electronics

Article

An Anomaly Mitigation Framework for IoT Using Fog Computing

Muhammad Aminu Lawal *, Riaz Ahmed Shaikh and Syed Raheel Hassan

Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; rashaikh@kau.edu.sa (R.A.S.); rhassan1@kau.edu.sa (S.R.H.)
* Correspondence: mlawal@stu.kau.edu.sa

Received: 10 August 2020; Accepted: 22 September 2020; Published: 24 September 2020

Abstract: The advancement in IoT has prompted its application in areas such as smart homes, smart cities, etc., and this has aided its exponential growth. However, alongside this development, IoT networks are experiencing a rise in security challenges such as botnet attacks, which often appear as network anomalies. Similarly, providing security solutions has been challenging due to the low resources that characterize the devices in IoT networks. To overcome these challenges, the fog computing paradigm has provided an enabling environment that offers additional resources for deploying security solutions such as anomaly mitigation schemes. In this paper, we propose a hybrid anomaly mitigation framework for IoT using fog computing to ensure faster and accurate anomaly detection. The framework employs signature- and anomaly-based detection methodologies for its two modules, respectively. The signature-based module utilizes a database of attack sources (blacklisted IP addresses) to ensure faster detection when attacks are executed from the blacklisted IP address, while the anomaly-based module uses an extreme gradient boosting algorithm for accurate classification of network traffic flow into normal or abnormal. We evaluated the performance of both modules using an IoT-based dataset in terms response time for the signature-based module and accuracy in binary and multiclass classification for the anomaly-based module. The results show that the signature-based module achieves a fast attack detection of at least six times faster than the anomaly-based module in each number of instances evaluated. The anomaly-based module using the XGBoost classifier detects attacks with an accuracy of 99% and at least 97% for average recall, average precision, and average F1 score for binary and multiclass classification. Additionally, it recorded 0.05 in terms of false-positive rates.

Keywords: anomaly mitigation; internet of things (IoT); Intrusion Detection System (IDS); fog computing; classification algorithms

1. Introduction

In recent times, the proliferation of IoT devices and their applications in various facets of our lives, such as smart cities, smart health, smart homes, etc., has provided numerous benefits. IoT networks are experiencing tremendous growth, with the expected number of these devices to reach around 50 billion at the end of 2020 [1]. This growth comes with a lot of challenges. On the one hand, the main challenge is the security of these connected devices, which are increasingly under attacks. On the other hand, there is a lack of adequate resources (storage and computational) that characterizes IoT devices, which are essential for deploying security solutions such as network anomaly mitigation, usually performed by Intrusion Detection Systems (IDS) on IoT networks [2].

The security challenges in IoT networks usually come as network anomalies, specifically when there is a deviation from the flow of normal network traffic. Examples of such abnormal network traffic flow are Distributed Denial of Service (DDoS) attack and Probing attacks [3]. These attacks are usually driven by a botnet and represent the common types of anomalies that exist in IoT networks. A botnet comprises a large number of hijacked nodes or systems in a network that are controlled by malicious users remotely. These nodes or systems are used to execute several types of attacks [4]. A botnet attack is usually characterized by three features, which are similarity of attack sources, divergence between normal and attack network traffic flow, and automation of attack execution [5].

The Mirai botnet attack remains as one of the popular attacks on IoT networks. The Mirai botnets have evolved over the years [6]; a recent example of attacks using a Mirai variant was recorded between March and April of 2019 on an entertainment industry that provides online streaming services. It utilized over 400,000 compromised IoT devices to execute the attacks. The botnet was able to generate around 292,000 requests per second and it lasted for 13 days [7]. This highlights the weakness and threat to the nodes in IoT networks, although the anomaly mitigation scheme cannot detect the exploitation of default authentication credentials by the Mirai malware at the device level. This necessitates the deployment of anomaly mitigation schemes as a vital part of the defense procedures to protect against the utilization of a large number of devices in IoT networks for execution of botnet attacks.

To protect and ensure efficient operation of IoT devices on the network, the fog computing paradigm can be employed to ameliorate the lack of resources required in the operation of the anomaly mitigation schemes in the IoT networks. Fog computing is conceived to ease computational, storage, and latency as well as energy consumption needs by bringing these resources to the edge of the network [8]. In this way, IoT applications and devices can receive a better and faster response as well as relief from performing operations that will stretch their resources and reduce their efficiency.

In this paper, we proposed an anomaly mitigation framework that leverages the benefits of the fog to deploy a hybrid anomaly mitigation framework for the IoT network. Firstly, it employs the signature-based IDS that utilizes the similarity feature of attack sources in botnet attacks to create a blacklist of attack sources (IP addresses) for timely attack detection. Secondly, it uses an anomaly-based IDS that utilizes an extreme boosting classifier to ensure attack detection with high accuracy and low false-positive rates. The utilization of the signature-based module in the framework for network traffic flow analyses ensures a speedy detection of known attack sources, thereby reducing the operational overhead and time of classification in the anomaly-based IDS module.

The contributions of this paper are explained below:

- We proposed a hybrid anomaly mitigation framework for IoT networks using fog computing, which harnesses the resources of the fog. The framework employs signature-based and anomaly-based modules to ensure faster and accurate attack detection.
- In the proposed framework, we have used a blacklist of IP addresses in our signature-based module and an Extreme Gradient Boosting (XGBoost) [9] classifier for the anomaly-based module due to its resilience against overfitting. The performance of the blacklist look-up and XGBoost classifier was evaluated in terms of response time and classification accuracy (binary and multiclass) using the BoT-IoT dataset, respectively. The results show that the signature-based module detects attacks fast. Similarly, the XGBoost classifier has a superior performance over other classifier algorithms such as Decision Tree (DT) [10], k Nearest Neighbors (k-NN) [11], and Naïve Bayes (NB) [12].
- A review and comparison of some proposed state of the art anomaly mitigation schemes in IoT are provided. The review covers the objectives, operational procedures as well as the strength of each scheme, while the comparison is in terms of detection methodology, techniques employed, evaluation strategy used, attacks detected, and dataset utilized for evaluations.

The remainder of the paper is organized as follows. Section 2 discusses a brief background of IoT, IDS, and fog computing. Section 3 provides the related work on anomaly mitigation schemes in IoT. Section 4 presents the proposed framework for anomaly mitigation in IoT using fog computing. Section 5 presents the performance evaluation and finally, Section 6 concludes the paper.

2. Background

This section gives a brief overview of the IoT concept, fog computing, and intrusion detection systems as well as its categories.

2.1. Internet of Things (IoT)

The IoT in simple terms can be defined as a network of physical objects (things/devices) that can communicate with each other using the internet. The development of the Auto-ID Centre at the Massachusetts Institute of Technology (MIT) in 1999 and utilization of radio frequency identification (RFID) for developing an Electronic Product Code (EPC) in 2003 serve as the foundation in the IoT voyage [13]. The definition and description of IoT have been provided by many organizations and researchers as highlighted in [14,15].

The IoT commonly adopts the generic three-tier architecture, namely application layer, communication (network and transport) layer, and physical (perception) layer. This is due to the unavailability of a standard architecture [16].

The physical (perception) layer collects data from its surrounding environment by utilizing devices such as RFID and sensors, and communication standards, for example, IEEE 802.15.4 and Bluetooth. These standards are short-range and support limited data rates. The communication (network and transport) layer transmits the acquired data from the physical layer upward to the application layer. It employs communication standards that are characterized by long-range capabilities such as IEEE 802.11, 4G, IEEE 802.3, etc. The application layer processes received data to obtain information that could be utilized by devices or applications in making decisions. In addition, a middleware is employed at this layer to make communication among applications and devices smooth [13,17]. Several protocols and standards have been proposed for IoT; a comprehensive survey of protocols as well as standards is provided in [18].

2.2. Fog Computing

Fog computing is a distributed computing paradigm conceived by Cisco [19]; it moves application services, storage, computation as well as data close to the users at the edge of the network. This helps provide a fast response to applications by decreasing latency and bandwidth usage. In addition, the fog offers scalability and availability through its deployment. These characteristics of fog computing are well-positioned to provide needed aid to IoT devices, which are characterized by lack of enough resources for storage and computations [20]. An extensive survey about fog computing and its characteristics can be found in [21].

For deployment in the IoT network, fog computing adopts the three-layered architecture [8], which consists of the cloud layer, fog nodes layer, and the IoT device layer as shown in Figure 1. The fog nodes layer is positioned between the cloud layer and the IoT devices layer. It comprises nodes such as routers or gateways, switches, base stations, servers or dedicated computer systems, etc. It serves as a complement to the cloud layer by performing the required computations, storage, and other services traditionally done by the cloud. It receives data from the IoT devices in the IoT devices layer and performs the processes required by users without using the cloud. The cloud layer comprises high-end servers which host different IoT applications. It serves as the universal manager of the applications.

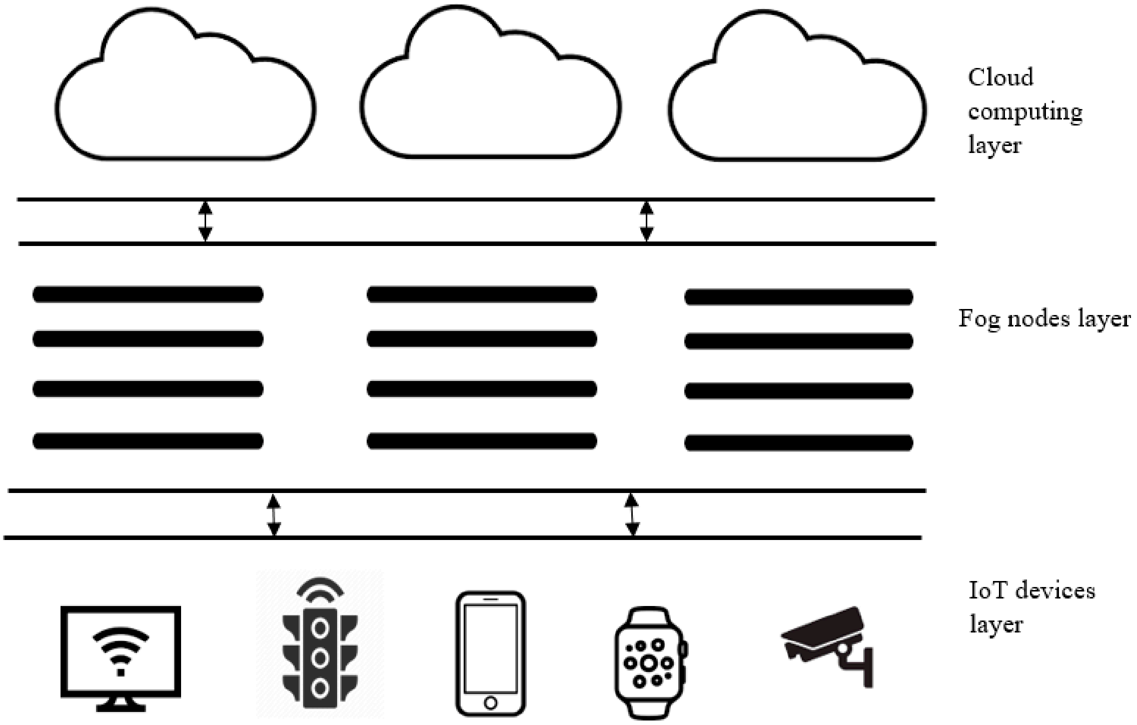

Figure 1. Fog computing architecture.

