Highlights

Collaborative Device-level Botnet Detection for Internet of Things

Muhammad Hassan Nasir,Junaid Arshad,Muhammad Mubashir Khan

- A review of the state-of-the-art device-level intrusion detection approaches.
- A detailed analysis of existing botnet datasets and their features to support evaluation of IDS.
- A novel trustworthy botnet detection framework for efficient and effective detection of IoT botnets.
- Evaluation of device-level botnet detection using IoT datasets (ISOT, IoT23 and BotIoT) with Snort and Suricata

Collaborative Device-level Botnet Detection for Internet of Things

Muhammad Hassan Nasir[a], Junaid Arshad[b] and Muhammad Mubashir Khan[c]

[a]*Department of Computer Science & IT, NED University of Engineering & Technology Karachi, Pakistan*
[b]*School of Computing and Digital Technology, Birmingham City University, UK*
[c]*Department of Computer Science & IT, NED University of Engineering & Technology Karachi, Pakistan*

ARTICLE INFO

Keywords:
Internet of Things
Botnets
Intrusion Detection
Device-level Security

ABSTRACT

Cyber attacks on the Internet of Things (IoT) have seen a significant increase in recent years. This is primarily due to the widespread adoption and prevalence of IoT within domestic and critical national infrastructures, as well as inherent security vulnerabilities within IoT endpoints. Therein, botnets have emerged as a major threat to IoT-based infrastructures targeting firmware vulnerabilities such as weak or default passwords to assemble an army of compromised devices which can serve as a lethal cyber-weapon against target systems, networks, and services. In this paper, we present our efforts to mitigate this challenge through the development of an intrusion detection system that resides within an IoT device to provide enhanced visibility thereby achieving security hardening of such devices. The device-level intrusion detection presented here is part of our research framework *BTC_SIGBDS* (Blockchain-powered, Trustworthy, Collaborative, Signature-based Botnet Detection System). We identify the research challenge through a systematic critical review of existing literature and present detailed design of the device-level component of the *BTC_SIGBDS* framework. We use a signature-based detection scheme with trusted signature updates to strengthen protection against emerging attacks. We have evaluated the suitability and enhanced the capability through the generation of custom signatures of two of the most famous signature-based IDS with ISOT, IoT23, and BoTIoT datasets to assess the effectiveness with respect to detection of anomalous traffic within a typical resource-constrained IoT network in terms of number of alerts, detection rates, detection time as well as in terms of peak CPU and memory usage.

1. Introduction

The emergence of Internet of Things (IoT) has significantly reformed the everyday life in the modern world. IoT paradigm enables small internet connected smart devices embedded with sensors and actuators to act as a facilitator to efficiently connect people, households, offices/businesses, and healthcare services etc. These devices are utilized to accomplish a number of activities including socializing, information sharing, monitoring & control. For instance, a sensor placed in the soil sends humidity & moisture telemetry to the water system for irrigation of agricultural land. These smart devices are typically resource constrained in nature with limited resources available for employing efficient security mechanisms. Moreover, the enormous amount of sensitive data generated by these devices also attracts malicious adversaries to exploit inherent vulnerabilities to gain access to these devices which may result in information leakage or a Distributed Denial of Service (DDoS) attack.

Due to the recent boom in utilizing ubiquitous smart devices, the world has witnessed increasing attacks employing botnets [1, 2] . Specifically, the proliferation of IoT botnet attacks has emerged as one of the primary security concerns as the malicious adversaries exploit weak security configurations for IoT devices to assemble a formidable botnet which can trigger a devastating and large scale DDoS attack. Contrary to the traditional botnets which infect computers or server machine, an IoT botnet is more complex and has a large scale impact which typically infects thousands of internet-connected devices including surveillance cameras, DVRs, smart appliances, and wearables etc. with a malware that allows carrying the tasks similar to traditional botnets. Since the number of connected devices is continuously increasing and almost 50 billion devices are expected in the next few years [3, 4], it entices malactors to leverage security vulnerabilities of IoT devices to spread malware at a faster pace.

Intrusion detection within IoT has attracted significant attention from the scientific community with number of efforts been made to establish efficient intrusion detection models by using centralized, distributed and hybrid approaches (Refer to Table 2, 3, and 4 for details). These approaches either uses collaborative or individual detection strategies. However, these approaches have issues including single point of failure, tested on small data samples or outdated datasets such as KDD-CUP99/NSL-KDD that do not represent modern day traffic, assuming a pre-trusted environment, therefore, do not contain mechanism to cope with insider adversaries. Furthermore, few studies including [5, 6, 7, 8] employ a signature detection mechanism which is although efficient but these approaches either present only a framework that does not target DDoS attack which is considered to be the most devastating botnet attack and do not have the mechanism for protection of signature database from being corrupted, or devise a HIDS model hence do not consider network traffic. Moreover, the collaborative approaches are either not tested in the real environment or only tested for energy efficiency (refer to Section 3 for details). This necessitates an intrusion detection model that is efficient enough to

Email addresses: mhassan.cse@gmail.com (M.H. Nasir); junaid.arshad@bcu.ac.uk (J. Arshad); mmkhan@neduet.edu.pk (M.M. Khan)
ORCID(s): 0000-0001-9803-9381 (M.H. Nasir); 0000-0003-2948-9527 (J. Arshad); 0000-0002-0011-9525 (M.M. Khan)

cater these issues.

1.1. Problem statement

The IoT botnet attacks have emerged as one of the primary concerns of research community due to their complexity and large scale effects. The impact of botnets for IoT devices is aggravated as these are typically resource-constrained and minuscule amount of resources are available to employ an efficient security mechanism. Existing researchers have contributed to this area however these approaches are limited in that they are generally susceptible to single point of failure, require significant profiling time for each device, have not been tested in real environment, evaluated using outdated datasets or small data samples, and do not consider insider attacks. We expand on these in Section 3 including a detailed gap analysis.

In order to address the issues highlighted during our literature review process, our focus is to develop a collaborative botnet detection framework *BTC_SigBDS* (Blockchain powered, Trusted, and Collaborative Signature-based Botnet Detection System) whereby multiple IoT nodes collaborate to achieve efficient detection of botnets. Due to the collaborative design of the approach, the workload for intrusion detection is shared across multiple nodes resulting in a more efficient detection process. The approach also incorporates a reputation management scheme to establish a trusted environment and prevent misbehaving nodes which may attempt to influence the detection process. Further, it leverages blockchain technology to securely store attack signatures and trust values of each node within the IoT network thereby preventing it from being corrupted by malicious adversaries.

1.2. Contributions

In this paper, we present a botnet detection framework in which our focus is on the device-level detection aspect emphasising the role of IoT nodes in achieving effective botnet detection. Therein, we present the design and development of device-level botnet detection engine which utilises signature based intrusion detection techniques. We also present an implementation and evaluation of this system using two signature based intrusion detection tools i.e. Snort & Suricata with three of the publicly available botnet datasets namely ISOT [9], BotIoT [10], and IoT23 [11] with default signature (rule) base as well with custom updates. Specifically, We make following contributions:

- We present a critical insight into state-of-the-art within device-level IoT botnet detection to highlight open research challenges in this domain. We have used a systematic approach to conduct this review, analysing research efforts published within IEEE, Elsevier, ACM and Springer.

- We present a novel collaborative botnet detection framework *BTC_SigBDS* (Blockchain-powered, Trustworthy, Collaborative, Signature-based Botnet Detection System). BTC_SigBDS is powered by node-level reputation system to create a trustworthy environment and a signatures based detection scheme to efficiently detect known attacks where signatures are shared in a trustworthy manner through the use of blockchain technology.

- We evaluate the device-level botnet detection approach using existing signature-based approaches i.e. Snort and Suricata to assess the effectiveness of the proposed scheme. The evaluation utilised three publicly available datasets namely ISOT, IoT23 and BoTIoT (available in PCAP format), firstly with default signature database and then with custom signature database representing botnet behaviour to assess detection of anomalous traffic in terms of number of alerts, detection accuracy, detection time, CPU and memory usage.

Rest of the paper is organised as follows. Section 2.4 discusses features of prominent publicly available IoT botnet datasets followed by Section 3 which presents a critical review of relevant existing work. A device level collaborative botnet detection framework is presented in Section 4. The experimental scenarios and results of applying three datasets namely IoT23, ISOT and BoTIoT are presented in Section 5. Section 6 concludes the paper highlighting future directions.

2. Background

This section presents an insight into the types of a typical intrusion detection system as well as underlying technologies that paved the way to construct our botnet detection framework. This helps readers to get preliminary understanding of the technologies including collaborative-signature detection, reputation/trust management and blockchain, that subsequently are utilized in the proposed framework.

