the attack detection time was reduced with DT classifier, however, we could not observe significant differences by increasing the packet frequencies. Although, when we conducted the experiments with varying sizes of simulation parameters we get different results. In this scenario, the Naive Bayes classifier gets less time with 30 and 45 simulation times. There is a big variation in detection time with the SVM classifier showing 596, 803, and 609 ms at 45, 30, and 15 min of simulation time respectively. The DT classifier is best and SVM gets the highest time to detect the attack with minimum simulation time as shown in Tables 18 to 20. According to the results, increasing the IoT node parameter has a negligible effect on detection time, but increasing the packet frequency parameter causes detection time to fluctuate.

4.3.2. CPU and memory utilization

According to the results of the experiments D1-D2 conducted for CPU and memory utilization, the SVM classifier gets less memory and CPU utilization compared to Naive Bayes and DT. The DT classifier utilizes the highest memory 35% as compared to others classifiers with the lowest CPU usage 27%. It is also observed that the SVM classifier has minimum usage of CPU and memory with periodic checking and gets the highest utilization of memory and CPU during all packet checking. The DT on the other hand utilizes high memory periodically as well as all packet checking as compared to other classifiers.

4.3.3. SDN-WISE controller throughput with machine learning module

The SDN-WISE controller processes a total of 1 049 756 packets in the experiment and processes 48 packets in a second at an average of 174 959 packets in one hour. It is also revealed that the module takes approximately 30% usage of memory and CPU and saves about 70% memory and keeps CPU free to 70% to process the SD-IoT network traffic. For the sake of performance evaluation, a comparison of the machine learning module results with other published work is provided in Table 23.

5. Conclusion and future work

IoT devices are critical components of today's digital ecosystem, as they provide service availability and mobility. Because IoT devices operate on low power and have limited resources and are typically deployed in open spaces, they are vulnerable to a variety of threats. This research focuses on the security risks associated with IoT devices. Detecting and preventing DDoS attacks made at and via IoT devices is crucial for any IoT system. This study introduces a novel machine learning-based approach for detecting DDoS attacks. The attack detection service is based on SDN and is placed on a centralized network management controller, allowing for efficient protection of the IoT from threats. The network has been designed with both normal and malicious nodes in order to create a large amount of traffic. On the top SDNWISE controller, the DDoS attack detection program classifies traffic using machine learning classifiers such as Naive Bayes, DT, and Support Vector Machine. It will be advantageous to implement countermeasures for early detection of DDoS attacks. As a result, we may isolate IoT devices that communicate with malicious nodes, avoiding

Table 23
Comparison of the proposed framework's results with other studies.

Study	Year	ML algorithms	Results	Our proposed framework
Zhang et al. (2020)	2020	Random forest	The authors achieved different results by varying features of RF. Accuracy has been the major factor in evaluation. While authors also optimized run-time overheads by reducing RF forest size and tree path.	In comparison to this framework, we used three different algorithms and achieved an accuracy rate of 97.4%, 96.1%, and 98.1%, for Naïve Bayes, SVM, and DT respectively. From the results, it is noticed that the forest size affects the accuracy.
Chen et al. (2020)	2020	Decision tree	F1-score is reported to be approximately 97% showing that the system detects DDoS attacks with high accuracy	In comparison to the presented study, we used three different algorithms and the DT achieved an accuracy ratio of 98.1%.
Silveira et al. (2020)	2020	Random Forest, Logistic Regression, and Extreme Gradient Boost (XGP)	The authors achieved results at a sampling rate of 20% of network traffic, showing high precision of approximately 93%, and a low false alarm rate of 96%.	Our proposed framework, on average, achieved an accuracy of 97.2%

the creation of a higher level of attack. To evaluate the developed framework, various experiments were conducted using a range of attack scenarios via simulation. By incorporating supervised and unsupervised classifiers, we may extend this research into constructing an attack mitigation solution and obtaining a more optimized result. Additionally, the solution may be deployed to include various sorts of DDoS attacks and train it on a variety of IoT-generated datasets to get new insights and enhance the existing framework.