Consequently, the fog computing paradigm can be employed in the IoT to effectively host anomaly mitigation frameworks. This will allow effective attack detection close to the IoT devices. In addition, it will ensure efficient operation of the IoT devices by relieving the operational requirements of the anomaly mitigation solution on the IoT devices.

2.3. Intrusion Detection System (IDS)

The IDS is employed to detect unauthorized access into a network or any system. It is widely deployed in two ways; firstly, at the Host level (HIDS) on a node to monitor its system activities on its system application files or the operating system running on the node. At this level, a node can be a computer system or device in IoT. Secondly, at the Network level (NIDS) on a gateway or border router, where it monitors network traffic flows [22].

The NIDS is categorized according to the method of detection and deployment architecture. In terms of deployment architecture, the NIDS can utilize a centralized, distributed, or hybrid deployment strategy [23]. In centralized deployment, the NIDS is positioned on a dedicated host or a router. It monitors the network traffic flow and transactions between the inside of its network and the internet. In the distributed deployment architecture, the NIDS is placed on each network node where nodes monitor each other's network transactions. The hybrid deployment employs both centralized and distributed architectures to leverage the benefits and reduce the shortcomings of the deployment strategies [4].

In terms of method of detection, the NIDS is widely classified as signature-based, anomaly-based, and hybrid-based. The signature-based IDS detects threats or attacks through matching stored attack signatures or rules with network traffic flow features. It successfully detects known attacks with 100% accuracy. The anomaly-based schemes employ statistical, machine, or protocol-specific information to build a model of legitimate network traffic flow as a reference point for its operation. It detects an attack by comparing network traffic flow with the model; a difference with the model translates to a threat or attack (abnormal traffic). The anomaly-based IDS has the capability of detecting unknown or zero-day attacks [24]. Similar to the deployment architecture strategy, the hybrid IDS fuses both detection methods to gain the advantages and decrease the disadvantages of the detection methods [25].

The signature-based IDS performs well in the detection of known attacks due to its operational procedure. However, it fails in the detection of unknown attacks and is unsuitable for the resource-constrained IoT. The anomaly-based IDS can detect unknown attacks, even though it can be suitable for IoT. It suffers from false alarms when normal traffic is classified as abnormal traffic [26].

3. Related Work

With the growth of IoT networks, several researchers have proposed anomaly mitigation schemes to protect the IoT from malicious users. As mentioned earlier, the IDS schemes can largely be classified into signature-based [27–29], anomaly-based [30–36], and hybrid-based [22,36]. Additionally, some of the anomaly mitigations' schemes [35–38] utilize the fog computing paradigm for deployment. This section presents a review and comparison of some proposed anomaly mitigation schemes in IoT networks.

An IDS scheme based on Raspberry Pi (RPiIDS) [27] is proposed for IoT. The RPiIDS scheme utilizes an open-source signature-based IDS called Snort. The full Snort IDS was installed on the Raspberry Pi. Experimental results showed that the Snort IDS can be hosted on the Raspberry Pi.

A signature-based IDS [28] is proposed to detect DDoS attacks in IoT networks. It comprises two units, namely IDS detectors and IDS routers, which are fused in a hybrid deployment. The IDS router is hosted in the border gateway and it performs firewall and detection functionalities. The IDS detectors employ sensors that monitor the internal traffic, i.e., behind the gateway. It sends information about malicious devices to the gateway for necessary action. The results showed that the scheme detects version number modification and hello flooding attacks.

A Denial-of-Service (DoS) detection scheme [29] is proposed for 6LoWPAN-based IoT networks. It utilizes the ebbits network framework and Suricata IDS. The adopted IDS uses packet threshold rules for the detection of network anomalies such as DoS. The DoS detection scheme employs probes connected to the IDS to capture network packets for inspection. It effectively detects UDP flooding.

Although it was shown that a signature-based IDS can be hosted on a Raspberry Pi, the schemes will not be able to detect zero-day attacks. In addition, the operational requirements are not suited for IoT devices. To deploy signature-based IDS, additional measures that will reduce the burden of storage and computation on the IoT devices are needed.

A two-level hybrid model for an anomalous activity detection scheme [30] is proposed to detect intrusion in IoT networks. This scheme consists of two phases for anomaly detection and identification of attack category. The first phase employs flow-based features and the decision tree classifier to classify normal and abnormal traffic. The second phase receives the anomalous traffic from the first phase and utilizes Recursive Feature Elimination (RFE) to select relevant features. It also uses Synthetic Minority Over-Sampling Technique (SMOTE) and Edited Nearest Neighbors (ENN) to deal with oversampled and under-sampled instances for training. Finally, it employs the random forest classifier to categorize the detected abnormal traffic according to attack types. The two-level hybrid scheme achieves satisfactory results in terms of recall, precision, F1 score, and specificity.

A lightweight IDS scheme [31] is proposed for IoT. This scheme consists of a training level and evaluation level. At the training phase, the scheme uses features obtained from the packet inter-arrival time of the received data to train the scheme to make the system lightweight. At the evaluation stage, the scheme employs the support vector machine (SVM) classifier to detect an intrusion or abnormal traffic. The lightweight IDS scheme achieves an acceptable result in terms of detection time and classification accuracy.

A real-time IDS scheme [32] is proposed to detect wormhole attacks in RPL-based IoT. It utilizes the routing information and Received Signal Strength Indicator (RSSI) to detect malicious users and nodes. The real-time IDS scheme is evaluated in centralized and distributed deployments. It achieves a detection rate of 90%.

A supervised IDS scheme [33] is proposed to detect attacks on smart home IoT devices. This scheme utilizes the Packet Description Markup Language (PDML) during preprocessing for feature selection.

This scheme employs the decision tree classifier and comprises three stages. At the first stage, the scheme identifies the IoT node based on its MAC address and groups them according to their activities. The second stage detects anomalous traffic flow and sends the abnormal traffic flow to the third stage. The third stage categorizes the abnormal traffic flow according to the attack type. The supervised IDS scheme was successful in detecting DoS, Man-In-The-Middle (MITM)/Spoofing, Reconnaissance, and Replay attacks. It also achieves a good F-measure in terms of device profiling, attack detection, and attack categorization.

A Deep Learning Intrusion Detection System (DL-IDS) scheme [34] is proposed to detect security threats in IoT. The DL-IDS employs Minkowski distance and k Nearest Neighbor to generate missing instances in the dataset at the preprocessing stage. It utilizes the Spider Monkey Optimization (SMO) algorithm for feature selection and Stacked-Deep Polynomial Network (SDPN) for distinguishing normal and abnormal instances. The DL-IDS scheme achieves satisfactory results in terms of accuracy, precision, recall, and F1 score.

An Ensemble Learning-based Network Intrusion Detection System (ELNIDS) scheme [35] is proposed to detect routing attacks in RPL-based IoT. This scheme employs ensemble machine learning classifiers (Boosted trees, Bagged Trees, RUSBoosted Trees, and Subspace Discriminant) to classify network traffic flow into normal or abnormal. This scheme was able to detect Blackhole, Sinkhole, Sybil, Selective Forwarding, Clone ID, Hello Flooding, and Local Repair attacks with good accuracy and Receiver Operating Characteristics (ROC) area. In addition, the Boosted Trees and RUSBoosted Trees classifiers achieved a better performance among other evaluated ensemble techniques.

A Real-time IDS scheme (SVELTE) [22] is proposed for IoT. This scheme is placed on the border router and comprises three units: 6LoWPAN Mapper (6Mapper), IDS component, and a distributed mini-firewall. The 6Mapper captures information about each node. The information includes node ID, node rank, parent ID, and all neighbor IDs and ranks. It uses the information and response from the mapping request to replicate the network structure on the router. The replicated network is utilized in detecting routing inconsistencies. The IDS component contains techniques to detect selective forwarding attacks, spoofed or altered information, and sinkhole attacks. The distributed mini-firewall protects the IoT network from external malicious users. The SVELTE scheme detects sinkhole and/or selective forwarding attacks and their sources with low energy consumption and overhead.

A Compression Header Analyzer Intrusion Detection Scheme (CHA-IDS) [36] is proposed for 6LoWPAN-based IoT networks. The CHA-IDS scheme is positioned on the router and utilizes anomaly- and signature-based detection methodologies. This scheme comprises four units: Sensor Agents (SA), Aggregator Agent (AGA), Analyzer Agent (ANA), and Actuator Agent (ACA). The SA collects network traffic flow packets from all nodes. The AGA utilizes the best-first search, greedy stepwise, and correlation-based feature selection algorithms to identify the features that will be used for the classification of network traffic. The ANA performs the classification of the traffic flow into normal or abnormal by using the information from AGA. Lastly, the ACA sends an alert to the admin in the event of attack detection. The scheme employs machine learning to distinguish types of attacks by using the compression header features and generates rules or signatures that are updated in the signature-based unit. These rules are used in identifying different attacks. The CHA-IDS scheme detects wormhole attacks, hello flooding attacks, and sinkhole attacks or a combination of the attacks.

The major advantage of anomaly-based schemes is the ability to detect zero-day attacks. However, the schemes in [22,29] are protocol specific, which means the schemes will not be able to work on all IoT devices. Similarly, the anomaly-based schemes in [20,30–39] employed machine learning or statistical techniques but may experience false-positive rates when wrong decisions about normal traffic flow are made. The hybrid schemes in [22,36] leverage the benefits of the combination of different detection methodologies. However, the schemes may inherit the problems of the adopted methods.

To solve the challenge of lack of enough resources in deploying the anomaly mitigation schemes as well as improve the services of IoT applications, the authors in [20,37–39] employed fog computing in anomaly mitigation for IoT networks.

An Anomaly Detection for IoT (AD-IoT) scheme [37] is proposed to detect cybersecurity threats in a smart city environment. This scheme is deployed on distributed fog nodes and employs the random forest (RF) classifier for classification of network traffic flow into normal or abnormal. The results obtained show that the RF classifier achieved satisfactory accuracy in attack detection with a low false-positive rate.

An anomaly detection scheme for IoT empowered by fog computing [38] is proposed to utilize the resources on the fog to ensure fast anomaly detection. The scheme employs the Hyper Ellipsoidal Clustering for Resource-Constrained Environments algorithm (HyCARCE) to cluster the data collected from the sensor nodes and Ellipsoidal neighborhood outlier factor (ENOF) to distinguish normal and abnormal clusters. This scheme is comprised of four phases: HyCARCE clustering, ENOF computation, cluster information transmission, and an anomaly detection process, which are all performed on the fog. The fog empowered anomaly detection scheme detects anomalies promptly with minimal overhead and low energy consumption.