2.1. Intrusion Detection System

An Intrusion Detection System is a software or hardware device that constantly monitors a network or system for detection of malicious activities or policy violations. Upon detection of such activity, it responds by generating an alert that can either be logged in a file or sent directly to a security analyst for further analysis and decision-making. The intrusion can be classified in several ways such as according to their scope i.e. Network Based (NIDS) that is monitoring network entry points, Host Based (HIDS) to monitor critical systems or combination of both (i.e. Hybrid). They can also be classified according to their detection approaches like Anomaly based which typically works by calculating the baseline profile of a network or system in order to detect malicious/abnormal activity or utilize a machine learning algorithm to classify an intrusion. The misuse/rule/signature based IDS contains intrusion's signature that consequently yields efficient detection of known attacks. In the past few years, an intrusion detection strategy known as collaborative intrusion detection has emerged and has already shown significant improvements in the efficiency of the ID model especially in resource-constrained IoT environments. The collaborative intrusion detection distributes the detection load on multiple devices also known as Intrusion Detection Agents (IDAs) that may not have the capability to perform such a task. This consequently results in efficient and effective detection of malicious activities within an IoT environment.

2.2. Blockchain

The blockchain has been evolved as one of the widely adopted Distributed Ledger Technology (DLT) that supports de-centralization mechanism in a distributed environment. The records are saved in a chain of blocks with the first block (aka Genesis block) containing Smart contracts (SC) and chaincode to control the actions within a blockchain. The blockchain was initially proposed to store financial transactions but with the addition of smart contracts, it is widely applied for development of Decentralized Applications (DApps). The processes in a typical blockchain are controlled by an underlying consensus algorithm and a group of devices/ entities called minors or validators. These minors validate each transaction that consequently results in a consensus amongst them to commit and store the transaction into a block. Typically a blockchain can be classified as public or private. Public blockchain is often called permissionless blockchain in which any node willing to participate is allowed after implementation of relevant protocol. Whereas, in a private blockchain, sometimes referred to as permissioned blockchain, participation permission is typically granted by a centralized controlling authority [12, 13].

Blockchain technology can greatly enhance a signature-based botnet detection process as well as a Trust Management System (TMS) in an IoT environment where multiple IoT nodes collaborate with each other to find out an intrusion with trustworthiness in several ways:

- **Decentralization:** In an IoT environment, a decentralized architecture provided by the blockchain can make the signature-based botnet detection process and trust management system more resilient to attack, as the signatures and trust values are not stored in a single location that can be compromised.

- **Immutability:** Blockchain provides immutability, once data is added to the blockchain, it cannot be altered. This can be very useful as it ensures that the signatures and trust values used for botnet detection and trust management respectively are tamper-proof and cannot be changed by an attacker.

- **Transparency:** Since all transactions on the blockchain are visible to all parties. This can be useful in a signature-based botnet detection process and trust management system in an IoT environment, as it allows for easy auditing of the system to detect and track botnet and all the iot devices can have a clear view of the transaction.

- **Consensus mechanism:** Consensus mechanism ensures that the data on the blockchain is accurate. This can be useful for both botnet detection process as well as TMS in an IoT environment, as it ensures that the signatures and trust values used for botnet detection and trust management respectively are accurate and have not been tampered with.

- **Tamper-proof signature and trust databases:** By using blockchain technology to store the hash of the signature and trust values database, it becomes tamper proof and can prevent the databases from corruption.

- **Automation through Smart contracts (SC):** The SC can be used to automate the process of botnet detection, trust management, and updating the signature and trust databases and the corresponding hash stored on the blockchain. When a new signature is added or an existing signature is updated, or when a new trust value is added or an existing trust value is updated. This method eliminates the need for manual intervention and reduces the risk of unauthorized updates.

- **Collaboration:** Blockchain allows multiple parties to access and share data on the blockchain, which can be useful in an IoT environment, as it allows multiple IoT devices to collaborate on botnet detection and trust management efforts and share threat intelligence.

The blockchain technology empowers IoT security by creating tamper-proof ledger where shared records can be stored with the consensus of others without the need of central management of control. The blockchain powered IoT network allows network devices to autonomously perform its action where the chained arrangements within blockchain ledger allows tracking of stored records. The implementation of blockchain within the IoT network is always a challenging task due to the typical resource-constrained nature of these systems. The private blockchain is more suited in an IoT environment due to the fact that they can scale to the organization's business needs which usually consists of lesser nodes than a public blockchain may have. The permissioned-private blockchain is typically owned centrally and each node wanting to participate in the network will be authenticated before a permission is granted. Furthermore, the central authority can define the rules, underlying consensus mechanism or delete any malicious activity.

Specifically, for saving the Trust values and intrusion signatures use of public blockchain will consequently add more computational as well as processing cost. Furthermore, private (permissioned) blockchains are well suited for proposed botnet detection framework since any unknown or malicious adversary with sufficient resources can not directly participate within the network.

2.3. Trust Management System

Trust Management Schemes (aka Reputation Management Schemes) have gained significant attention of the researchers especially with the exponential increase and reliance on IoT devices. Since a typical IoT network consists of heterogeneous nodes / devices that interact with each other for accomplishment of a task. The trust management schemes are employed in a network to automatically assess the reputation of a user / node to determine a measure of trust that other users / nodes may have to start interaction with them. The trust is calculated based on various factors including initial belief, observed behavior, and run-time feedback. The individual opinion (aka local trust) of a node about other peers is often collected to create an aggregated trust matrix that contains the global reputation of every node within the network. These global trust values influence the overall network that how, or with whom, a node / user can interact [14, 15].

2.4. Public Botnet Datasets

The evaluation of an IDS model using credible datasets plays a significant role in assessing the effectiveness of proposed intrusion detection framework. This section provides a brief insight into some of the important publicly available IoT botnet datasets. This is envisaged to help researchers in selecting a suitable botnet detection dataset according to available attack types, traffic type, features used, and training strength in terms of number of malicious-benign instances provided by a dataset.

2.4.1. *ISOT Botnet Datasets*

The ISOT dataset [9] combines multiple malicious and benign datasets mainly targeting HTTP based Peer-to-Peer (P2P) botnets. The malicious datasets contain traces of Storm and Waledac botnets which were extracted from Honeynet project [16].The dataset consists of more than 1.6 million instances with a malicious benign ratio of 96.66 and 3.33 % respectively. The instances of benign dataset were extracted from Ericsson's Traffic Lab [17] and Lawrence Berkeley National Lab (LBNL)[18].

2.4.2. *BOT IOT Dataset*

The Bot-IoT [10] is a labelled dataset, with diverse attack scenarios, generated from realistic testbed. It is a publicly available dataset(in both pcap and csv formats) containing around 72M instances each having 46 features. The dataset overcomes the cons of various available datasets including lack of reliably labeled data, poor attack diversity such as botnet scenarios and missing ground truth. The features of the dataset are analyzed using correlation coefficient and joint entropy approaches. The dataset consists of normal traffic by activating *Ostinato*, so that the generated data resembles normal traffic. This prevents the botnet malware from detecting virtualized environments, and attack traffic using attack software in a virtualized environment. The testbed uses four machines having kali Linux OS to simulate DoS, DDoS (HTTP, TCP, UDP), port scan for service or OS scans, key-logging and data ex-filtration attacks. The existence of IoT devices, running MQTT, in the network is ensured by using a middle-ware known as Node-RED in a virtual network. The virtualized environment has the advantages of low cost implementation, portability with ease of setup and recovery of devices since it prevents real machines to become a zombie and be a part of Botnet. On the contrary, this type of environment does not allow to launch deeper attacks especially in the firmware and hardware thereby limiting the ways an attack could be launched on such devices. The dataset contains five IoT scenarios including weather station, smart fridge, motion activated lights, remotely activated garage door and a smart thermostat which uses MQTT protocol to generate data.

2.4.3. *NBaIoT Dataset*

Meidan et al. [19] presented a botnet data set named NBaIoT that employs deep auto encoders for behavioral analysis for detection of anomalous traffic within IoT network. The detection process takes place by taking the snapshot of benign IoT traffic of nine commercial IoT devices for detection of two variants of botnets i.e. Mirai [20] and Bashlite[21] (that includes Gafgyt, Q-bot, Torlus, Lizard-Stresser and Lizkebab). The dataset contains more than 7 million instances with 115 features that can be classified into eleven different classes including ten attack classes and one benign class.

2.4.4. *IoTID20 Dataset*

I.Ullah and Mahmoud [22] presented a dataset which is generated in a testbed environment that consists of multiple devices including security camera, AI speaker and smart phone. The dataset contains 625784 network flows with a malicious benign ratio of 93.6% and 6.4% respectively. Each flow in the dataset contains 12 features that can be used to classify four attack classes (DoS, Mirai, MITM, & scans) and one benign class. These four attack classes can further be divided into syn/HTTP/UDP flooding, Brute force, ARP spoofing, and port/OS scans.