This work can be extended through other supervised learning algorithms including Random Forest, Xg boost, and other statistical-based approaches to machine learning. Furthermore, we can also such as unsupervised learning, semi-supervised learning, and reinforcement learning. Additionally, it can be also extended to integrate a DDoS attack Mitigation module to drop the malicious traffic and block the vulnerable nodes. This paper only focuses on the flooding types of DDoS attacks; however, it can be extended to other types of DDoS attacks and also with different types of IoT networks.

CRediT authorship contribution statement

Jalal Bhayo: Conceptualization, Methodology/study design, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Syed Attique Shah:** Conceptualization, Methodology/study design, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Supervision, Project administration. **Sufian Hameed:** Conceptualization, Methodology/study design, Software, Validation, Formal analysis, Writing – original draft, Writing – review & editing, Supervision, Project administration. **Awais Ahmed:** Conceptualization, Methodology/study design, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft. **Jamal Nasir:** Methodology/study design, Software, Resources, Data curation. **Dirk Draheim:** Formal analysis, Writing – original draft, Writing – review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Syed Attique Shah reports financial support and administrative support were provided by Birmingham City University.

Data availability

No data was used for the research described in the article.

References

Abdolmaleki, Nasim, Ahmadi, Mahmood, Malazi, Hadi Tabatabaee, Milardo, Sebastiano, 2017. Fuzzy topology discovery protocol for SDN-based wireless sensor networks. Simul. Model. Pract. Theory 79, 54–68.

Adeniji, Oluwashola David, Adekeye, Deji Babatunde, Ajagbe, Sunday Adeola, Adesina, Ademola Olusola, Oguns, Yetunde Josephine, Oladipupo, Matthew Abiola, 2023. Development of DDoS attack detection approach in software defined network using support vector machine classifier. In: Pervasive Computing and Social Networking. Springer, pp. 319–331.

Agarwal, Ankit, Khari, Manju, Singh, Rajiv, 2022. Detection of DDOS attack using deep learning model in cloud storage application. Wirel. Pers. Commun. 127 (1), 419–439.

Agrawal, Ankit, Singh, Rajiv, Khari, Manju, Vimal, S, Lim, Sangsoon, 2022. Autoencoder for design of mitigation model for DDOS attacks via M-DBNN. Wirel. Commun. Mob. Comput. 2022.

Ahmad, Ijaz, Namal, Suneth, Ylianttila, Mika, Gurtov, Andrei, 2015. Security in software defined networks: A survey. IEEE Commun. Surv. Tutor. 17 (4), 2317–2346.

Ahmed, Awais, Hameed, Sufian, Rafi, Muhammad, Mirza, Qublai Khan Ali, 2020. An intelligent and time-efficient DDoS identification framework for real-time enterprise networks: SAD-F: Spark based anomaly detection framework. IEEE Access 8, 219483–219502.

Ahmed, A., Hameed, S., Rafi, M., Mirza, Q.K.A., 2020. An intelligent and time-efficient DDoS identification framework for real-time enterprise networks: SAD-F: Spark based anomaly detection framework. IEEE Access 8, 219483–219502. http://dx.doi.org/10.1109/ACCESS.2020.3042905.

Ahmed, M. Ejaz, Kim, Hyoungshick, 2017. DDoS attack mitigation in Internet of Things using software defined networking. In: 2017 IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService). IEEE, pp. 271–276.

Alamri, Hassan A., Thayananthan, Vijey, 2020. Bandwidth control mechanism and extreme gradient boosting algorithm for protecting software-defined networks against DDoS attacks. IEEE Access 8, 194269–194288.

Ali, Ihsan, Ahmed, Abdelmuttlib Ibrahim Abdalla, Almogren, Ahmad, Raza, Muhammad Ahsan, Shah, Syed Attique, Khan, Anwar, Gani, Abdullah, 2020. Systematic literature review on IoT-based botnet attack. IEEE Access 8, 212220–212232. http://dx.doi.org/10.1109/ACCESS.2020.3039985.

An, Yufei, Yu, F. Richard, Li, Jianqiang, Chen, Jianyong, Leung, Victor C.M., 2021. Edge intelligence (EI)-enabled HTTP anomaly detection framework for the internet of things (IoT). IEEE Internet Things J. 8 (5), 3554–3566. http://dx.doi.org/10.1109/JIOT.2020.3024645.