A distributed attack detection scheme [39] is proposed for the IoT. This scheme utilizes a Deep Learning (DL) method for attack detection. It uses the fog level nodes' resources to train the DL models as well as host the detection schemes. The distributed attack detection scheme employs a coordinator (master node) to improve cooperation between the fog nodes in terms of optimization and parameter sharing. The results show that the scheme has superior performance than the centralized scheme in detecting attacks.

A semi-supervised learning-based distributed attack detection framework [20] is proposed for attack detection in IoT networks. This scheme utilizes fog devices and employs an ELM-based Semi-Supervised Fuzzy C-Means (ESFCM) technique, which combines the Extreme Learning Machine (ELM) and Semi-Supervised Fuzzy C-Means (SFCM) algorithm. It uses the ELM to train the classifier model and SFCM algorithm for clustering unlabeled data. This scheme achieves good performance with an accuracy of 86.53% and a low detection time.

Fog computing-based schemes [20,37–39] employed anomaly-based detection methodology using machine learning or statistical techniques, which means the traffic will always be scanned to check if it is normal or abnormal. However, these schemes fail to utilize the full potential of the fog nodes in terms of storage. Storing the signatures of the previously detected attacks will improve attack detection accuracy, reduce computational overheads, and further decrease the detection response time.

Table 1 presents the comparison of the anomaly mitigation schemes in IoT. The comparison is based on detection methodology, techniques employed, evaluation strategy used, attacks detected, and dataset utilized.

Table 1. Comparison of proposed anomaly mitigation schemes in IoT.

S/No	Scheme	Detection Methodology	Technique	Evaluation Strategy	Attacks Detected	Dataset
1	A signature-based intrusion detection system (2018) [28]	Signature-based	Routing protocol information	Simulation	i. Hello flooding ii. Version number modification	N/A
2	A Raspberry Pi Intrusion Detection System (RPiIDS) (2016) [27]	Signature-based	Snort	Experiments	Not specified	N/A
3	A Denial-of-Service detection scheme (ebbits, 2013) [29]	Signature-based	Suricata (Packet threshold rule)	Simulation	DoS	N/A
4	A two-level hybrid model for anomalous activity detection scheme [30]	Anomaly-based	i. Decision Tree ii. Random Forest iii. RFE iv. SMOTE	Experiment	Attacks in the dataset	i. UNSW-NB 15 ii.CICIDS2017

Table 1. *Cont.*

S/No	Scheme	Detection Methodology	Technique	Evaluation Strategy	Attacks Detected	Dataset
5	A lightweight IDS scheme [31]	Anomaly-based	SVM	i. Simulation ii. Experiments	DDoS	CICIDS2017
6	A real-time IDS scheme [32]	Anomaly-based	i. Routing information ii. RSSI	Simulation	Wormhole	N/A
7	A supervised IDS scheme [33]	Anomaly-based	Decision Tree	Experiments	i. DoS ii. MITM iii. Reconnaissance iv. Replay	Generated from the experiment testbed.
8	A Deep Learning Intrusion Detection System (DL-IDS) scheme [34]	Anomaly-based	i. k-NN ii. SMO iii. SPDN	Experiments	i. DoS ii. Probe iii.R2L iv.U2R	NSL KDD
9	An Ensemble Learning based Network Intrusion Detection System (ELNIDS) scheme [35]	Anomaly-based	Ensemble ML	Experiments	Routing attacks.	RPL-NIDDS17
10	SVELTE (2013) [22]	Hybrid	i. Routing protocol information ii. Signature based IDS	Simulation	i. Sinkhole ii. Selective forwarding attacks	N/A
11	CHA–IDS (2018) [36]	Hybrid	i. Best first search algorithm ii. Greedy stepwise algorithm iii. Correlation-based features selection iv. Signature-based IDS	Simulation	i. Hello flooding attack, ii. Sinkhole attack iii. Wormhole	N/A
12	An Anomaly Detection for IoT (AD-IoT) scheme [37]	Anomaly-based	Random Forest Algorithm	Experiment	Not specified	UNSW-NB 15
13	A distributed attack detection scheme (2018) [39]	Anomaly-based	Deep learning	Experiments	i. DoS ii. Probe iii.R2L iv.U2R	NSL KDD
14	A fog empowered anomaly detection scheme for IoT (2017) [38]	Anomaly-based	i. HyCARCE ii. ENOF	Experiments	Not specified	i. S12 ii. Banana iii. Melbourne IoT data iv. Intel Berkeley Research Laboratory IBRL(IBRL)
15	Semi-Supervised learning based distributed attack detection framework (2019) [20]	Anomaly-based	i. Extreme Learning Machine (ELM) and ii. Semi-Supervised Fuzzy C-Means (SFCM) algorithm	Experiments	i. DoS ii. Probe iii. R2L iv.U2R	NSL KDD

4. Proposed Framework

This section presents the proposed framework, a use case on the application of the fog computing framework and anomaly mitigation, and a brief description of the components employed in the framework.

Towards mitigating anomalies such as botnet attacks in the IoT network, a hybrid framework using the fog computing paradigm is proposed, as shown in Figure 2. The fog computing paradigm is utilized to compensate the lack of enough resources in IoT networks. It removes the burden of computational overhead and other related operational requirements in anomalies mitigation from the resource-constrained IoT devices. The framework employs two modules, which are signature-based and anomaly-based IDS modules, as shown in Figure 2. These modules utilize a database of blacklisted IP addresses and an extreme gradient boosting [9] classifier for the signature-based module and anomaly-based module, respectively. This is to enable the detection modules to leverage their strengths. The signature-based module will ensure 100% accuracy in the detection of known attacks through its source, while the anomaly-based will detect the zero-day attacks with satisfactory accuracy. Hence, the framework ensures a safe IoT network.

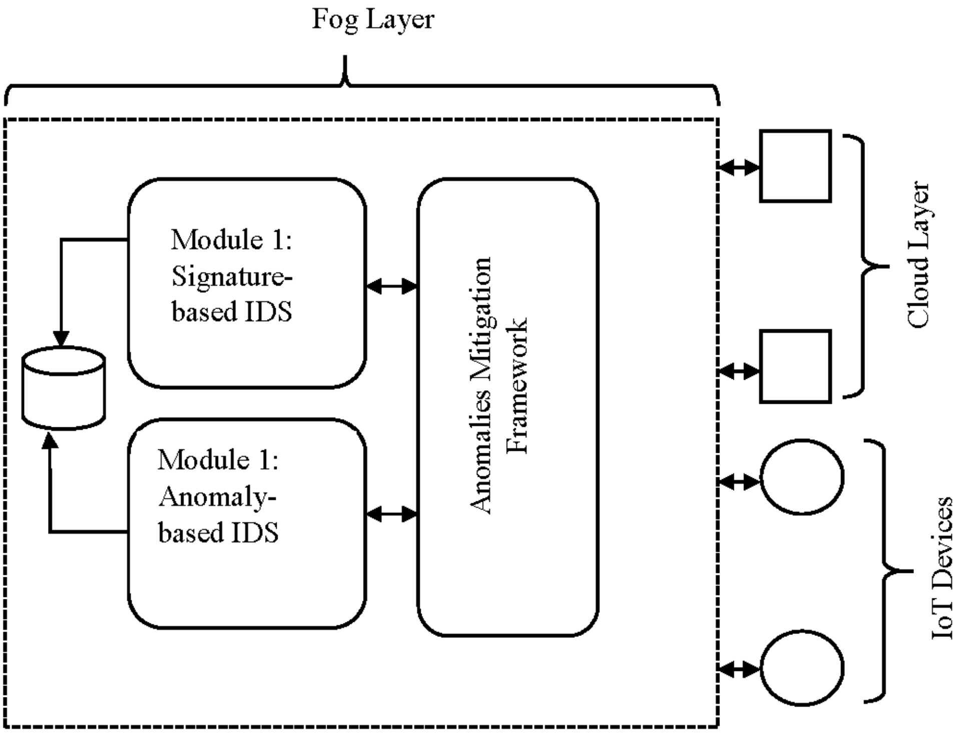
Fog Layer
Module 1: Signature-based IDS
Module 1: Anomaly-based IDS
Anomalies Mitigation Framework
Cloud Layer
IoT Devices

Figure 2. Proposed framework.

The network traffic flow is given as x, signature of known attacks (blacklisted IP addresses) as d, normal traffic flow as N, and abnormal traffic flow as A.

Firstly, the network traffic flow x passes through the first module, i.e., the signature-based detection module. The IP address of traffic flow x is scanned against d stored in the database of the module. If x ϵ d, then x is blocked/dropped, and an alert is generated and sent to the administrator. Otherwise, the flow is forwarded to the second module, which is the anomaly-based detection module.

The anomaly-based detection module classifies the network traffic flow x into N or A. If x is classified as A, the module blocks/drop the x. Then, an alert is generated and sent to the administrator. Finally, the signature (IP address) of A is updated in the database of the signature-based detection module. Otherwise, the x is allowed to pass. The operational flow of the framework is shown in Figure 3.

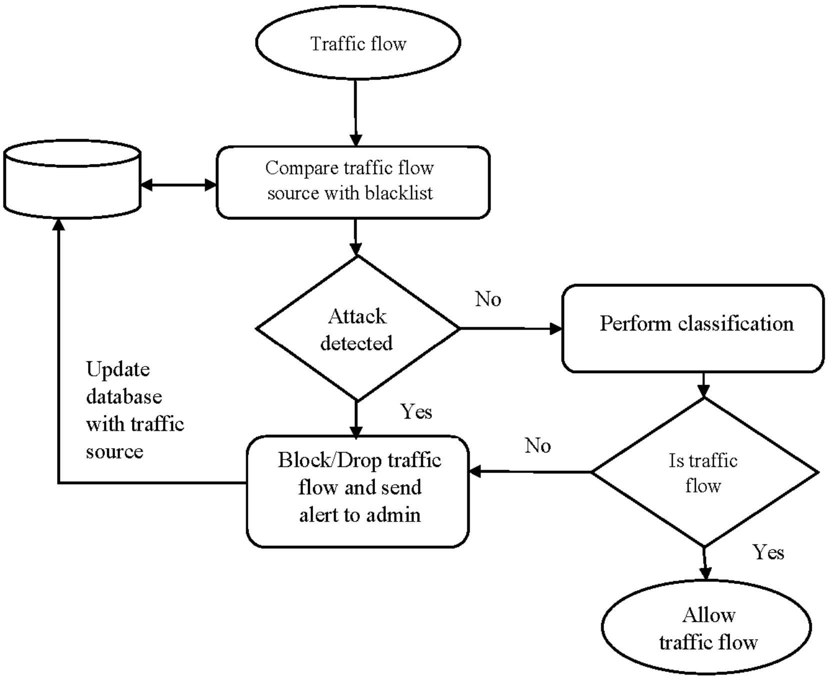
Traffic flow
Compare traffic flow source with blacklist
Attack detected
No
Perform classification
Update database with traffic source
Yes
Block/Drop traffic flow and send alert to admin
No
Is traffic flow
Yes
Allow traffic flow

Figure 3. Flow chart of the operation of the proposed framework.