2.4.5. *Anthi Dataset*

Another publicly available IoT dataset is by Anthi et al [23]. In this dataset, 2 million instances with 135 in number features are extracted from 8 IoT Devices including Amazon Echo Dot, Belkin NetCam, TP-Link NC200, Hive Hub, Samsung Smart Things Hub, TP-Link SmartPlug, Apple TV, and Lifx Smart Lamp.Different fields (and a payload information) extracted from 5 different layers (i.e. Physical, DLL, Network, Transport and Application) with six classes of attacks including DoS, DDoS, MITM, Spoofing, Insecure Firmware, & Data Leakage. The dataset is labelled through a 4-step process that includes feature selection, device profiling, malicious-benign classification and attack type classification.

2.4.6. *IoT23 Dataset*

The IoT23 dataset [11] is a labeled dataset funded by Avast and developed in Stratosphere Laboratory, AIC group, FEL, CTU University, Czech Republic. The dataset consists of twenty three captures (20 malicious and 3 benign) which simulates various botnet attacks including Mirai, Torii, Trojan, Gagfyt, Kenjiro, Okiru, Hakai, IRCBot, Hajime, Muhstik, Hide&Seek etc. The dataset is generated in realistic but controlled IoT environment. The 280.77 million flows are labeled with the help of Zeek out of which 213 millions are Partof HorizontalScan, 47.38 million for Okiru and 19M DDoS flows.

Table 1
Publicly available datasets

Name	Format	Size	No. of Records	Attack Types	Features	Data Types	Environment	Publisher	Year
ISOT	PCAP	1.74 GB	1.67+M unique flows	HTTP Botnet	49	Network (App Layer)	Testbed	University of Victoria	2017
Bot-IoT Dataset	Pcap, argus, csv	PCAP (69.3GB) CSV (16.7GB)	72M	DoS, DDoS (TCP, UDP, HTTP), Services scan, OS Scan, Keylogging, Data ex-filtration attacks	46	Network	Testbed	UNSW Canberra Cyber	2018
N_BaIoT	CSV	-	7062606	Mirai and BASHLITE (10 attack classes, 1 benign Class)	115	Network	ReaL (9 Commercial IoT Devices)	Maiden et al.	2018
Anthi Dataset	Arff	977MB	2M Malicious-Benign Ration 50-50%)	DoS,DDoS ,MITM, Spoofing, Insecure Firmware, Data Leakage	135	Network	Real (8 Devices)	Anthi et. al	2019
IoTID20	CSV	294MB	625784	DoS, Mirai, MITM, Scan	12	Network	-	OntarioTech University	2020
IoT-23	Pcap, csv	21GB 8.8GB (Lighter Ver)	-	Mirai, Torii, Gagfyt, Kenjiro, Hakai, IRCBot, Linux.MIrai, Linux.Hajmi, Muhsitk, Hide and Seek, Trojan, Okiru	21	Network (Application layer protocols)	Real (23 Devices)	Avast, AIC group, CTU	2020

3. State-of-the-art within Device Level Intrusion Detection for IoT

This section presents a critical insight into the state-of-the-art IoT botnet intrusion detection strategies (refer to Table 2, 3, and 4 for details) with their strengths and weaknesses so as to devise an effective botnet detection framework for timely and efficient detection of malicious attacks. In order to achieve a comprehensive understanding, we begin with the article extraction process by searching from four most popular digital scientific libraries including IEEE Digital Library, Springer, Elsevier & ACM with the following search terms:

(IoT OR Internet of Things) AND (Device Level Intrusion Detection OR Device-level Intrusion Detection OR Host-based intrusion detection OR Node-based Intrusion Detection)

Since the research on intrusion detection within IoT especially for detection of botnet received significant attention when a botnet named "Mirai" targeted the vulnerable IoT devices having weak password settings and overwhelmed more than 6,00,000 devices in 2016 [20]. Therefore, in order to extract device level ID approaches from the post Mirai era, the duration has been set to last five years (between 2016 to 2021) that initially yields 2710 articles without any exclusion criteria. The basic search to the four digital libraries also reveals that a large number of studies are irrelevant such as articles from the fields of social sciences, Energy, business management, material science, environmental science etc., due to the massive adoption of IoT in various fields. Therefore, in order to refine the search, an exclusion criterion has been built to remove irrelevant (such as articles which are not focusing on intrusion detection, review articles, posters, book chapters etc.), redundant and non-English literature. The articles which are published in the relevant field such as computer science/engineering, IoT, security etc, and relevant areas (Intrusion detection within IoT environments) were taken for initial assessment. Since in this study, we are focusing on only those studies in which the IoT device itself participates in the intrusion detection process. Therefore, the search has been refined by eliminating those studies in which the intrusion detection process solely relies on edge devices, central servers and nodes other than IoT. As a result, 23 studies were selected for review and gap analysis in which participation of IoT devices was found for detection of an intrusion. These studies were further divided into three major categories. The first category contains 9 articles and consists of those approaches which employed as standalone model. The second category consists of distributed and decentralised approaches that contains 7 articles. Finally, remaining 7 articles fall into the category in which a hybrid approach is used for detection of intrusion.

3.1. Standalone Approaches

This section presents a review of various device-level intrusion detection approaches which have been summarised in Table 2. Murali and Jamalipour [24] present the ABC-based lightweight ID classification model to detect Sybil attacks within a resource-constrained RPL environment that is evaluated for static and mobile RPL in three Sybil attack scenarios using the Cooja simulator. Their results in terms of accuracy, specificity, sensitivity, and F-scores in the case of mobile RPL are lower because of misinterpretation of mobile malicious nodes as benign. Similar approach is presented by Qureshi et al.[25] present an anomaly-based IDS where ABC is employed to optimize the weights of the Random Neural Network (RNN) classifier with improved performance and an accuracy of up to 91.65% when the size of the colony is increased.

An ID framework presented by Tian et al. [26] with a deep auto-encoder technique and dimensionality reduction with ABC for optimum parameter selection of SVM classifier shows decent accuracy and acceptable FAR at the cost of

Table 2
Standalone approaches for device-level intrusion detection

Paper ID	Attack Types	Detection Method	Dataset	Network /Host Traffic	Node Participation	Protocol	Tool/ Language	Algorithm	Implementation (simulation/ Testbed)	Performance Matrics
[24]	Sybil Attack	Anomaly	No Dataset	Host	Anomaly Classification	RPL	Cooja (Contiki OS)	ABC	Simulation	Accuracy= 0.968(SA1), 0.952(SA2),0.948(SA3) Sensitivity= 0.974 (SA1), 0.935(SA2), 0.955(SA3) Specificity=0.952(SA1), 0.904(SA2),0.852(SA3) F-Score=0.972(SA1), 0.943(SA2),0.894SA3)
[25]	DoS, Prob, U2R, R2L	Anomaly	NSL-KDD	Network	Classification	Protocol Independent	Matlab	ABC, RRN	Simulation	Accuracy=91.65%
[26]	-	Anomaly	UNSW-NB15	Network	Classification Feature Selection	Protocol Independent	-	SVM, ABC Deep Auto Encoder	Simulation	Accuracy=90%, FAR<10%
[27]	DoS, DDoS, Reconnaissance, Information Theft	Anomaly	BoT-IoT	Network	Binary & Multiclass Classification	Protocol Independent	Python	FFN	Simulation (TensorFlow, Keras & Google Colaboratory)	BINARY CLASSIFICATION: Accuracy, Precision, Recall & F1scores are above 99.9% Info Theft: exfilteration=92.78% , key-logging = 96.82%, ACC: (DoS/DDoS = 99.41%, Reconnaissance = 98.375% Info: theft 88.918%).
[28]	DoS,U2R, R2L, Probe	Anomaly	NSL-KDD	Network	Attack Classification	Protocol Independent	Python-Tensorflow	SDPN	Simulation	Accuracy = 99.02 Precision = 99.38 Recall = 99.29 F1 Score = 98.83
[5]	Botnet (DoS)	Signature	No Dataset	Network		Protocol Independent	Bot Hunter for Detection, Snort Inline Prevention Engine	-	Testbed for detection of extrusions	Infected Devices found= 41 C&C Server=65 Egg downloads=32 Outbound Scanning=24
[29]	Impersonation	Behavior	No Dataset	Host	Mote module to calculate behavior	Zigbee	Labview	CNN	USRP based testbed	Accuracy = 98.8
[30]	DoS (black Snarfing, Power Draining attack)	Behavior	No Dataset	Network	Master Slave (hierarchical architecture)	blacktooth	Wireshark, Smote	C4.5, Adaboost, SVM, NB, Jrip & Bagging	Testbed	Precision=99.6% Recall=99.6
[31]	VPN Filter, IoT Reaper	Behavior	No Dataset	Host	Direct Deployment	Protocol Independent	-	-	Testbed (7 IoT Devices consists of routers and cameras)	Accuracy (2-Class Classification) = 94% Accuracy (3-Class Classification) = 81% Detection Rate 100%, Memory consumption =5.5%

increased training time of the model. However, the framework needs further improvement to manage the enormous amount of IoT communication.