Barbehenn, Michael, 1998. A note on the complexity of dijkstra's algorithm for graphs with weighted vertices. IEEE Trans. Comput. 47 (2), 263.

Bawany, Narmeen Zakaria, Shamsi, Jawwad A., 2019. SEAL: SDN based secure and agile framework for protecting smart city applications from DDoS attacks. J. Netw. Comput. Appl. 145, 102381.

Bhayo, Jalal, Hameed, Sufian, Shah, Syed Attique, 2020. An efficient counter-based DDoS attack detection framework leveraging software defined IoT (SD-IoT). IEEE Access 8, 221612–221631.

Bhayo, Jalal, Jafaq, Riaz, Ahmed, Awais, Hameed, Sufian, Shah, Syed Attique, 2022. A time-efficient approach toward DDoS attack detection in IoT network using SDN. IEEE Internet Things J. 9 (5), 3612–3630. http://dx.doi.org/10.1109/JIOT.2021.3098029.

Chen, Yixin, Pei, Jianing, Li, Defang, 2019. Detpro: A high-efficiency and low-latency system against DDoS attacks in SDN based on decision tree. In: ICC 2019-2019 IEEE International Conference on Communications (ICC). IEEE, pp. 1–6.

Chen, Yi-Wen, Sheu, Jang-Ping, Kuo, Yung-Ching, Van Cuong, Nguyen, 2020. Design and implementation of IoT DDoS attacks detection system based on machine learning. In: 2020 European Conference on Networks and Communications (EuCNC). IEEE, pp. 122–127.

Chernyshev, Maxim, Baig, Zubair, Bello, Oladayo, Zeadally, Sherali, 2017. Internet of things (IoT): Research, simulators, and testbeds. IEEE Internet Things J. 5 (3), 1637–1647.

Chhabra, Meghna, Gupta, Brij, Almomani, Ammar, 2013. A novel solution to handle DDOS attack in MANET. J. Inf. Secur. 4 (3), 165–179.

2020. Corero DDoS trends report. http://info.corero.com/rs/258-JCF-941/images/2017-q2q3-ddos-trends-report.pdf (Accessed on 10/06/2020).

da Costa, Kelton AP, Papa, João P, Lisboa, Celso O, Munoz, Roberto, de Albuquerque, Victor Hugo C, 2019. Internet of things: A survey on machine learning-based intrusion detection approaches. Comput. Netw. 151, 147–157.

Cui, Jie, Wang, Mingjun, Luo, Yonglong, Zhong, Hong, 2019. DDoS detection and defense mechanism based on cognitive-inspired computing in SDN. Future Gener. Comput. Syst. 97, 275–283.

Cui, Yunhe, Yan, Lianshan, Li, Saifei, Xing, Huanlai, Pan, Wei, Zhu, Jian, Zheng, Xiaoyang, 2016. SD-Anti-DDoS: Fast and efficient DDoS defense in software-defined networks. J. Netw. Comput. Appl. 68, 65–79.

Davies, Simon R., Macfarlane, Richard, Buchanan, William J., 2021. Differential area analysis for ransomware attack detection within mixed file datasets. Comput. Secur. 102377.

2016. DDoS attack that disrupted internet was largest of its kind in history, experts say. https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet (Accessed on 03/03/2021).

2021. DDoS attacks spiked, became more complex in 2020. https://www.darkreading.com/attacks-breaches/ddos-attacks-spiked-became-more-complex-in-2020/d/d-id/1339814 (Accessed on 16/01/2021).

Di Mauro, M., Galatro, G., Fortino, G., Liotta, A., 2021. Supervised feature selection techniques in network intrusion detection: A critical review. Eng. Appl. Artif. Intell. 101, 104216.

Dovom, Ensieh Modiri, Azmoodeh, Amin, Dehghantanha, Ali, Newton, David Ellis, Parizi, Reza M, Karimipour, Hadis, 2019. Fuzzy pattern tree for edge malware detection and categorization in IoT. J. Syst. Archit. 97, 1–7.

Du, Miao, Wang, Kun, 2019. An SDN-enabled pseudo-honeypot strategy for distributed denial of service attacks in industrial Internet of Things. IEEE Trans. Ind. Inform. 16 (1), 648–657.