4.1. Use Case on the Application of the Fog Computing Framework and the Anomaly Mitigation

Considering a smart building scenario, the proposed fog framework can be deployed on a dedicated system that could act as a fog node. It can be used to deploy security solutions such as anomaly mitigation systems (IDS) efficiently to safeguard the IoT nodes or systems, such as a smart garage, smart fridge, smart thermostat, or smart weather station. This will protect the IoT nodes from external attacks or being utilized by malicious users as part of a botnet used in executing DDoS attacks. This scenario corresponds to the scenario used in developing the dataset utilized for the evaluation of the anomaly-based module of our proposed framework.

Table 2 presents a comparison of the fog-based frameworks proposed for IoT and our proposed framework. The comparison is based on detection methodology used, techniques employed, and the dataset used for evaluations.

Table 2. Comparison of some fog-based schemes with the proposed framework.

S/No	Scheme	Detection Methodology	Techniques	Evaluated with IoT-Based Dataset
1	A distributed attack detection scheme (2018) [39]	Anomaly-based	Deep learning	No
2	A fog empowered anomaly detection scheme for IoT (2017) [38]	Anomaly-based	i. HyCARCE ii. ENOF	No
3	Semi-Supervised learning based distributed attack detection framework (2018) [22]	Anomaly-based	i. Extreme Learning Machine (ELM) and ii. Semi-Supervised Fuzzy C-Means (SFCM) algorithm	No
4	An Anomaly Detection for IoT (AD-IoT) scheme [37]	Anomaly-based	Random Forest	No
5	Proposed framework	Hybrid-based	Extreme Gradient Boosting	Yes

4.2. Extreme Gradient Boosting

The Extreme Gradient Boosting (XGBoost) algorithm was proposed in [9] by Chen and Guestrin. XGBoost is an improved variant of the gradient boosting algorithm, which is generally based on ensemble techniques. In XGBoost, the predictions of weak learners are combined to develop a strong learner by employing additive techniques. Apart from the benefits of speed and performance of XGBoost, additional advantages are avoiding overfitting and full utilization of computational resources. These are achieved through simplifying objective functions that permit a combination of regularization and predictive terms and its ability to execute the training stage in parallel, respectively.

The procedures used in XGBoost [9] for additive learning are elaborated below. The first learner is fitted to the entire input data; subsequently, the second learner is fitted with the errors of the first learner. The process continues until a stopping condition is attained, which translates to a final prediction model that is achieved by adding up the prediction of all the learners. The equation below presents the prediction function at step t.

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{1}$$

where f_t (x_i) is the learner at step t, $\hat{y}_i^{(t)}$ and $\hat{y}_i^{(t-1))}$ are the predictions at steps t and $t-1$, and x_i is the input variable.

To avoid the problem of overfitting without trading the speed of the model in terms of computation, the equation below, which is obtained from the original function, evaluates the goodness of the model.

$$Obj^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^{t} \alpha(f_i) \tag{2}$$

where l is the loss function, n is the number of observations used, and α is the regularization function. To define the complexity of the tree $\alpha(f_i)$, regularization plays a vital role and the definition of f_i is refined as:

$$f_t(x) = \mu_{p(x)}, \ \mu \,\epsilon\, R^T, \ p : R^d \rightarrow \{1, 2, 3 \ldots..T\}. \tag{3}$$

where T is the number of leaves, μ represents the vector of scores in the leaves, and p assigns each data point to the corresponding leaf. The regularization function is expressed as:

$$\alpha(f) = \delta T + \frac{1}{2}\sigma \parallel \mu \parallel^2 \tag{4}$$

where σ is the regularization parameter and δ denotes the minimum loss needed to further partition the leaf node. The tree structure is obtained by computing the objective function, leaf scores, and regularization at each level [40]. More information on XGBoost can be found in [9,41].

5. Performance Evaluation

In this section, a performance evaluation of the signature-based module and the proposed algorithm utilized in the anomaly-based module of the framework is conducted. The dataset used, performance metrics evaluated, evaluation methodology, as well as the results and discussions, are presented.

5.1. Description of the Dataset

Performance evaluation is essential in the development of anomaly mitigation schemes. The evaluation assists in determining the efficiency of the schemes. However, since the evaluations are impossible on a real network, there is a need to use a well-structured dataset that gives a good representation of the traffic flow characteristics of the environment where the scheme will be deployed.

In line with this, we are going to utilize the BoT-IoT dataset [42], which provides an ideal scenario of the traffic flow in an IoT environment.

The BoT-IoT dataset was developed at the University of New South Wales Canberra, Australia. It consists of legitimate traffic and simulated IoT traffic alongside different types of attacks. The dataset was developed using a testbed that consists of network platforms, simulated IoT services, and feature extraction tools.

The utilized network platforms used consist of virtual machines that generated legitimate and illegitimate traffic. The IoT traffic is generated from a simulation using the Node-red tool [43]. It utilizes the Message Queuing Telemetry Transport (MQTT) protocol [44] to simulate network traffic of a smart building. The scenarios simulated include a weather station, smart fridge, smart garage, smart lights, and smart thermostat. The Argus tool [45] was used to extract features from the generated traffic, which consists of the legitimate and illegitimate traffic. Cron Linux functions [46] were employed for labeling of the extracted traffic. A total of 30 features were extracted and 14 new features were generated from them to enhance the prediction abilities of the classifiers to be employed. The composition of the BoT-IoT dataset is depicted in Figure 4.

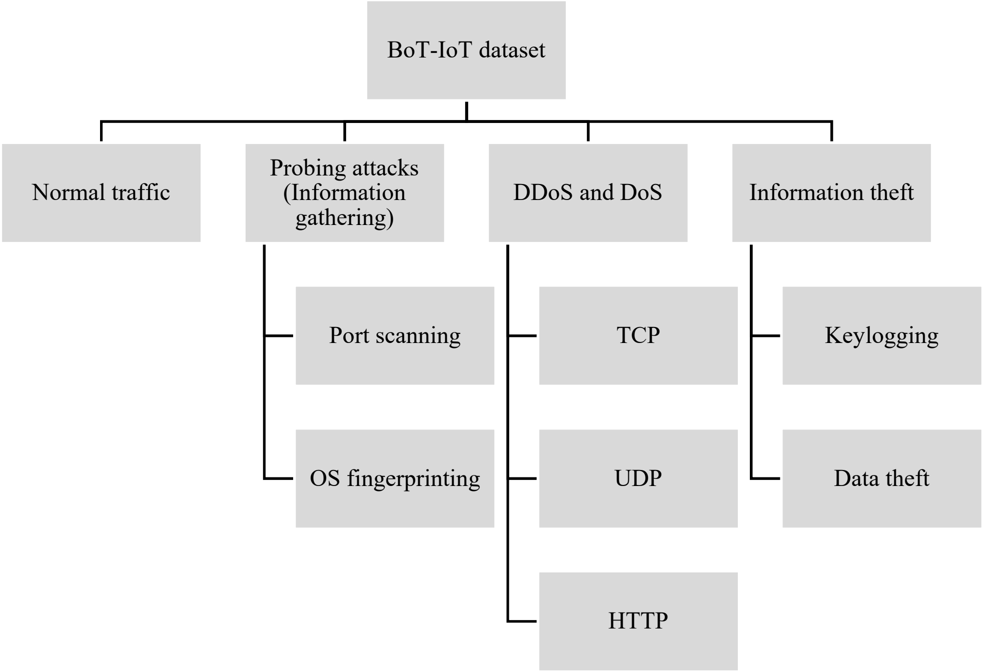

Figure 4. Composition of the BoT-IoT dataset.

- Normal Traffic: Normal traffic consists of legitimate network transactions. The dataset contains legitimate traffic flows generated from the virtual machines using the Ostinato tool [47].
- Probing attacks [42]: A probing attack, also called an information gathering attack, involves the process applied by malicious users in collecting information illegitimately from remote systems through scanning or fingerprinting. The BoT-IoT dataset contains two types of probing attacks which are: port scanning and OS fingerprinting. The port scanning attack instances are generated using the Nmap [42] and Hping3 [48] tools, while the fingerprinting was performed using Nmap [49] and Xprobe2 [50] tools.
- Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks [42]: These attacks involve the process employed by malicious users to deny legitimate users access to resources or services by overwhelming these resources or services with illegitimate requests. The attacks are usually executed using botnets, which are a collection of compromised nodes in the network. The dataset contains DoS and DDoS attacks instances based on HTTP, TCP, and UDP protocols.

The TCP and UDP DoS and DDoS attacks were generated using the Hping3 [48] tool, while the HTTP DoS and DDoS attacks were generated using the Golden-eye tool.

- Information theft [42]: An information theft attack is a process employed by malicious users to violate the security of the system to obtain confidential or sensitive data. The dataset contains two types of information theft attacks, which are data theft and keylogging. These attacks were executed using the Metasploit framework [51].

The BoT-IoT dataset consists of a total of 9543 and 73,360,900 instances of legitimate and illegitimate traffic flow, respectively. However, for the experiment in this paper, 740,637 instances were extracted randomly. The extracted instances contain all the types of attack, excluding data theft attacks, which have a very small number in the dataset. Table 3 presents a summary of the extracted instances.

Table 3. Summary of original and extracted instances.

S/No	Type of Attack	Number of Instances	Number of Extracted Instances
1	Normal	9543	225
2	DoS TCP	12,315,997	123,185
3	DoS UDP	20,659,491	206,626
4	DoS HTTP	29,706	301
5	DDoS TCP	19,547,603	1,951,525
6	DDoS UDP	18,965,106	189,954
7	DDoS HTTP	19,771	203
8	Keylogging	1469	34
9	OS fingerprinting	358,275	4953
10	Port scanning	1,463,364	20,004
11	Data theft	118	-
	Total	73,370,443	740,637

5.2. Performance Metrics

The confusion matrix estimation is used in evaluating the performance of IDS schemes [52]. It describes the performance by relating the actual and predicted observations (positive and negative) on the labels. Table 4 describes a typical confusion matrix of an IDS. It is usually defined as a two by two matrix because it has two classes (normal and abnormal).

Table 4. Confusion Matrix.

Actual Class Label	Predicted Class Label	
	Positive	Negative
Positive	True positive	False negative
Negative	False positive	True negative

The terms in the confusion matrix are defined below. These terms are used in computing performance metrics.

i. True Positive (TP): Total actual positive observations that are predicted positive.
ii. False Positive (FP): Total actual negative observations that are predicated positive.
iii. True Negative (TN): Total actual negative observations that are predicted negative.
iv. False Negative (FN): Total actual positive observations that are predicted negative.