A deep learning-based approach for anomaly detection with higher accuracy is presented by M.Ge at al. [27] for the detection of multiclass attacks with 90% except for multiclass classification of information theft which comes out to be 88.9%. A similar approach is presented by Otoum et al. [28] using Stacked-Deep Polynomial Network (SDPN) as an attack classifier extractor and Spider Monkey Optimization (SMO) metaheuristic to select the optimal features within the dataset with an accuracy of 99.02% with F1-score of 98.83% and precision and recall of 99.38%, 98.29% respectively.

A Snort-inspired signature based Botnet detection NIDS model N-EDPS that monitors extrusions by correlating the inbound malicious alarms with outbound communication patterns is presented by Behal et al. [5]. However, this model lacks in detecting encrypted C&C channel and requires an updated rule base to efficiently detect intrusions.

Another important class of IDS utilises behaviour based approach. For example, Bassey et al.[32] present a deep learning based model to detect malicious IoT devices via RF fingerprinting which is evaluated using RF data provided by a Mote module from Crossbow Technology with an accuracy of 98.8%. S.Satam et al. [30] also present a behavioral Multi-level blacktooth IDS (MLBIDS) mainly focusing on blacktooth network of medical devices by utilising 10 features for the classification of network traffic with the detection accuracy of 99.6% to detect a DoS attacks. However, it does not address the situation when the master device is compromised. Breitenbacher et al. [31] present a Host-based Anomaly DEtection System for IoT (HADES-IoT) to monitor the behavior of a linux-based IoT device. It is a lightweight model that only utilizes a maximum of 12% CPU load with a maximum memory usage of 14% with 100% detection rate.

3.2. Distributed Approaches

There have been a number of approaches presented for IoT device-level intrusion detection, as summarised in Table 3. For example, behavior-based models presented by M.Jagadeesh Babu and A.RajiReddy [33] in which Specification Heuristics (SH) based model at each LLN (low-power lossy Network) device is applied to detect intrusion at device level. The sensitivity & specificity of the model comes out to be 92.11% & 88.22% respectively, with almost linear memory and energy consumption. Similar contribution at device level is presented by Raja et al. [34] in which a decentralized IDS is proposed. It uses blockchain based consensus mechanism to prevent Goldfinger attack. However, this approach is not effective if the intruder after compromising the node, somehow, maintains the pattern i.e. normal usage. Another behavior based distributed NIDS, Hawkware, is presented by S.Ahn et al. [35] that ulitises LSTM based ANN model to lighten memory load. It is designed to analyze as well as correlate network and device behavior for intrusion detection. However, the model is only designed to detects network at-

Table 3
Distributed approaches for device-level intrusion detection

Paper ID	Attack Types	Detection Method	Dataset	Network /Host Traffic	Node Participation	Protocol	Tool/ Language	Algorithm	Implementation (simulation/ Testbed)	Performance Matrics
[33]	Generic Intrusions	Behavior	UNSW-NB15	Network	Participation of each node	LLN	CUPCORBAN, RStudio	Specification Heuristics	Simulation	Accuracy = 0.9177 Specificity & NPV= 0.9138 Sensitivity & PPV= 0.9211
[34]	DDoS, Goldfinger attack	Behavior	No Dataset	Network	Behavioral Monitoring	Protocol Independent	-	Goldfinger Resistance Consensus	Simulation	Consensus Probability Goldfinger attack = 0.26 (peak value)
[35]	DDoS botnets, bitcoin miner, backdoor.	Behavior	Virustotal	Network	Behavioral Monitoring	Protocol Independent	Python,	ANN	Testbed (Raspberry PI 3 with tshark, ftrace, Tensoreflow)	Error Equal Rate(EER) FPR=(0.0328-0.0014) Area Under Curve=0.99+ Performance(Single input): Runtime Overhead:7.38 CPU Performance: 4033.196 cycles

[36]	DoS, Prob, U2R, R2L	Anomaly	NSL-KDD	Network	Collaborative Fog Nodes	Protocol Independent	Python (Keras, Theano) Apache Spark	Deep Learning	Simulation	Binary Classification
										Acc(99.2) , DR(99.27), FAR(0.85), Pre(99.02) Recall(99.27), F1(99.1)
										Acc(98.27)
										DR(96.5), FAR(2.57)

[37]	DoS, Prob, U2R, R2L	Anomaly	KDD Cup 99, HoneyBird .HK	Hybrid	Collaborative	Protocol Independent	Weka	DT based Semi Supervised Learning	Simulation & Testbed	Honeypot Dataset Hit Rate (92.48), Error Rate (10.5) Real IoT Environment Hit Rate (92.43), Error Rate (7.3)
[6]	Botnet(DoS, Prob, Info Theft)	Signature	BoT-IoT	Network	Classification	Protocol Independent	Generic rules for Signature detection	Corelation based FS , J48(C4.5)	Framework Only	-
[7]	Flooding, Insider Exploration, worm	Signature	No Dataset	Host	Collaborative	-	Snort	None	Simulation + Testbed	Survival Rate= 66.7%

tacks and assumes a pre-trusted environment therefore does not cope with insider attacks whose origin is from inside a device.

Anomaly based ID approaches that employ a collaborative mechanism to lighten the load on IoT devices have also been investigated earlier. Diro and Chilamkurti [36] presented AI enabled distributed Deep-Learning (DL) based model that works in a collaborative manner for the detection of small variants of known attacks in Social IoT. For binary classification, an accuracy, DR, Precision, Recall and F1 scores of greater than 99% is achieved with FAR of 0.85. for multi class classification, an accuracy of 98.27 with DR of 96.5 and FAR of 2.57 is achieved. Wenjuan Li et al. [37] presents a disagreement based semi supervised approach to label the data automatically for collaborative IDS (DAS-CIDS). Their experiments with KDDCup99, honeypot datasets, and Snort inline showed the lowest results while in real IoT environment an error rate of 8.2 is achieved. Although the overall IDS performance is enhanced in a collaborative manner, however, these approaches still suffer from insider attacks.

An effective method for creating rule for signature based botnet detection systems that creates miniscule amount of rules from large number of malicious signatures within BoT-IoT dataset has been presented by Yan NaungSoe et al. [6]. Their rule set consists of one theft rule, one DoS rule and fourteen rules to detect probe attack. However, this is a conceptual framework that needs to be further evaluated for accuracy and performance.

A trust based collaborative approach that applies a signature based mechanism on contrary to anomaly detection in [37] is presented by W.Li et al.[7]. It utilizes a consortium blockchain-based collaborative ID model for detection of insider attacks and to incrementally build a trusted database to store malicious signatures in an un-trusted IoT environment. This model though effective for known attacks, but lacks a mechanism to update the signature-base for detection of novel attacks.

3.3. Hybrid Approaches

Thamilarasu et al. [38] presented a hierarchical, hybrid and autonomous mobile agent based IDS designed for internet enabled medical devices (IoMT) to detect device and network level anomalies over a Wireless Body Area Network (WBAN). The authors achieve device-level detection by profiling each sensor device via polynomial regression model that flags activities above a preset threshold. Simulation-based evaluation showed best case detection accuracy of 97.8 % however, it decreases if the malicious adversaries are elusive in nature.