Eskin, Eleazar, Arnold, Andrew, Prerau, Michael, Portnoy, Leonid, Stolfo, Sal, 2002. A geometric framework for unsupervised anomaly detection. In: Applications of Data Mining in Computer Security. Springer, pp. 77–101.

Galluccio, Laura, Milardo, Sebastiano, Morabito, Giacomo, Palazzo, Sergio, 2015. SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless SEnsor networks. In: 2015 IEEE Conference on Computer Communications (INFOCOM). IEEE, pp. 513–521.

Ghaffar, Zeba, Alshahrani, Abdullah, Fayaz, Muhammad, Alghamdi, Ahmed Mohammed, Gwak, Jeonghwan, 2021. A topical review on machine learning, software defined networking, internet of things applications: Research limitations and challenges. Electronics 10 (8), 880.

Gillani, Fida, Al-Shaer, Ehab, Duan, Qi, 2018. In-design resilient SDN control plane and elastic forwarding against aggressive DDoS attacks. In: Proceedings of the 5th ACM Workshop on Moving Target Defense. pp. 80–89.

Haddadi, Fariba, Khanchi, Sara, Shetabi, Mehran, Derhami, Vali, 2010. Intrusion detection and attack classification using feed-forward neural network. In: 2010 Second International Conference on Computer and Network Technology. IEEE, pp. 262–266.

Hallman, Roger, Bryan, Josiah, Palavicini, Geancarlo, Divita, Joseph, Romero-Mariona, Jose, 2017. IoDDoS-the Internet of distributed denial of sevice attacks. In: 2nd International Conference on Internet of Things, Big Data and Security. SCITEPRESS. pp. 47–58.

Hameed, Sufian, Ali, Usman, 2018. HADEC: Hadoop-based live DDoS detection framework. EURASIP J. Inf. Secur. 2018 (1), 1–19.

Hameed, Sufian, Shah, Syed Attique, Saeed, Qazi Sarmad, Siddiqui, Shahbaz, Ali, Ihsan, Vedeshin, Anton, Draheim, Dirk, 2021. A scalable key and trust management solution for IoT sensors using SDN and blockchain technology. IEEE Sens. J. 21 (6), 8716–8733.

Hamidouche, Ranida, Aliouat, Zibouda, Ari, Ado Adamou Abba, Gueroui, Mourad, 2019. An efficient clustering strategy avoiding buffer overflow in IoT sensors: a bio-inspired based approach. IEEE Access 7, 156733–156751.

Hofstede, Rick, Celeda, Pavel, Trammell, Brian, Drago, Idilio, Sadre, Ramin, Sperotto, Anna, Pras, Aiko, 2014. Flow monitoring explained: From packet capture to data analysis with NetFlow and IPFIX. IEEE Commun. Surv. Tutor. 16 (4), 2037–2064.

Idhammad, Mohamed, Afdel, Karim, Belouch, Mustapha, 2018. Semi-supervised machine learning approach for DDoS detection. Appl. Intell. 48 (10), 3193–3208.

Jiang, Jehn-Ruey, Huang, Hsin-Wen, Liao, Ji-Hau, Chen, Szu-Yuan, 2014. Extending Dijkstra's shortest path algorithm for software defined networking. In: The 16th Asia-Pacific Network Operations and Management Symposium. IEEE, pp. 1–4.

Karagiannis, Dimitrios, Argyriou, Antonios, 2018. Jamming attack detection in a pair of RF communicating vehicles using unsupervised machine learning. Veh. Commun. 13, 56–63.

Khalid, Mizna, Hameed, Sufian, Qadir, Abdul, Shah, Syed Attique, Draheim, Dirk, 2023. Towards SDN-based smart contract solution for IoT access control. Comput. Commun. 198, 1–31.

Lee, Keunsoo, Kim, Juhyun, Kwon, Ki Hoon, Han, Younggoo, Kim, Sehun, 2008. DDoS attack detection method using cluster analysis. Expert Syst. Appl. 34 (3), 1659–1665.

Meidan, Yair, Bohadana, Michael, Mathov, Yael, Mirsky, Yisroel, Shabtai, Asaf, Breitenbacher, Dominik, Elovici, Yuval, 2018. N-BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders. IEEE Pervasive Comput. 17 (3), 12–22. http://dx.doi.org/10.1109/MPRV.2018.03367731.