Given the True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN), the performance metrics can be defined below:

- Accuracy is the ratio of the overall positive observations to the total observations; it shows the general success rate of any IDS, and is computed as,

$$\text{Accuracy} = (\text{TN} + \text{TP})/(\text{TP} + \text{FP} + \text{TN} + \text{FN}) \tag{5}$$

- The Detection Rate (DR), also called the true positive rate (TPR) or recall, is the ratio of correctly classified malicious observations to the total number of malicious observations and is computed as,

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

- The False Positive Rate (FPR), also called the false alarm rate, is the ratio of normal observations to the total number of normal observations misclassified as attacks and is computed as,

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

- The False Negative Rate (FNR), also called precision, is the ratio of misclassified attack observations to the total number of attack observations, given as,

$$\text{FNR} = \text{FN}/(\text{FN} + \text{TP}) \tag{8}$$

- The F1 Score is the weighted average of the recall and the precision, and is computed as,

$$\text{F1 Score} = 2 \times (\text{Recall} \times \text{Precision})/(\text{Recall} + \text{Precision}) \tag{9}$$

The F1 score is important and gives more insight into the performance of the IDS. It considers the false positives and false negatives. The F1 score is beneficial, especially when the amount of the class labels is uneven or skewed.

5.3. *Evaluation Methodology*

As a proof-of-concept, the proposed framework was implemented using Python programming language. The extracted dataset was used as our data source. We utilized a virtual machine with a Windows 8 operating system and 4GB RAM as a fog node.

To evaluate the signature-based module, the IP addresses of the attack instances were utilized to create the blacklist from the extracted dataset. The blacklist consists of source IP address that were labeled as attack instances. To avoid redundancy, all duplicates are deleted before storing the final blacklist. The signature-based module was evaluated against the anomaly-based module in terms of response time in attack detection.

Similarly, in order to evaluate the performance of the XGBoost algorithm, the extracted dataset was duplicated into two copies and was passed through some data preprocessing steps, which include data transformation and feature selection. As part of the data transformation, the labeling of the class feature in the first copy of the dataset was encoded to binary, i.e., normal (0) and the attack (1) traffic instances. Furthermore, the label of the class feature in the second copy of the dataset was encoded to 0–9, with each number representing a traffic instance type, i.e., normal (0) traffic instances and nine different types of attack (DoS TCP (1), DoS UDP (2), DoS HTTP (3), DDoS TCP (4), DDoS UDP (5), DDoS HTTP (6), Keylogging (7), OS fingerprinting (8), and Port Scanning (9)).

For the feature selection step, similar to [42], 10 features were selected using the Correlation Coefficient [53] and Entropy [54] techniques. These features represent the best features that will give good performance in terms of classification. The 10 selected features are presented in Table 5.

Table 5. Features Selected for Experiments.

S/No	Feature Name	Description
1	seq	Argus sequence number
2	N_IN_Conn_P_DstIP	Number of inbound connections per destination IP
3	stddev	Standard deviation of aggregated records
4	N_IN_Conn_P_SrcIP	Number of inbound connections per source IP
5	min	Minimum duration of aggregated records
6	state_number	Numerical representation of feature state
7	srate	Source-to-destination packets per second
8	mean	Average duration of aggregated records
9	max	Maximum duration of aggregated records
10	drate	Destination-to-source packets per second

To interpret and gain more information on the performance of the XGBoost classifier in relation to the dataset used, we employed SHAP (SHapley Additive exPlanation) values [55] to understand the most important features that affect the output of the XGBoost classifier. Figure 5 shows the average SHAP value impact of the selected features on the classifier output. It shows that mean (average duration of aggregated records) and N_IN_Conn_P_DstIP (number of inbound connections per destination IP) are the most important features used by the classifier to learn and determine an attack. Similarly, Figure 6 shows the SHAP value summary plot. Each traffic flow instance is represented by a dot. The position and color of the dot show its impact and value on the classifiers output.

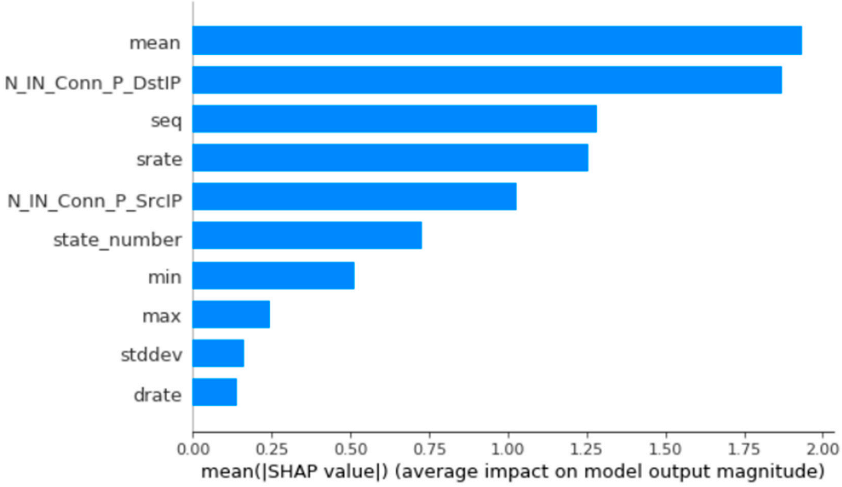

Figure 5. Average SHapley Additive exPlanation (SHAP) values.

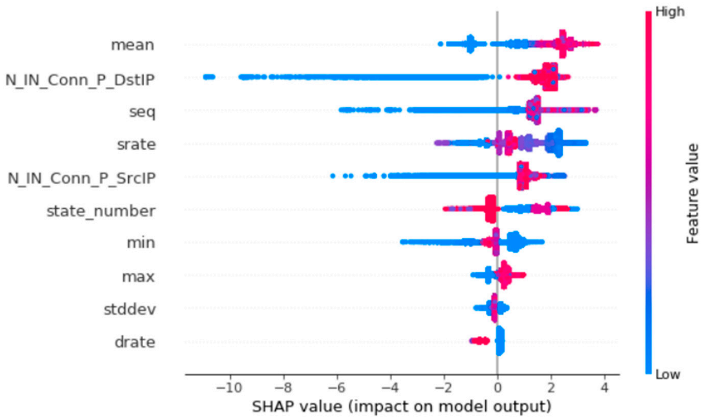

Figure 6. SHAP summary plot.

The XGBoost classifier is evaluated against Decision Tree (DT), k-Nearest Neighbors (k-NN), Naïve Bayes (NB), and Gradient Boosting (GRB) classifiers using k-fold cross-validation (where k = 10). The 10-fold cross-validation divides the dataset into ten parts at random. For each evaluation, one part of the divided dataset is used as a test set, while the other nine parts are used as a training. The process is repeated for each part of the divided dataset. It ensures low variance, bias, and avoids overfitting.

Due to the imbalanced nature of the dataset in terms of its instances of the traffic flows, the classifiers evaluation will be conducted based on the whole classification report to capture the classification abilities of the classifiers in classifying different instances of binary classification (normal and attack instances), multiclass classification (normal, DoS TCP, DoS UDP, DoS HTTP, DDoS TCP, DDoS UDP, DDoS HTTP, keylogging, OS fingerprinting (OS FR), and port scanning (Port SC)), and False Positive Rate (FPR) of the binary classification. The classification report comprises the accuracy, precision, recall, and F1 score.

5.4. Results and Discussion

The evaluation results of the signature-based module and the XGBoost classifier employed in our anomaly-based module of the fog-based framework in binary and multiclass classifications are discussed below.

Figure 7 presents the response time of the signature-based module against the anomaly-based module over the number of network traffic instances. The signature-based module recorded 0.03, 0.046, and 0.14 s, while the anomaly-based module recorded 0.60, 0.64, and 0.96 s over 100, 1000, and 10,000 network traffic instances, respectively. The results show that the signature-based module outperforms the anomaly-based module for all of the network traffic instances.

Table 6 presents the accuracy and FPR of the classifiers in binary class classification. All the classifiers recorded a good result in terms of accuracy in attack detection, with XGBoost recording the highest value of 99.99% and NB recording the lowest value of 99.85%. Meanwhile, in terms of FPR, the XGBoost records the smallest value with 0.05, which shows that the XGBoost classifier can classify the network traffic instances with the lowest false positives.

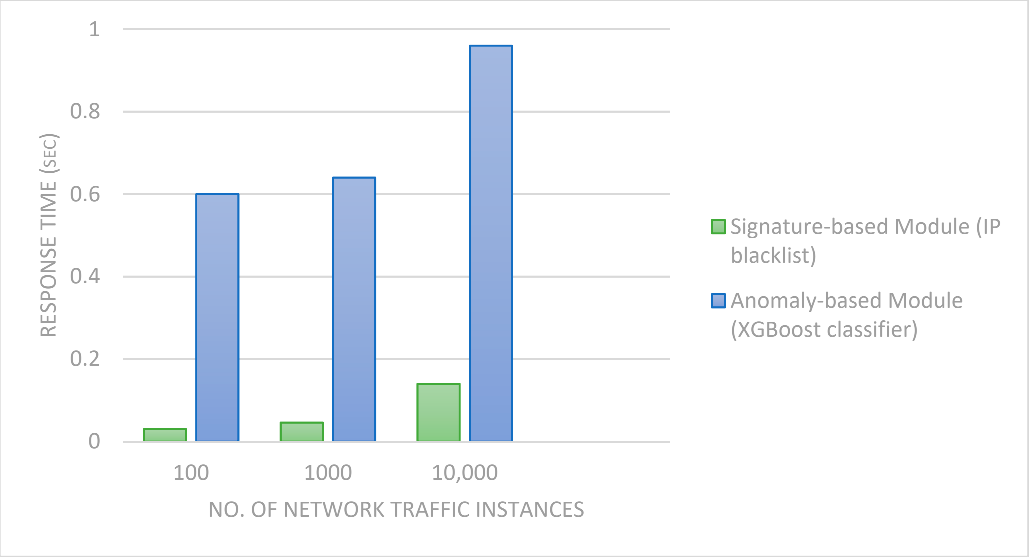

Figure 7. Response time of the signature-based and anomaly-based modules.

Table 6. Accuracy and false positive rate of binary classification.

S/No	ML Algorithm	Accuracy (%)	False Positive Rate
1	DT	99.96	0.13
2	k-NN	99.97	0.68
3	NB	99.85	0.91
4	Gradient Boosting	99.99	0.89
5	XGBoost	99.99	0.05

To gain more understanding in the performance of the classifiers, Figure 8 presents the average performance of the classifiers in binary classification, i.e., normal and attack instances. The XGBoost classifier obtained superior results in terms of average recall, precision, and F1 score than the other classifiers.