With respect to anomaly based approaches for device-level intrusion detection, Liang et al. [39] presents a novel IDS based on multi-agent reinforcement learning model where communication between multiple agents is stored on private blockchain. The model is simulated using NSL-KDD dataset, the simulation results shows performance of DNN is better than other techniques such as DTs.However, The model is still low accuracy in various types of attacks and is resource hungry, it also has complex data flow which needs to be addressed. Nandita Sengupta [40] presented a Server/device level ML based ID model for detection of anomalous traffic and hybrid encryption of data to maintain the integrity of the data in the communication medium. The classifica-

Table 4
Hybrid approaches for device-level intrusion detection

Paper ID	Attack Types	Detection Method	Dataset	Network /Host traffic	Node Participation	Protocol	Tool/Language	Implementation (simulation /Testbed)	Performance Matrics
[38]	DoS, Data Fabrication & falsification, PrivacyData Breach	Behavior	No Dataset	Device/ Network	Agent based	Zigbee, WBAN	OMNET (Castalia 3.2)	Simulation	Acc =99.9% (Nw Lvl) = 97.81% (Device Lvl) Energy Overhead = 2-7%
[39]	DoS,Prob,U2R, R2L	Anomaly	NSL-KDD	Network	Agent based	Protocol Independent	-	Simulation	Acc= 99 precision = 99 Recall= 99% (TCP)
[40]	Generic Intrusions	Anomaly	synthetic Dataset	Network	Classification	Protocol Independent	RST	Simulation	-
[41]	DoS, Prob /Host Exploit, generic	Anomaly	UNSW-NB15	Network	Cluster based	Protocol Independent	-	Simulation	Acc = 88.92%, Mean F-Measure = 79.12% ADR= 86.15% FAR of 3.8%
[42]	Mirai, Hail Marry attack, Scan	Anomaly	No Dataset	Host	Collaborative	Protocol Independent	Python	Testbed (Raspberry PI)	(Max achieved on GBT) Accuracy = 100 Precision = Recall = F1 Score =99.99
[8]	DoS (Hello Flooding, Ver No. Modification)	Signature	No Dataset	Network	Collaborative	RPL	Cooja	Testbed	Avg Power Consumption: Normal= 0.03% (TX), 0.08% (RX) Under attack= 0.35%(TX), 1.03%(RX)
[43], [44]	Multi-Stage Attacks	Hybrid	No Dataset	Network	Collaborative	6LowPAN	Cooja with Tmote Sky Motes, Powertrace	Testbed	Pkts/s Power Consumption 1 Pkt/s <2mW <2.5mW 10 Pkt/s <10mW <10mW 100/1000 Pkt/s <30mW <25mW RAM Overhead (With/without Duty Cycle) When Pkt Size 5/10: RAM 230bytes/420 bytes Rom =980 bytes

tion is performed using four different ML-based classifier i.e. SVM, RF, NN and NB and evaluation results highlighted RF classifier to have the highest classification accuracy.

Another simulation based study that uses KMC based behavioral analysis optimized by information gain is presented by Kumar et al. [41] which presents a unified ID model for detecting malicious activities within the IoT network. The IoT network is divided into multiple clusters and a cluster head, that also contains ID model, controls and routes the data for internal communication as well as sending information to the gateway node. The evaluation is done using UNSW-NB15 dataset and various DT models including C5, CHAID, CART, QUEST, etc. The intrusion detection model is designed using 13 features. The evaluation results of UIDS, when compared with C5, ENADS and Dendron models, showed better performance with an accuracy of 88.92, Mean F-Measure of 79.12, Attack Detection Rate of 86.15 and FAR of 3.8 is achieved

Ioulianou et al. [8] presented a signature-based DoS detection mechanism in RPL scenario with both centralized and distributed modules implemented using cooja simulator to detect two major variants of DoS attacks i.e. *hello flooding* and *version number modification*. The framework includes a centralized router node that runs the detection module along with a firewall and distributed IoT detectors that run lightweight module for monitoring and reporting. Authors focus on evaluating power consumption during normal and attack scenario within this approach which is critical for resource constrained devices. COLIDE [43, 44] focus on the performance implications of intrusion detection within IoT. The authors use collaboration between device and network level components to achieve detection of large scale attacks where signature based detection is adopted at sensor nodes and network-level detection is performed using anomaly based approaches. Authors reported efficiency of the approach with respect to power and memory consumption.

3.4. Discussion and analysis of literature

This section presents a critical analysis of the limitations of existing research and serves as a foundation to develop a novel botnet detection framework for identification of malicious events in an IoT environment.

- **A Centralized approach** such as Satam et el. [30] by installing IDS on the master device of each piconet and a white listing server containing the list of trusted devices. However, there is no mechanism to deal with the situation if the master device or white listing server is compromised.

- **The Standalone approaches** based on Swarm intelligence (SI) methods are widely deployed to optimize the intrusion detection process. However, Its applicability in a resource-constrained IoT environment, where much less processing, storage, memory and energy is left for employing a security mechanism, requires further efforts at higher level of abstraction to further optimize the intrusion detection process to lighten the detection load from individual IoT devices especially working in collaboration. Current approaches, such as by Qureshi et al.[25] has already made some initial efforts by applying Artificial Bee Colony (ABC) algorithm to automatically select and optimize the weights and biases of RNN classifier and evaluated using NSL-

KDD dataset with an accuracy of 91%, and Otoum et al.[28] that applies Spider Monkey Optimization approach for optimum feature selection in NSL-KDD dataset with an accuracy of more than 99%. These approaches shows promising results however, these approaches are only evaluated in **standalone mode** with outdated datasets. The effectiveness of SI-based approaches in optimizing various aspects of ID process such as weight/feature selection, classification etc. in a centralized, collaborative, or hybrid IoT scenarios still needs to be evaluated using various latest datasets (or testbeds) to further optimize the ID process.

- **Other Standalone approaches** such as S.Behal et al. [5] that present signature-based botnet detection but lacks in detecting encrypted C& C channels and Bretenbacher et al. [31] leverage standalone anomaly detection mechanism using whitelisting approach that has low CPU and memory load. However, significant device profiling time is required to establish accurate profile. Moreover, if a new process or application needs to be executed it has to be added after profiling period.

- **Distributed approaches** presented by Babu & Reddy [33] and Raja [34] are not tested in real world environments. Moreover, Diro et al. [36] present another distributed approach that deploys a collaborative mechanism to detect intrusions locally at device level with multiple master nodes at fog level to reduce the overheads at individual IoT devices. However, the approach is tested on NSL-KDD which does not represent modern day traffic. Moreover, it is not tested for energy efficiency such as processor, memory, and energy consumption. Li et al [37, 7] propose to add a trust management module within IDS to cope with insider attacks, and are tested in real world environments. However, the authors evaluate the performance using small data samples and require further evaluation using larger data samples to assess its effectiveness in real world IoT scenarios. Further, [7] utilise a signature-based detection which is not effective for zero day attacks.

- **A Distributed approach** with real world collaborative NIDS implementation presented by S.Ahn et al. [35] that monitors device behaviors and uses optimized weight quantization with ANN to reduce memory load. Although this approach has decent accuracy while detecting network attacks however, the model assumes a pre-trusted environment and is not intended to cope with insider attacks.

- **The Hybrid approaches**, such as by Thamilarasu et al. [38] and Arshad et al. [43, 44] present a collaborative architecture where the individual IoT nodes perform basic intrusion detection either by using signature or behavior based approach and a centralized powerful node that can perform resource hungry tasks such as anomaly detection. However, the performance of the model degrades if the master node or edge device fails. Moreover, the approach either only performs power/ energy evaluation or do not tested in real IoT environments which is a critical aspect to assess the real-world efficiency of the ID model. Other Hybrid models such as Liang et al. [39] Sengupta [40] and kumar et al.[41] also present simulation-based studies. However, these models have issues of low accuracy, outdated datasets or requirement of resourceful hardware that are required to be resolved before deploying and testing it in a real world scenarios.

The open challenges in the existing approaches necessitate an efficient mechanism to efficiently and effectively identify botnet attacks in a resource-constrained IoT environment. In this paper we have introduced a novel botnet detection framework *BTC_SigBDS* that uses a collaborative approach to share the load amongst multiple IoT devices, a trust Management Module to cope with internal adversaries, an updated signature based detection to increase efficiency and a blockchain based ledger to securely store malicious signatures as well as individual node's trust for effective detection of IoT botnets. The next section discusses various components of botnet detection framework along with its working mechanism. We have also performed experimentation on the device level botnet detection component to evaluate the efficiency of custom signatures.

4. Device Level Botnet Detection Framework

This section presents a discussion on the proposed botnet detection framework (*BTC_SigBDS*) and its components for efficient and effective detection of botnet attacks within IoT networks.It employs a trustworthy collaborative device-level botnet detection in which multiple IoT nodes work together for efficient and effective detection of IoT botnets. The framework is empowered with a collaborative signature-based botnet detection component for early detection of known attacks. Since an IoT-based environment typically consists of resource-constrained devices with limited resources available to employ an efficient security mechanism. Therefore, a collaborative approach will be helpful to share detection load amongst multiple IoT devices which may not be able to perform such tasks single-handedly. The trust management module helps mitigate against the increasing and complex nature of botnet attacks in a trustworthy manner and to cope with insider adversaries. Since IoT devices can potentially span across different security domains and therefore trustworthiness of information and knowledge sharing within a typical IoT environment is critical to achieve trust sharing of security events. The blockchain component to develop an immutable (tamper proof) database for storing attack signatures and trust values. As both the attack signatures and global trust values if compromised can put the whole IoT network to stake. A blockchain will be helpful since it allows amendments in the ledger/ database after the consensus of other mining nodes ensures trust in a trustless envi-