Mohammadi, Reza, Conti, Mauro, Lal, Chhagan, Kulhari, Satish C, 2019. SYN-Guard: An effective counter for SYN flooding attack in software-defined networking. Int. J. Commun. Syst. 32 (17), e4061.

Nagtilak, Saraswati, Rai, Sunil, Kale, Rohini, 2020. Internet of things: A survey on distributed attack detection using deep learning approach. In: Proceeding of International Conference on Computational Science and Applications. Springer, pp. 157–165.

Nguyen, Hoai-Vu, Choi, Yongsun, 2010. Proactive detection of DDoS attacks utilizing k-NN classifier in an anti-DDoS framework. Int. J. Electr. Comput. Syst. Eng. 4 (4), 247–252.

Österlind, F., Dunkels, A., Eriksson, J., Finne, N., Voigt, T., 2006. Cross-level sensor network simulation with COOJA. In: Proceedings of LCN'2006 – the 31st IEEE Conference on Local Computer Networks. pp. 641–648.

2021. Over 100 million IoT attacks detected in 1H 2019 – Infosecurity Magazine. https://www.infosecurity-magazine.com/news/over-100-million-iot-attacks/ (Accessed on 13/01/2021).

Pande, Sagar, Khamparia, Aditya, Gupta, Deepak, Thanh, Dang NH, 2020. DDoS detection using machine learning technique. In: Recent Studies on Computational Intelligence. Springer, pp. 59–68.

Patil, Nilesh Vishwasrao, Krishna, C Rama, Kumar, Krishan, Behal, Sunny, 2019. E-Had: A distributed and collaborative detection framework for early detection of DDoS attacks. J. King Saud Univ.-Comput. Inf. Sci..

Rai, Ankur, Challa, Rama Krishna, 2016. Survey on recent DDoS mitigation techniques and comparative analysis. In: 2016 Second International Conference on Computational Intelligence Communication Technology (CICT). pp. 96–101. http://dx.doi.org/10.1109/CICT.2016.27.

Raikar, Meenaxi M., Meena, S.M., 2021. SSH brute force attack mitigation in internet of things (IoT) network : An edge device security measure. In: 2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC). pp. 72–77. http://dx.doi.org/10.1109/ICSCCC51823.2021.9478131.

Sahay, Rishikesh, Blanc, Gregory, Zhang, Zonghua, Debar, Hervé, 2015. Towards autonomic DDoS mitigation using software defined networking. In: SENT 2015: NDSS Workshop on Security of Emerging Networking Technologies. Internet society.

Sahoo, Kshira Sagar, Tripathy, Bata Krishna, Naik, Kshirasagar, Ramasubbareddy, Somula, Balusamy, Balamurugan, Khari, Manju, Burgos, Daniel, 2020. An evolutionary SVM model for DDOS attack detection in software defined networks. IEEE Access 8, 132502–132513.

2021. SDN-WISE core. https://sdnwiselab.github.io/docs/guides/Core.html (Accessed on 07/28/2021).

Shalimov, Alexander, Zuikov, Dmitry, Zimarina, Daria, Pashkov, Vasily, Smeliansky, Ruslan, 2013. Advanced study of SDN/OpenFlow controllers. In: Proceedings of the 9th Central & Eastern European Software Engineering Conference in Russia. pp. 1–6.

Shen, Limin, Li, Hui, Wang, Hongyi, Wang, Yihuan, 2020. Multifeature-based behavior of privilege escalation attack detection method for android applications. Mob. Inf. Syst. 2020.

Siddiqui, Shahbaz, Hameed, Sufian, Shah, Syed Attique, Ahmad, Ijaz, Aneiba, Adel, Draheim, Dirk, Dustdar, Schahram, 2022. Toward software-defined networking-based IoT frameworks: A systematic literature review, taxonomy, open challenges and prospects. IEEE Access 10, 70850–70901. http://dx.doi.org/10.1109/ACCESS.2022.3188311.