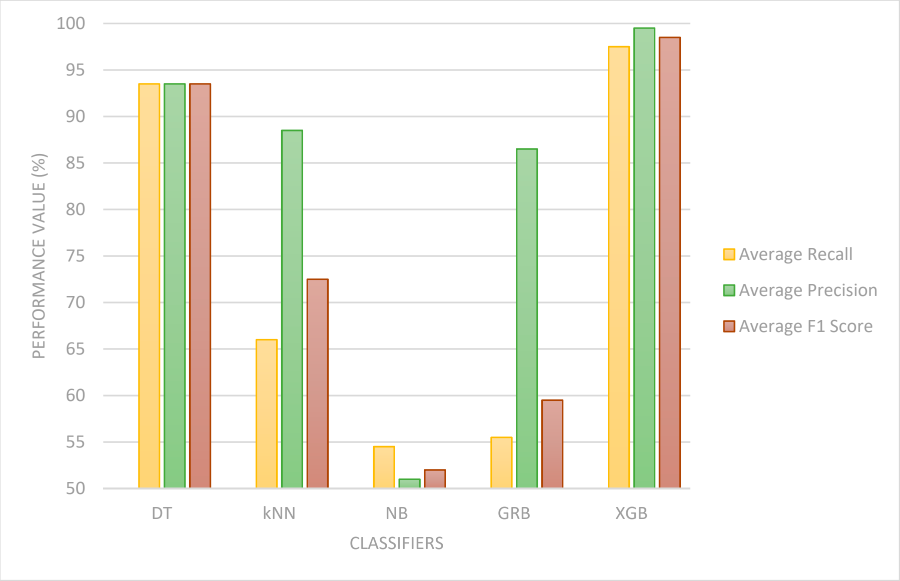

Figure 8. Average performance of classifiers (normal and attack instances).

Figure 9 presents the binary class classification performance of the classifiers in terms of classifying normal traffic instances. The XGBoost recorded the best results with 99.5%, 97.5%, and 98.5% in precision, recall, and F1 score, respectively.

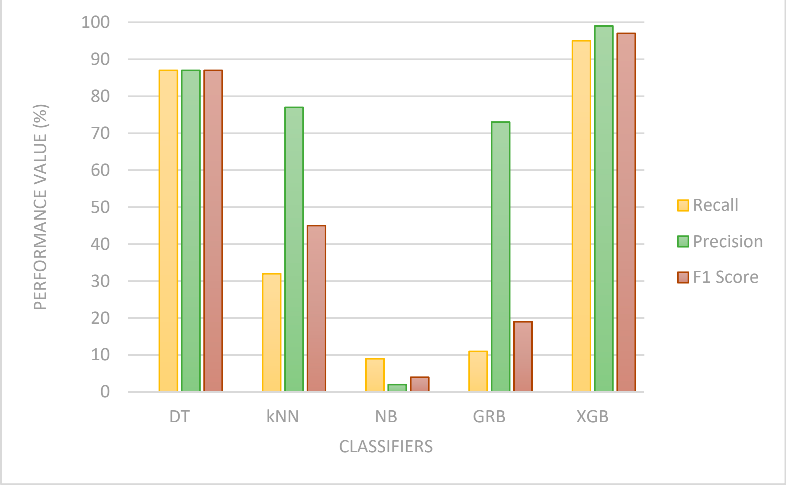

Figure 9. Performance for normal instances classification in binary classification.

Figure 10 presents the accuracy of the classifiers in multiclass classification. Similar to the accuracy in the binary class classification, all the classifiers achieved a good result in multiclass classification with above 99% detecting the majority of the attacks. The k-NN and NB classifiers recorded the lowest result in detecting normal instances, DDoS TCP, DDoS UDP, keylogging, and OS fingerprinting attacks. The XGBoost and DT have identical results; however, the XGBoost recorded the highest accuracy in detecting all the attacks and normal instances.

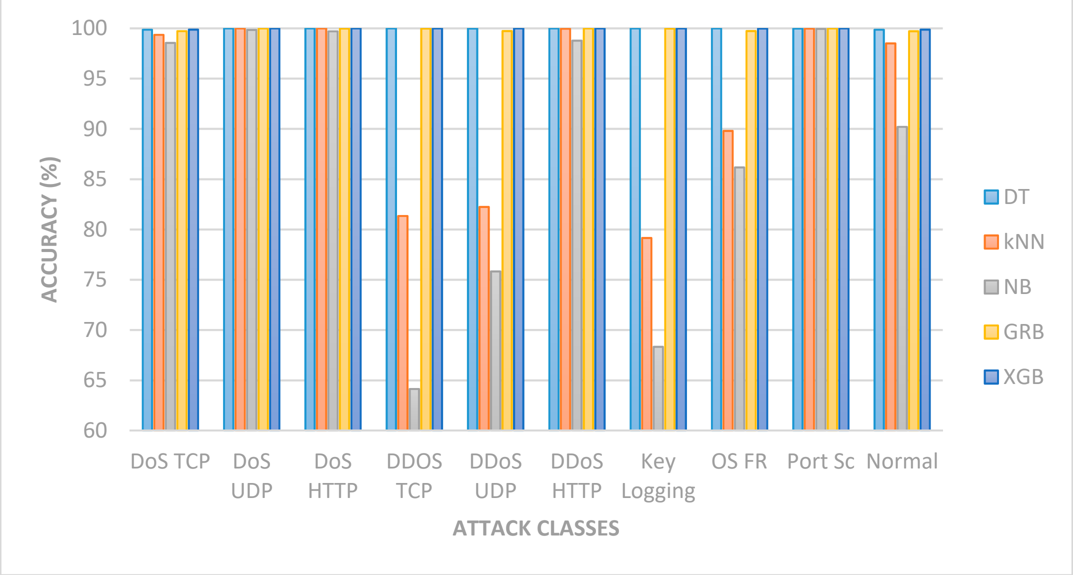

Figure 10. Accuracy of multiclass classification of all Instances.

Figure 11 presents the precision of the classifiers in multiclass classification. The XGBoost, gradient boosting, and DT classifiers achieved a precision of 84–100% in all the attacks and normal instances. The k-NN and NB achieved a precision 2–70% in all the attacks and normal instances with the exception of DDoS TCP attacks, where NB recorded 84% and keylogging attacks, where k-NN recorded 100%.

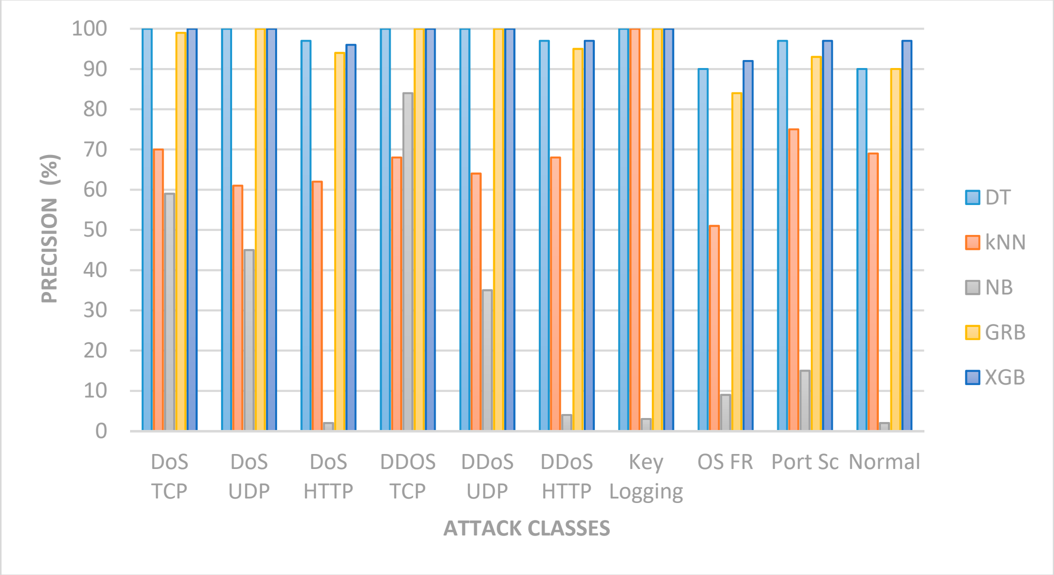

Figure 11. Precision of multiclass classification Instances.

Figure 12 presents the recall of the classifiers in multiclass classification The XGBoost, gradient boosting, and DT classifiers achieved a recall of 83–100% in all the attacks and normal instances, excluding keylogging attacks and the normal instances where gradient boosting obtained 79% and 75%, respectively. The k-NN and NB achieved a recall of 10–79% in all the attacks and normal instances with the exception of keylogging attacks, where NB recorded 86%.

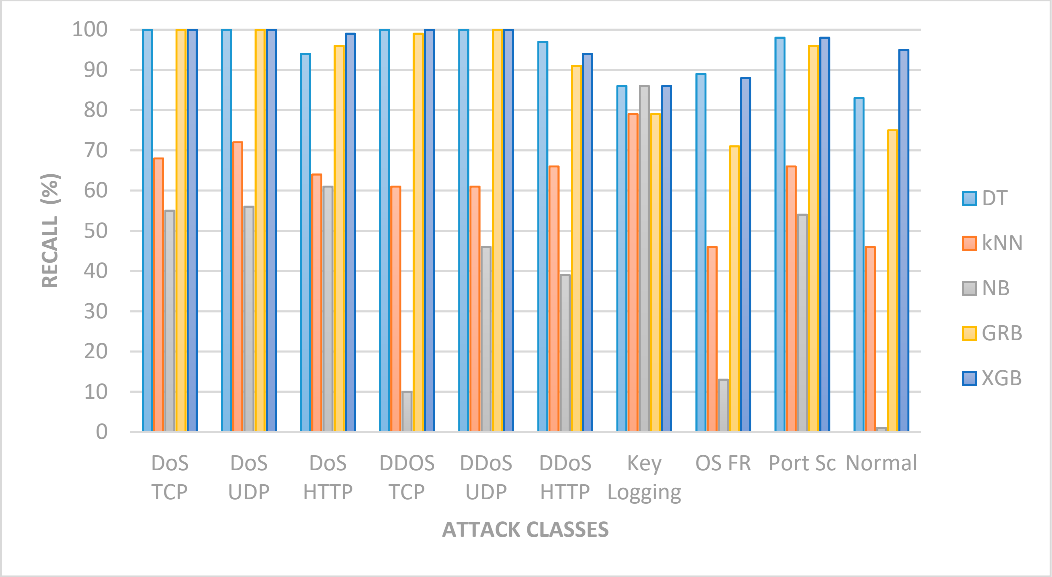

Figure 12. Recall of multiclass classification Instances.

Figure 13 presents the F1 score of the classifiers in multiclass classification. XGBoost, gradient boosting, and DT classifiers achieved an F1 score of 82–100% in all the attacks and normal instances. The k-NN and NB achieved an F1 score of 4–70% in all the attacks and normal instances except for OS fingerprinting attacks, where k-NN recorded 88%.