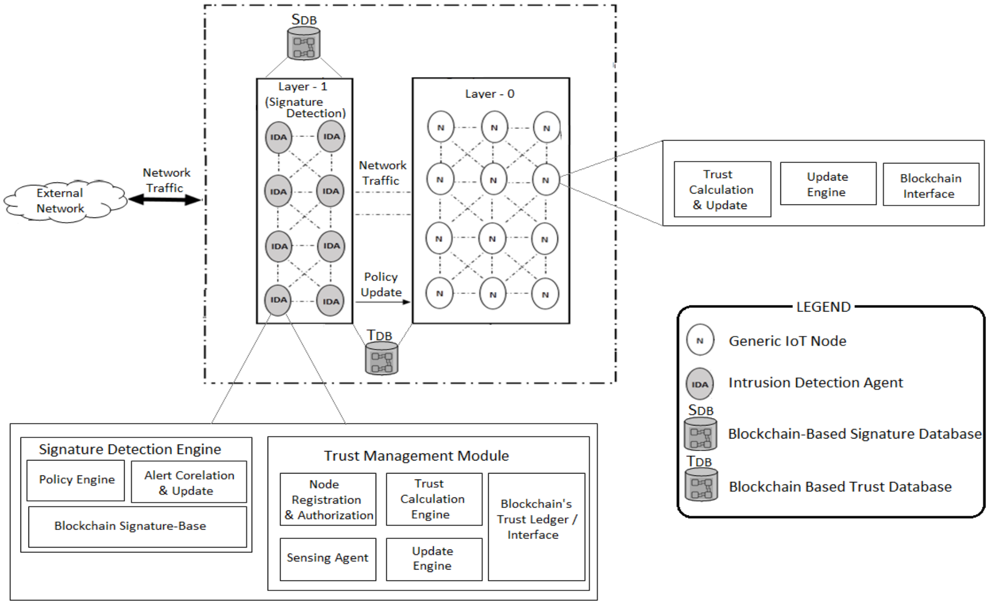

SDB
Layer - 1
(Signature Detection)
IDA
Layer - 0
N
External Network
Network Traffic
Network Traffic
Policy Update
TDB
Trust Calculation & Update
Update Engine
Blockchain Interface
LEGEND
Generic IoT Node
Intrusion Detection Agent
Blockchain-Based Signature Database
Blockchain Based Trust Database
Signature Detection Engine
Policy Engine
Alert Corelation & Update
Blockchain Signature-Base
Trust Management Module
Node Registration & Authorization
Trust Calculation Engine
Blockchain's Trust Ledger / Interface
Sensing Agent
Update Engine

Figure 1: An overview of the proposed device-level botnet detection framework

ronment especially if the trust management component of the proposed framework is somehow being compromised. Although the framework consists of various components including the Reputation management module and blockchain database for storing signatures and trust values which are briefly described below. However, in this paper, our main focus is on device-level intrusion detection and evaluation of Snort/ Suricata within IoT devices/environments.

4.1. Components

A high-level illustration of the proposed framework is presented in Figure 1. It consists of a trusted IoT environment with two layers of nodes including generic IoT nodes (H) at layer 0, and specialized IoT nodes (IDA or detection agents) at layer 1. The core of the botnet detection framework is collaborative detection of malicious activities backed by a decentralized Reputation/ Trust Management System to facilitate trustworthiness amongst IoT nodes within the network thereby addressing the challenge of misbehaving nodes. Further, we envisage using a blockchain-based distributed database for immutable storage of attack signatures. Primary components of the framework are described below:

- **Generic IoT Nodes:** The generic IoT nodes are devices such as sensors, actuators, etc. that are capable of sending/receiving data to and from network. These devices run a *trust calculation engine* to participate in calculating reputation of other devices and also forward updates to shared blockchain ledger via *update engine* and *blockchain interface*. Upon reception of any data within IoT network, it verifies the trust value (reputation) of the node from the blockchain ledger before starting communication.

- **Intrusion Detection Agents (IDA):** Specialized IoT nodes also known as Intrusion Detection Agents (IDA) are nodes that have enough computational power in terms of power, memory and storage to run a *signature detection Engine*. These devices are the most trusted IoT devices within the network they also collaboratively control and manage the de-centralized *trust management module* and perform activities including node registration and authorization, and participating in calculating and updating reputation of other devices within the environment. These nodes also manage the blockchain-based signature database as well as node's global trust values.

- **Trust Management Module (TMM):** The distributed trust management module is responsible for calculating, aggregating and interpreting the local as well as global trust values within the IoT network. It has a *Node Registration and Authorization engine* that acknowledges the registration request of incoming node by giving an initial (minimum) level of trust. The *Sensing Agent Sub-module* controls the *Blockchain's Trust Ledger/ Interface* that contains the global trust values of all nodes within the network to allow or deny any further communication with other node. It also receives the local trust values from all nodes and sends it to *Trust Calculation Engine* that subsequently normalize and aggregate local trust values to create global reputation of individual nodes and update the blockchain trust database via *Update Engine*.In future, the complete implementation along with testing of TMM along with other related components will be performed to evaluate the efficiency of proposed botnet detection framework.

- **Blockchain Ledger:** The botnet detection module will also leverage the inherent benefits of security, immutability, and trustworthiness of blockchain technology by storing important information in the blockchain's ledger. The ledger has a dual role; it maintains the integrity of the trust database by protecting the values calculated by the reputation management system as well as maintains the integrity of malicious signatures that can be used by IDA for the detection of a malicious adversary. This is accomplished by calculating the hash of both databases. Each node maintains a trust-database that contains the global trust values of other nodes within the network and the hash of the database is stored in the blockchain ledger that prevents it from corruption. Similarly, all IDAs maintain a signature database with an underlying blockchain maintaining its hash.

We have discussed the suitability of private (permissioned) blockchain in this scenario. However, the implementation and testing of the blockchain ledger is not in the scope of this study and is left for future work.

- **Policy Engine:** The policy engine has the capability to receive, interpret and distribute the rules (signatures) amongst collaborating agents (IDAs). It controls the blockchain-based signature database and is responsible for network packet capture, inspection, and matching of signatures with incoming traffic. Upon identification of anomalous traffic, it sends an alert to *alert correlation and update engine*. This paper examines the suitability of two engines (i.e. Snort and Suricata) to find out their suitability for IoT environments.

- **Alert Correlation and Update**: This module will perform a number of rich activities including alert aggregation, normalization, analysis of generated logs, and correlate amongst a number of generated alerts and sequence of events and uses predictive analytics to finalize the event of a botnet attack. This will also help in suggesting a security vulnerability. Upon identification of a malicious event, it sends its recommendations back to the policy engine that will subsequently enforce the updated policy to block any malicious traffic/IP/port/mac to all other nodes as well as inform the network administrator.

4.2. Working Mechanism

Figure 1 presents our proposed botnet detection framework (*BTC_SigBDS*) whereas Figure 2 presents an detailed insight into the working of the model. The proposed model consists of a trusted IoT environment with two layers of nodes including generic IoT nodes (H) at layer 0, and specialized IoT nodes (IDA or detection agents) at layer 1. A distributed reputation/Trust Management Module (TMM) is established within the network to ensure a trusted environment by calculating the reputation of individual nodes and updating it to a blockchain-based decentralized (Shared) ledger. Further, an additional layer of security is added by blockchain that prevents any malicious adversary to modify the trust values as well as signature-base which, if compromised, may lead to any security breach or complete system failure.

The framework ensures a multi-level collaborative environment in which multiple components (including TMM, signature detection process etc.) as well as IDAs collaborate with each other for efficient and effective detection of botnet attacks within the IoT environment. The collaboration amongst the components is done through a blockchain interface that ensures the integrity of signatures as well as node's global trust. This consequently prevents malicious/untrustworthy nodes or nodes with less reputation to sabotage the intrusion detection process and allows only highly trustworthy nodes (having reputation greater than a pre-set threshold) to take part in the botnet detection. Moreover, since the trust and signature databases are stored on a blockchain, the hash of the databases are stored on the blockchain instead of the actual trust values or signatures. This allows for the integrity of the databases to be verified without having to store the entire database on the blockchain. By comparing the stored hash value of the signature database and/ or trust database with the hash value of these databases on the IoT device, the system can detect if any changes have been made to the signature database, and if so, it can prevent it from being used, and thus preventing corruption. Then the IDA can download the legitimate database from its peer node.

Whenever a new node enters the network, it goes through a node registration process through the trust management module and is assigned an initial level of trust (T_i). All the IoT nodes (H and IDA) within the network takes part in calculating the reputation of other nodes and update the underlying ledger (blockchain) by aggregating the trust values. If a node's reputation value reaches below a preset threshold (i.e. zero), it will be blocked from the network. The botnet detection process consists of a collaborative model which mainly involves IDA nodes running a signature detection engines with underlying blockchain based database to securely store the malicious signatures. The IDAs perform signature detection and also governs the TMM module. Upon detection of an intrusion, they update each other to stop inspecting that network packet. It also sends policy updates to both IDA & H. These IDAs also take part in aggregating the local trust value to transform the local trust into global reputation while updating the trust ledger (blockchain) for each node in the IoT network. This collaboration amongst these IDAs results in creating a trusted environment as well as efficient and effective detection of botnets within the network.