Siddiqui, Shahbaz, Hameed, Sufian, Shah, Syed Attique, Khan, Abdul Kareem, Aneiba, Adel, 2023. Smart contract-based security architecture for collaborative services in municipal smart cities. J. Syst. Archit. 135, 102802.

Silveira, Frederico Augusto Fernandes, Lima-Filho, Francisco, Silva, Felipe Sampaio Dantas, Junior, Agostinho de Medeiros Brito, Silveira, Luiz Felipe, 2020. Smart detection-IoT: A DDoS sensor system for Internet of Things. In: 2020 International Conference on Systems, Signals and Image Processing (IWSSIP). IEEE, pp. 343–348.

Snehi, Manish, Bhandari, Abhinav, 2021. Vulnerability retrospection of security solutions for software-defined Cyber–Physical System against DDoS and IoT-DDoS attacks. Comp. Sci. Rev. 40, 100371.

2020. Software-defined networking (SDN) definition - open networking foundation. https://www.opennetworking.org/sdn-definition/ (Accessed on 10/06/2020).

Suresh, Manjula, Anitha, R., 2011. Evaluating machine learning algorithms for detecting DDoS attacks. In: International Conference on Network Security and Applications. Springer, pp. 441–452.

Taylor, R., Baron, D., Schmidt, D., 2015. The world in 2025 – predictions for the next ten years. In: 2015 10th International Microsystems, Packaging, Assembly and Circuits Technology Conference (IMPACT). pp. 192–195. http://dx.doi.org/10.1109/IMPACT.2015.7365193.

Tayyab, Mohammad, Belaton, Bahari, Anbar, Mohammed, 2020. ICMPv6-based DoS and DDoS attacks detection using machine learning techniques, open challenges, and blockchain applicability: A review. IEEE Access 8, 170529–170547.

Ujjan, Raja Majid Ali, Pervez, Zeeshan, Dahal, Keshav, Bashir, Ali Kashif, Mumtaz, Rao, González, J, 2020. Towards sFlow and adaptive polling sampling for deep learning based DDoS detection in SDN. Future Gener. Comput. Syst. 111, 763–779.

Van Adrichem, Niels L.M., Doerr, Christian, Kuipers, Fernando A., 2014. Opennetmon: Network monitoring in openflow software-defined networks. In: 2014 IEEE Network Operations and Management Symposium (NOMS). IEEE, pp. 1–8.

Verma, Abhishek, Ranga, Virender, 2020. Machine learning based intrusion detection systems for IoT applications. Wirel. Pers. Commun. 111 (4), 2287–2310.

Wang, Jingjing, Jiang, Chunxiao, Zhang, Haijun, Ren, Yong, Chen, Kwang-Cheng, Hanzo, Lajos, 2020. Thirty years of machine learning: The road to Pareto-optimal wireless networks. IEEE Commun. Surv. Tutor. 22 (3), 1472–1514.

Witten, Ian H, Frank, Eibe, Hall, Mark A, Pal, Christopher J, Data, Mining, 2005. Practical machine learning tools and techniques. In: Data Mining, Vol. 2.

Xiao, Liang, Wan, Xiaoyue, Lu, Xiaozhen, Zhang, Yanyong, Wu, Di, 2018. IoT security techniques based on machine learning: How do IoT devices use AI to enhance security? IEEE Signal Process. Mag. 35 (5), 41–49.

Xie, Junfeng, Yu, F Richard, Huang, Tao, Xie, Renchao, Liu, Jiang, Wang, Chenmeng, Liu, Yunjie, 2018. A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges. IEEE Commun. Surv. Tutor. 21 (1), 393–430.

Xu, Tu, He, Dake, Luo, Yu, 2007. DDoS attack detection based on RLT features. In: 2007 International Conference on Computational Intelligence and Security (CIS 2007). IEEE, pp. 697–701.

Yaqoob, Ibrar, Hashem, Ibrahim Abaker Targio, Ahmed, Arif, Kazmi, SM Ahsan, Hong, Choong Seon, 2019. Internet of things forensics: Recent advances, taxonomy, requirements, and open challenges. Future Gener. Comput. Syst. 92, 265–275.

Yin, Da, Zhang, Lianming, Yang, Kun, 2018. A DDoS attack detection and mitigation with software-defined Internet of things framework. IEEE Access 6, 24694–24705.