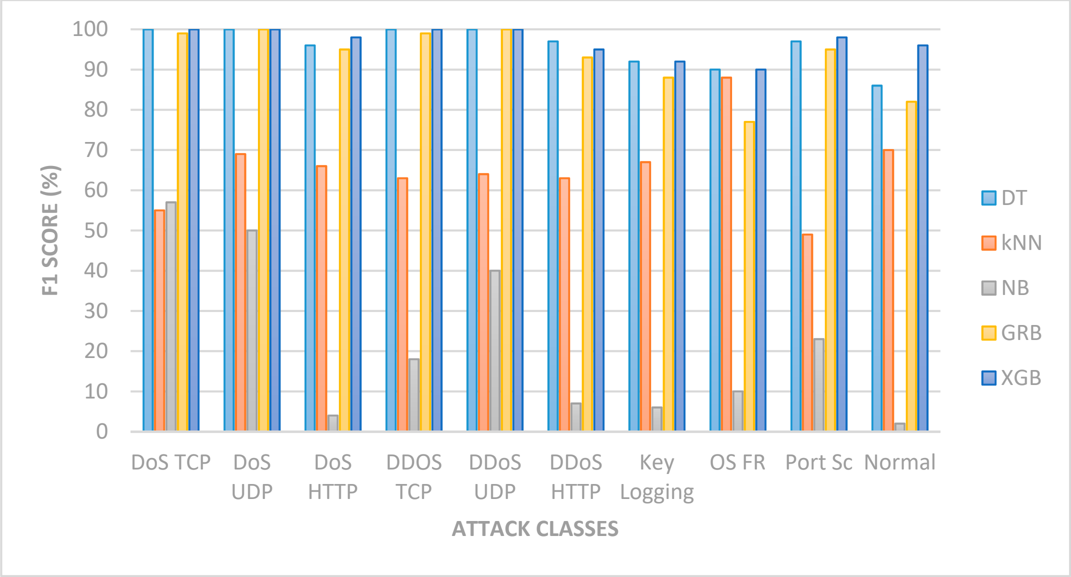

Figure 13. F1 score of multiclass classification Instances.

Figure 14 presents the average multiclass classification results of the classifiers for all normal and attack instances. The XGBoost and DT classifiers recorded good results in terms of the average accuracy and average precision in normal instances and attacks classification with 99.96% and 97%, respectively. In contrast, the NB recorded the lowest average accuracy and average precision with 78.17% and 26%, respectively. XGBoost recorded the highest values in terms of average recall and F1 score with 98% and 97%, respectively, while the NB obtained the lowest results with 42% and 22% in average recall and average F1 score, respectively. Tables 7–11 give a summary of the results of the multiclass classification.

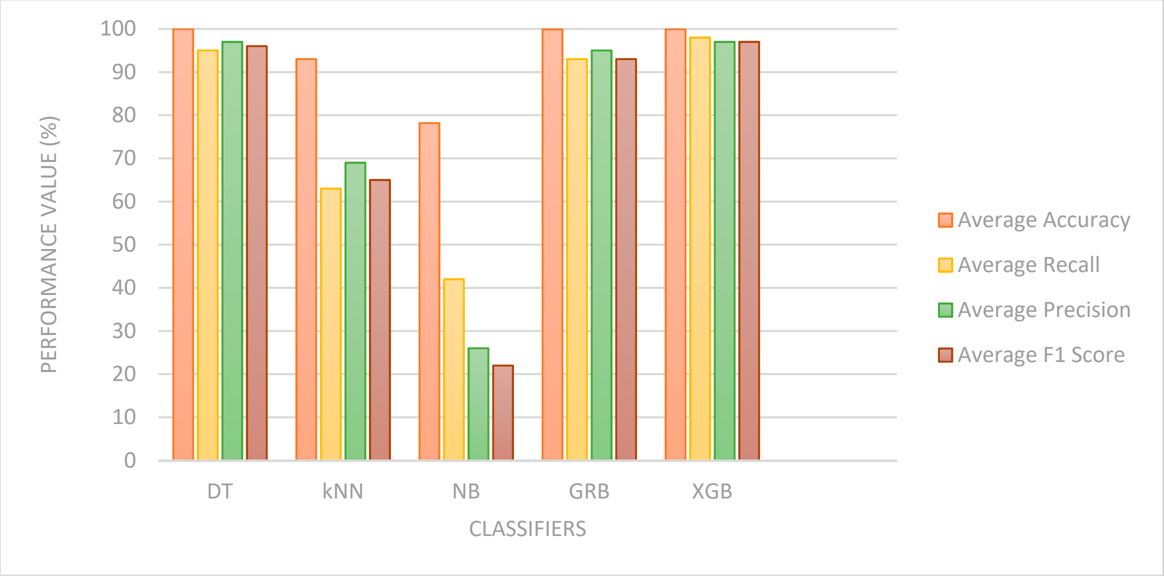

Figure 14. Average performance of classifiers in multiclass classification.

Table 7. Binary classification precision, recall, and F1 Score.

S/No	ML Algorithm	Recall (%)			Precision (%)			F1 Score (%)		
		Normal	Attack	Average	Normal	Attack	Average	Normal	Attack	Average
1	DT	87	100	93.5	87	100	93.5	87	100	93.5
2	k-NN	32	100	66	77	100	88.5	45	100	72.5
3	NB	9	100	54.5	2	100	51	4	100	52
4	Gradient Boosting	11	100	55.5	73	100	86.5	19	100	59.5
5	XGBoost	95	100	97.5	99	100	99.5	97	100	98.5

Table 8. Accuracy summary of the multiclass classification.

S/No	ML Algo	DoS TCP	DoS UDP	DoS HTTP	DDoS TCP	DDoS UDP	DDoS HTTP	Key Logging	OS F	Port Sc	Normal	Average Accuracy
1	DT	99.86	99.99	99.99	99.99	99.98	99.99	99.99	99.99	99.97	99.86	99.96
2	k-NN	99.35	99.99	99.99	81.34	82.24	99.96	79.15	89.80	99.97	98.50	93.00
3	NB	98.55	99.84	99.70	64.14	75.83	98.77	68.33	86.17	99.95	90.20	78.17
4	GRB	99.72	99.99	99.97	99.97	99.73	99.99	99.97	99.73	99.99	99.71	99.88
5	XGB	99.87	99.99	99.99	99.99	99.99	99.99	99.99	99.97	99.99	99.87	99.96

Table 9. Recall summary of the multiclass classification.

S/No	ML Algo	DoS TCP	DoS UDP	DoS HTTP	DDoS TCP	DDoS UDP	DDoS HTTP	Key Logging	OS F	Port Sc	Normal	Average Recall
1	DT	100	100	94	100	100	97	86	89	98	83	95
2	k-NN	68	72	64	61	61	66	79	46	66	46	63
3	NB	55	56	61	10	46	39	86	13	54	1	42
4	GRB	100	100	96	99	100	91	79	71	96	75	93
5	XGB	100	100	99	100	100	94	86	88	98	95	98

Table 10. Precision summary of the multiclass classification.

S/No	ML Algo	DoS TCP	DoS UDP	DoS HTTP	DDoS TCP	DDoS UDP	DDoS HTTP	Key Logging	OS F	Port Sc	Normal	Average Precision
1	DT	100	100	97	100	100	97	100	90	97	90.0	97
2	k-NN	70	61	62	68	64	68	100	51	75	69	69
3	NB	59	45	2	84	35	4	3	9	15	2	26
4	GRB	99	100	94	100	100	95	100	84	93	90	95
5	XGB	100	100	96	100	100	97	100	92	97	97	97

Table 11. F1 score summary of the multiclass classification.

S/No	ML Algo	DoS TCP	DoS UDP	DoS HTTP	DDoS TCP	DDoS UDP	DDoS HTTP	Key Logging	OS F	Port Sc	Normal	Average F1 Score
1	DT	100	100	96	100	100	97	92	90	97	86	96
2	k-NN	55	69	66	63	64	63	67	88	49	70	65
3	NB	57	50	4	18	40	7	6	10	23	2	22
4	GRB	99	100	95	99	100	93	88	77	95	82	93
5	XGB	100	100	98	100	100	95	92	90	98	96	97

In summary, the results show that the signature-based module has the ability to detect attacks with lesser time as compared with the anomaly-based module. Similarly, the binary and multiclass classification results show that the XGBoost classifier obtained superior results in terms of accuracy, recall, precision, and F1 score as well as the FPR than the DT, NB, k-NN, and gradient boosting classifiers. This indicates its effectiveness in distinguishing normal and attack instances with high accuracy and minimal errors. These evaluations show that employing both signature-based and anomaly-based IDS in the anomaly mitigation module of the proposed framework will yield faster and accurate attack detection. Hence, these good performances will help improve the security of the IoT networks. Table 12 presents summary results of binary and multiclass classification.

Table 12. Summary of binary and multiclass results.

S/No	ML Algo	Accuracy (%)		Recall (%)		Precision (%)		F1 Score (%)	
		Binary	Multi Class	Binary	Multi Class	Binary	Multi Class	Binary	Multi Class
1	DT	99.96	99.96	93.5	95	93.5	97	93.5	96
2	k-NN	99.97	93.00	66	63	88.5	69	72.5	65
3	NB	99.85	78.17	54.5	42	51	26	52	22
4	Gradient Boosting	99.99	99.88	55.5	93	86.5	95	59.5	93
5	XGBoost	99.99	99.96	97.5	97	99.5	97	98.5	97

6. Conclusions

The combination of fog computing and IoT have provided an efficient platform for the deployment of anomaly mitigation schemes to solve security challenges such as botnet attacks. This paper proposed a hybrid anomaly mitigation framework for IoT using fog computing to ensure faster and accurate anomaly detection. The framework employs two modules, namely signature-based and anomaly-based. The signature-based module employs an IP blacklist to ensure faster attack detection, while the anomaly-based module utilizes an extreme gradient boosting algorithm for classifying network traffic flow into normal or abnormal. The IP blacklist is updated with attack sources detected by the anomaly-based module to ensure fast detection when these attacks are executed again. We evaluated the proposed modules using a BoT-IoT dataset. The results show that the signature-based module is 19 times faster, 12 times faster, and 6 times faster than the anomaly-based module over 100, 1000, and 10,000 network traffic instances, respectively. Additionally, the XGBoost achieved superior results with 99.99% accuracy, 97.5% recall, 99.5% precision, 98.5% F1 score, and a false positive rate of

0.05 for binary classification. Similarly, it recorded 99.96% average accuracy, 98% average recall, 97% average precision, and 97% average F1 score for multiclass classification.

These results demonstrate that the signature-based module detects attacks faster than the anomaly-based module. Similarly, the anomaly-based module can detect different types of attacks with satisfactory performance.

As future work, we intend to explore other features of botnet attacks to create additional signatures for our framework.