The Signature detection engine and TMM interact with each other via blockchain interface. Each time the detection engine receives detection information from an IDA, it will first check the reputation of the sending agent in the blockchain based trust ledger before accepting the information and updating other participating nodes within the network. Thereby allowing only trusted agents to update the blockchain ledger. An IoT device with sufficient resources (see Figures 8 & 9 for peak memory and CPU usage respectively) and a reputation or global trust value greater than an specified threshold ($G_T \geq$ T) will be allowed to become IDA and participate in the botnet detection process. On the contrary, if the reputation falls below the preset threshold (i.e.

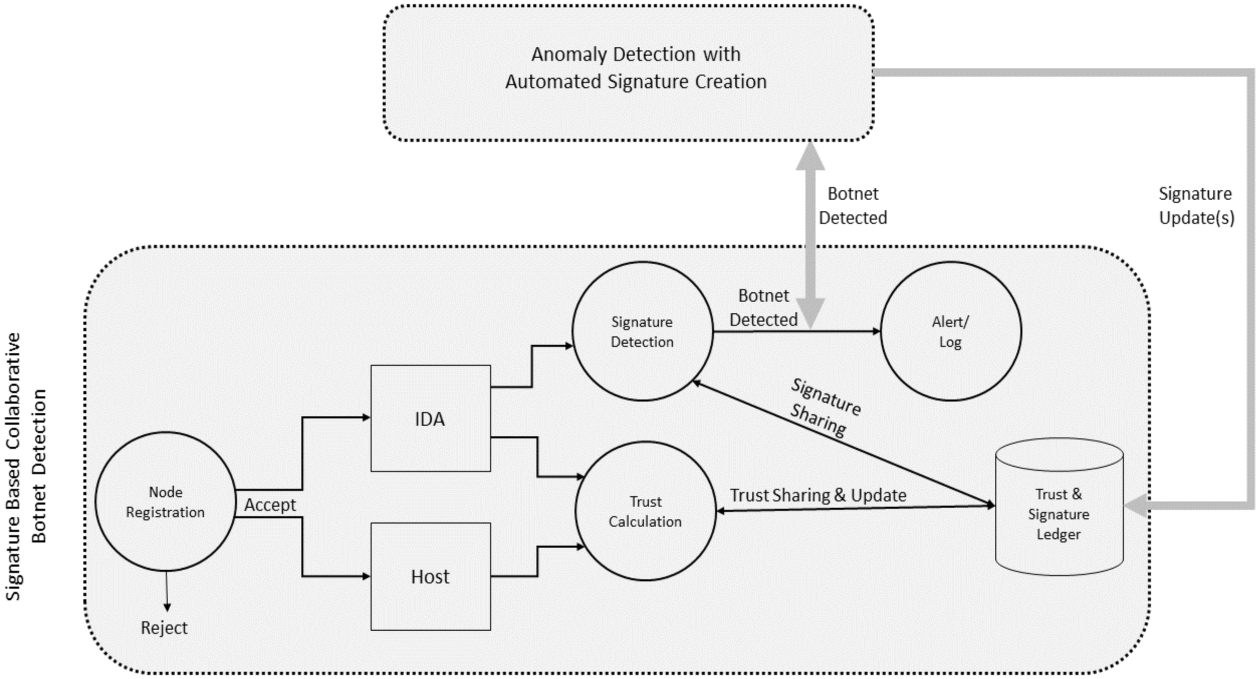


Figure 2: Process model for the proposed botnet detection mechanism

G_T<T), the IDA is then prevented from participating in the botnet detection process until it regains the desired reputation. However, it will continue to remain part of the IoT network as a generic IoT node (i.e. H), until its reputation is greater than zero.

In this study, we have created signatures for identification of malicious botnet activities (for details see Section 5). In order to automate the signature detection process, an anomaly detection with automated signature creation component is added in Figure 2 for efficient and effective detection of botnet attacks. As an anomaly detection module, running a Machine Learning classification algorithm can identify novel attacks, whereas the signature creation module automatically creates signatures for identified attacks and updates the signature database. This also leads to detection of this type of attack more efficiently in future by IDA running signature detection component. However, this anomaly detection and automated signature generation is not in the scope of current study and left for future work.

4.3. Threat Model

Protecting the IoT ecosystem with the help of the ID process necessitates a slight distinct security requirement due to its inherent features including ubiquitous connectivity, heterogeneous nodes, & resource-constrained device thereby adding more vulnerabilities to such an environment. This section contains a discussion on various scenarios that pose a threat to a typical IoT environment and how the devised framework would react to them or what are the possible solutions.

Single internal Malicious Adversary: The scenario, when a malicious individual/node entered the network or a legitimate IDA is somehow being compromised or infected. As a result, it always sends wrong detection information to other IDAs to sabotage the whole intrusion detection process.

Discussion: If the model does not include a trust management mechanism, a malicious adversary will succeed in deceiving the detection process, especially if it is able to infect other agents (IDAs). However, if our Trust Management Module (TMM) is activated, the trust value of malicious individuals (IDA) will decrease at every wrong information (i.e. false detection information) that it propagates and ultimately the malicious adversary will be thrown-out (blocked) from the network. Since the presence of TMM will never allow the infected/malicious individuals to receive high trust values from other IDAs it may strive to increase it by infecting other legitimate IDAs to make a team of infected nodes.

Collusion amongst Multiple Malicious Adversaries: The scenario, when a malicious collective is formed and starts a secret cooperation in order to deceive legitimate nodes as well as the botnet detection process. The collusion amongst these malicious nodes would result in receiving positive trust values from other malicious agents thereby increasing the reputation of malicious collectives. This subsequently results in increased reliance on malicious nodes for data sharing as well as in the botnet detection process. detection sabotages the botnet detection process as well as the whole network.

Discussion: In our framework, if a collusion amongst

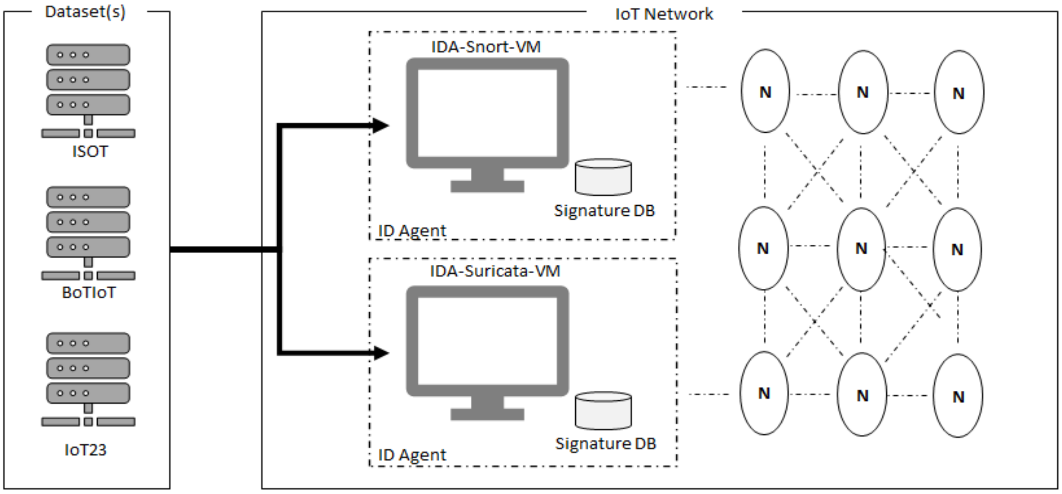


Figure 3: Evaluation Topology

multiple malicious IDAs is formed. Even if they give each other a positive trust value, still they need to provide enough correct detection information to maintain their trust values above the threshold. This is very unlikely for the nodes having malicious intent to send correct detection information. Our future work involves empirical investigation of how our framework behaves in present of malicious collectives especially how much correct detection information it gets from these infected agents (IDA)

Trustworthy Vs Malicious Signatures: The scenario when an IDA wants to include a malicious signature in a blockchain ledger. This would result in wrong detection information. Furthermore, if a large number of malicious signatures are added to the ledger, the detection engine may allow/or block traffic which may not need to be allowed/blocked. Moreover, a large number of malicious signatures may also consume more memory and CPU usage which is critical especially in a typical resource-constrained IoT environment.

Discussion: When our framework is activated, the signatures are maintained in a blockchain ledger therefore addition of any signature can only be possible upon consensus of majority of trusted IDAs within IoT network and a single IDS or malicious collectives that constitutes less than 51% of total agents can never add malicious signatures to the signature database. Our model also ensures that no malicious node or collective is formed as discussed in the threat model of worm spreading agents.