Yuan, Jian, Mills, Kevin, 2005. Monitoring the macroscopic effect of DDoS flooding attacks. IEEE Trans. Dependable Secure Comput. 2 (4), 324–335.

Zhang, Yuntong, Xu, Jingye, Wang, Zhiwei, Geng, Rong, Choo, Kim-Kwang Raymond, Pérez-Díaz, Jesús Arturo, Zhu, Dakai, 2020. Efficient and intelligent attack detection in software defined IoT networks. In: 2020 IEEE International Conference on Embedded Software and Systems (ICESS). pp. 1–9. http://dx.doi.org/10.1109/ICESS49830.2020.9301591.

Jalal Bhayo received the Ph.D. degree in Computer Science from the National University of Computer and Emerging Science (NUECS-FAST), Pakistan (Karachi Campus). He was working in IT Security lab at NUECS. His research interest areas include IoT, network security, web security, and SDN applications in security. He has industrial expertise in different networking-related products and also received Cisco certification. He is working as an Assistant Professor in Computer Science for the CED, Government of Sindh.

Syed Attique Shah received the Ph.D. degree from the Institute of Informatics, Istanbul Technical University, Istanbul, Turkey. During his Ph.D., he studied as a Visiting Scholar at the University of Tokyo, Japan, the National Chiao Tung University, Taiwan, and the Tallinn University of Technology, Estonia, where he completed the major content of his thesis. He has worked as an Associate Professor and the Chairperson at the Department of Computer Science, BUITEMS, Quetta, Pakistan. He was also engaged as a Lecturer at the Data Systems Group, Institute of Computer Science, University of Tartu, Estonia. Currently, he is working as a Lecturer in Smart Computer Systems, at the School of Computing and Digital Technology, Birmingham City University, United Kingdom. He is a Senior Member, IEEE. His research interests include big data analytics, the Internet of Things, machine learning, network security, and information management.

Sufian Hameed received the Ph.D. degree in the field of networks and information security, from University of Göttingen, Germany. He is an Associate Professor in the Department of Computer Science, National University of Computer and Emerging Sciences, Pakistan. He also leads the IT Security Labs at NUCES. The research lab studies and teaches security problems and solutions for different types of information and communication paradigms. His research interests include network security, web security, mobile security and secure architectures, and protocols for cloud and the IoTs.

Awais Ahmed completed his Bachelor's as well as Master's in Computer Science from FAST-NUCES in 2016 & 2019 respectively. He is currently a full-time Ph.D. Scholar at UESTC - University of Electronic Science and Technology of China and is on study leave from his position as a Lecturer in the Department of Computer Science at Muhammad Ali Jinnah University, Karachi. Previously, he worked at NUCES-FAST as a Research Associate and Instructor. His field of interest includes Big Data Healthcare, Multimodal Data, Data Science, Machine Learning, Natural Language Processing, Network Security, and Analytics.

Jamal Nasir completed his Bachelor's in computer science in 2011 from The National University of Computer and Emerging Sciences, (NUCES) Karachi Campus. (also commonly known as "Foundation for Advancement of Science and Technology" – FAST). While gaining experience in the field of IT and Software Development he pursues his Master's in Computer network and security (CNS) from NUCES. He is currently working as a Senior Software Engineer in a Multinational Software House.

Dirk Draheim received a Ph.D. from Freie Universität Berlin and a habilitation from Universität Mannheim, Germany. From 2006–2008, he was area manager for database systems at the Software Competence Center Hagenberg, Austria. From 2008–2016 he was head of the data center of the University of Innsbruck and, in parallel, Adjunct Reader at the Faculty of Information Systems of the University of Mannheim. Currently, he is Full Professor of Information Systems at Tallinn University of Technology (Taltech), Estonia, and heading the Taltech Information Systems Group. The Taltech Information Systems Group conducts research in large- and ultra-large-scale IT systems, in particular, next generation of digital government technologies and digital government ecosystems. Dirk is co-author of the Springer book "Form-Oriented Analysis" and author of the Springer books "Business Process Technology", "Semantics of the Probabilistic Typed Lambda Calculus" and "Generalized Jeffrey Conditionalization". He is also an initiator and a leader of numerous digital transformation initiatives.