Author Contributions: Conceptualization, M.A.L., R.A.S. and S.R.H.; methodology, M.A.L., R.A.S. and S.R.H.; writing—original draft preparation, M.A.L.; writing—review and editing, R.A.S. and S.R.H.; supervision, R.A.S. and S.R.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors wish to thank King Abdulaziz University Jeddah, Saudi Arabia for providing the necessary facilities for conducting their research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pacheco, J.; Hariri, S. Anomaly behavior analysis for IoT sensors. *Trans. Emerg. Telecommun. Technol.* **2018**, *29*, 1–15. [CrossRef]
2. Ahmad, M.; Salah, K. IoT security: Review, blockchain solutions, and open challenges. *Future Gener. Comput. Syst.* **2018**, *82*, 395–411.
3. Hoang, D.H.; Nguyen, H.D. A PCA-based method for IoT network traffic anomaly detection. In Proceedings of the International Conference on Advanced Communication Technology, ICACT, Chuncheon-si Gangwon-do, Korea, 11–14 February 2018; Volume 2018, pp. 381–386.
4. Moustafa, N.; Hu, J.; Slay, J. A holistic review of Network Anomaly Detection Systems: A comprehensive survey. *J. Netw. Comput. Appl.* **2019**, *128*, 33–55. [CrossRef]
5. Acarali, D.; Rajarajan, M.; Komninos, N.; Herwono, I. Survey of approaches and features for the identification of HTTP-based botnet traffic. *J. Netw. Comput. Appl.* **2016**, *76*, 1–15. [CrossRef]
6. Simonovich, V. Imperva Blocks Our Largest DDoS L7/Brute Force Attack Ever (Peaking at 292,000 RPS). Available online: https://www.imperva.com/blog/imperva-blocks-our-largest-ddos-l7-brute-force-attack-ever-peaking-at-292000-rps (accessed on 18 December 2019).
7. Asokan, A. Massive Botnet Attack Used More Than 400,000 IoT Devices. Available online: https://www.bankinfosecurity.com/massive-botnet-attack-used-more-than-400000-iot-devices-a-12841 (accessed on 18 December 2019).
8. Yaseen, Q.; Albalas, F.; Jararwah, Y.; Al-Ayyoub, M. Leveraging fog computing and software defined systems for selective forwarding attacks detection in mobile wireless sensor networks. *Trans. Emerg. Telecommun. Technol.* **2018**, *29*, e3183. [CrossRef]
9. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; ACM: New York, NY, USA, 2016; pp. 785–794.
10. Quinlan, J.R. *C4. 5: Programs for Machine Learning*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1993.
11. Fix, E.; Hodges, J.L. *Discriminatory Analysis. Nonparametric Discrimination; Consistency Properties*; Technical Report 4; USAF School of Aviation Medicine Randolph Field: San Antonio, TX, USA, 1951.
12. Zhang, H. Exploring conditions for the optimality of naïve bayes. *Int. J. Pattern Recognit. Artif. Intell.* **2005**, *19*, 183–198. [CrossRef]
13. Elrawy, M.F.; Awad, A.I. Intrusion detection systems for IoT-based smart environments: A survey. *J. Cloud Comput. Adv. Syst. Appl.* **2018**, *7*, 21. [CrossRef]
14. Minerva, R.; Biru, A.; Rotondi, D. Towards a definition of the Internet of Things (IoT). *IEE Internet Initiat.* **2015**, *1*, 1–86.
15. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Futur. Gener. Comput. Syst.* **2013**, *29*, 1645–1660. [CrossRef]

16. Ammar, M.; Russello, G.; Crispo, B. Internet of Things: A survey on the security of IoT frameworks. *J. Inf. Secur. Appl.* **2018**, *38*, 8–27. [CrossRef]
17. Khattak, H.A.; Shah, M.A.; Khan, S.; Ali, I.; Imran, M. Perception layer security in Internet of Things. *Futur. Gener. Comput. Syst.* **2019**, *100*, 144–164. [CrossRef]
18. Al-fuqaha, A.; Member, S.; Guizani, M.; Mohammadi, M.; Member, S. Internet of Things: A Survey on Enabling. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2347–2376. [CrossRef]
19. Bonomi, F.; Milito, R.; Zhu, J.; Addepalli, S. Fog Computing and Its Role in the Internet of Things. In Proceedings of the First Edition Workshop on Mobile Cloud Computing (MCC), Helsinki, Finland, 17 August 2012; ACM: New York, NY, USA, 2012; pp. 13–16.
20. Rathore, S.; Park, J.H. Semi-supervised learning based distributed attack detection framework for IoT. *Appl. Soft Comput. J.* **2018**, *72*, 79–89. [CrossRef]
21. Neware, R.; Shrawankar, U. Fog Computing Architecture, Applications and Security Issues: A Survey. *Int. J. Fog Comput.* **2020**, *3*, 75–105. [CrossRef]
22. Raza, S.; Wallgren, L.; Voigt, T. SVELTE: Real-time intrusion detection in the Internet of Things. *Ad Hoc Netw.* **2013**, *11*, 2661–2674. [CrossRef]
23. Shaikh, R.A.; Jameel, H.; d'Auriol, B.J.; Lee, H.; Lee, S.; Song, Y.J. Intrusion-aware alert validation algorithm for cooperative distributed intrusion detection schemes of wireless sensor networks. *Sensors* **2009**, *9*, 5989–6007. [CrossRef]
24. L-Hawawreh, M.A.; Moustafa, N.; Sitnikova, E. Identification of malicious activities in industrial internet of things based on deep learning models. *J. Inf. Secur. Appl.* **2018**, *41*, 1–11.
25. Zarpelão, B.B.; Miani, R.S.; Kawakani, C.T.; de Alvarenga, S.C. A survey of intrusion detection in Internet of Things. *J. Netw. Comput. Appl.* **2017**, *84*, 25–37. [CrossRef]
26. Lawal, M.A.; Shaikh, R.A.; Hassan, S.R. Security Analysis of Network Anomalies Mitigation Schemes in IoT Networks. *IEEE Access* **2020**, *8*, 43355–43374. [CrossRef]
27. Sforzin, A.; Marmol, F.G.; Conti, M.; Bohli, J.M. RPiDS: Raspberry Pi IDS—A Fruitful Intrusion Detection System for IoT. In Proceedings of the 2016 Intl IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), Toulouse, France, 18–21 July 2016; pp. 440–448.
28. Ioulianou, P.P.; Vassilakis, V.G.; Moscholios, I.D.; Logothetis, M.D. A Signature-based Intrusion Detection System for the Internet of Things. In Proceedings of the Information and Communication Technology Forum (ICTF), Graz, Austria, 11–13 July 2018.
29. Kasinathan, P.; Pastrone, C.; Spirito, M.A.; Vinkovits, M. Denial-of-Service detection in 6LoWPAN based Internet of Things. In Proceedings of the International Conference on Wireless and Mobile Computing, Networking and Communications, Lyon, France, 7–9 October 2013; pp. 600–607.
30. Ullah, I.; Mahmoud, Q.H. A Two-Level Hybrid Model for Anomalous Activity Detection in IoT Networks. In Proceedings of the 2019 16th IEEE Annual Consumer Communications and Networking Conference, CCNC 2019, Las Vegas, NV, USA, 11–14 January 2019; pp. 1–6.
31. Jan, S.U.; Ahmed, S.; Shakhov, V.; Koo, I. Toward a Lightweight Intrusion Detection System for the Internet of Things. *IEEE Access* **2019**, *7*, 42450–42471. [CrossRef]
32. Deshmukh-Bhosale, S.; Sonavane, S.S. A Real-Time Intrusion Detection System for Wormhole Attack in the RPL based Internet of Things. *Procedia Manuf.* **2019**, *32*, 840–847. [CrossRef]
33. Anthi, E.; Williams, L.; Slowinska, M.; Theodorakopoulos, G.; Burnap, P. A Supervised Intrusion Detection System for Smart Home IoT Devices. *IEEE Internet Things J.* **2019**, *6*, 9042–9053. [CrossRef]
34. Otoum, Y.; Liu, D.; Nayak, A. DL-IDS: A deep learning–based intrusion detection framework for securing IoT. *Trans. Emerg. Telecommun. Technol.* **2019**, e3803. [CrossRef]
35. Verma, A.; Ranga, V. ELNIDS: Ensemble Learning based Network Intrusion Detection System for RPL based Internet of Things. In Proceedings of the 4th International Conference on Internet of Things: Smart Innovation and Usages, IoT-SIU 2019, Ghaziabad, India, 18–19 April 2019.
36. Napiah, M.N.; Idris, M.Y.I.B.; Ramli, R.; Ahmedy, I. Compression Header Analyzer Intrusion Detection System (CHA - IDS) for 6LoWPAN Communication Protocol. *IEEE Access* **2018**, *6*, 16623–16638. [CrossRef]

37. Alrashdi, I.; Alqazzaz, A.; Aloufi, E.; Alharthi, R.; Zohdy, M.; Ming, H. AD-IoT: Anomaly detection of IoT cyberattacks in smart city using machine learning. In Proceedings of the IEEE 9th Annual Computing and Communication Workshop and Conference, CCWC 2019, Las Vegas, NV, USA, 7–9 January 2019; pp. 305–310.
38. Lyu, L.; Jin, J.; Rajasegarar, S.; He, X.; Palaniswami, M. Fog-Empowered Anomaly Detection in Internet of Things using Hyperellipsoidal Clustering. *IEEE Internet Things J.* **2017**, *4*, 1174–1184. [CrossRef]
39. Diro, A.A.; Chilamkurti, N. Distributed attack detection scheme using deep learning approach for Internet of Things. *Futur. Gener. Comput. Syst.* **2018**, *82*, 761–768.
40. Dhaliwal, S.S.; Al-Nahid, A.; Abbas, R. Effective Intrusion Detection System Using XGBoost. *Information* **2018**, *9*, 149.
41. Chatterjee, D.R. Log Book—XGBoost, the Math behind the Algorithm. Available online: https://towardsdatascience.com/log-book-xgboost-the-math-behind-the-algorithm-54ddc5008850 (accessed on 11 September 2020).
42. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. *Futur. Gener. Comput. Syst.* **2019**, *100*, 779–796. [CrossRef]
43. Node-Red, Node Red Tool. Available online: https://nodered.org/ (accessed on 4 December 2019).
44. Eclipse, Mosquitto MQTT Broker. Available online: https://mosquitto.org/ (accessed on 4 December 2019).
45. Argus Tool. Available online: https://qosient.com/argus/index.shtm (accessed on 1 December 2019).
46. Cron Scheduling Package. Available online: https://packages.ubuntu.com/search?keywords=cron (accessed on 3 December 2019).
47. Ostinato Tool. Available online: https://ostinato.org (accessed on 30 November 2019).
48. Hping. Available online: http://www.hping.org (accessed on 30 November 2019).
49. Lyon, G.F. *Nmap Network Scanning: The Offcial Nmap Project Guide to Network Discovery and Security Scanning*; Insecure: Los Angeles, CA, USA, 2009.
50. Xprobe2. Available online: https://www.aldeid.com/wiki/Xprobe2 (accessed on 29 November 2019).
51. Metasploit Framework. Available online: https://www.metasploit.com (accessed on 30 November 2019).
52. Bhuyan, M.H.; Bhattacharyya, D.K.; Kalita, J.K. Network Anomaly Detection: Methods, Systems and Tools. *IEEE Commun. Surv. Tutor.* **2014**, *6*, 303–336.
53. Hall, G. Pearson's correlation coefficient. *Other Words* **2015**, *1*, 1–4.
54. Lesne, A.; Etudes, H. Shannon entropy: A rigorous mathematical notion at the crossroads between probability, information theory, dynamical systems and statistical physics. *Math. Struct. Comput. Sci.* **2014**, *24*. [CrossRef]
55. Lundberg, S.M.; Erion, G.G.; Lee, S. Consistent Individualized Feature Attribution for Tree Ensembles. *arXiv* **2019**, arXiv:1802.03888.