Ledger Based Attacks The scenario when, integrity of blockchain ledger is somehow compromised.

Discussion: This is a threat scenario that is not addressed by *BTC_SigBDS*. Since our framework assumes that the integrity of ledger is maintained by blockchain and therefore it does not consider ledger based attacks.

Worm Spreading Agents (IDAs) Scenario when Malicious IDAs spread a worm and infect other legitimate nodes and try to built malicious collectives. This most threatening scenario since the may lead to establishment of a bot-army (zombie network) to launch devastating DDoS attacks.

Discussion: This is a threat model when an infected agent enters the network and sending malicious codes to infect other legitimate nodes. since new entrants will be assigned minimum level of trust. After the activation of the framework, the node sending malicious code files will gradually be assigned negative trust value and will ultimately be blocked from the network.

Agents with Sybil Identities Scenario when a malicious IDA create thousands of sybil identities to influence signature ledger as well as network.

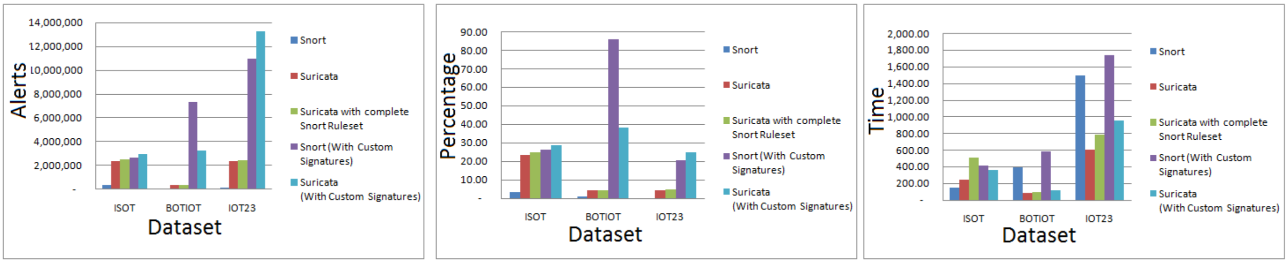
Discussion: This threat model takes the advantage of the absence of any cost to enter the network. This scenario can be averted by employing an entry cost mechanism such as imposing a mandatory capthca to enter the network. Consequently making it hard for a malicious IDA to create high number of sybil identities.

5. Experimental Setup and Results

This section presents experimental scenarios used to conduct an empirical investigation to find out the efficiency and effectiveness of IDS/IPS tools for device-level botnet detection within IoT environments. The evaluation network topology for the research, as shown in figure 3, was designed to assess the device-level botnet detection system for the Internet of Things (IoT). Two intrusion detection/prevention systems (IDS/IPS) tools, namely Snort and Suricata, were used to perform experiments. These tools were installed on the Kali Linux operating system and ran in a virtual environment. The use of a virtual environment was crucial in order to simulate a resource-constrained environment, which is typical for IoT devices. To achieve this, each virtual machine was allocated a processor with 2 cores and 2 GB of memory. This ensured that the experiments were run in a controlled environment, allowing for a consistent evaluation of the intrusion detection systems. Out of several publicly available datasets, as shown in Table 1 IoT23, ISOT, and BoTIoT have been used in experimentation. These three datasets are selected (out of six mentioned in Table 1) due to their availability in PCAP format. Since both Snort and Suricata can

Table 5
Experimental results

Dataset	Packets	Snort			Suricata			Suricata with complete Snort Ruleset			Snort (With Custom Signatures)			Suricata (With Custom Signatures)		
		Alerts	%age	Time	Alerts	%age	Time	Alerts	%age	Time	Alerts	%age	Time	Alerts	%age	Time
ISOT	10,250,791	347,459	3.39	152.87	2,370,559	23.13	240.78	2,547,445	24.85	506.54	2,670,356	26.05	407.67	2,934,583	28.63	359.87
BOTIOT	8,559,054	110,718	1.29	391.77	352,946	4.12	81.56	352,986	4.12	91.23	7,331,120	85.65	578.02	3,270,352	38.21	112.46
IOT23	53,942,175	118,079	0.22	1,493.62	2,374,581	4.40	607.06	2,480,907	4.60	787.26	10,926,140	20.26	1,736.44	13,279,260	24.62	949.11



a) Number of Alerts

b) Percentage of Alerts w.r.t Total No. of Packets

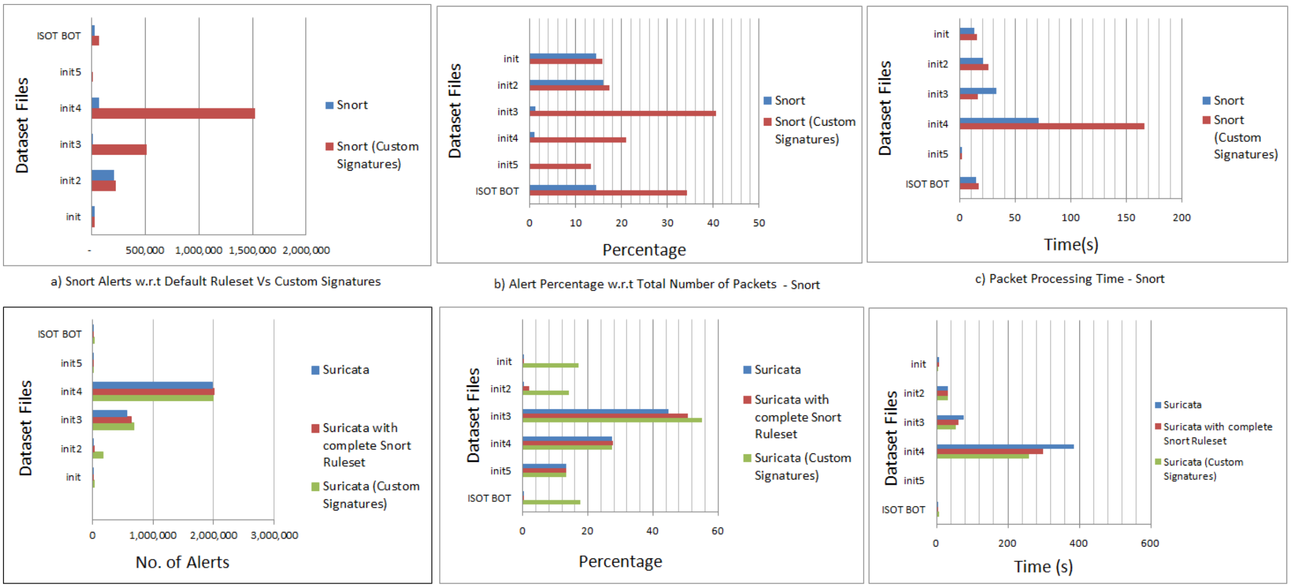
c) Processing Time

Figure 4: Comparison of Snort & Suricata results on ISOT, BOTIOT & IoT23 datasets

not process files(Such as CSV, Ariff, etc) other than PCAPs. Moreover, out of these three datasets, ISOT and BoTIoT are generated using a testbed environment whereas IoT23 is generated using 23 real IoT devices. In order to simulate a real network scenario and to evaluate the behavior of both the IDSs in real time, *TCPReplay* utility is utilized to replay the captured traffic (PCAP files) to the ethernet port as these packets were recorded. The IDS/IPS tools were used to monitor the incoming traffic from various datasets, to identify any security threats.The performance of Snort and Suricata

Table 6
Detailed results for ISOT dataset

ISOT Dataset		Snort			Snort (Custom Signatures)			Suricata			Suricata with complete Snort Ruleset			Suricata (Custom Signatures)		
FileName	Packets Analyzed	Alerts	%age	Time	Alerts	%age	Time	Alerts	%age	Time	Alerts	%age	Time	Alerts	%age	Time
init	203398	29378	14.4	13.0	32038	15.8	15.1	427	0.2	7.7	430	0.2	7.4	34439	16.9	5.8
init2	1313543	209081	15.9	20.7	227456	17.3	25.3	828	0.1	32.2	25549	1.9	33.2	183222	13.9	33.2
init3	1263913	13565	1.1	32.4	513271	40.6	15.6	563460	44.6	76.9	638872	50.5	62.4	692504	54.8	55.3
init4	7266524	66057	0.9	71.1	1528255	21.0	166.6	1982301	27.3	384.2	2005073	27.6	298.5	1988607	27.4	258.3
init5	15	0	0.0	1.5	2	13.3	1.2	2	13.3	0.1	2	13.3	0.1	2	13.3	0.2
ISOT BOT	203398	29378	14.4	14.2	69537	34.2	17.0	427	0.2	5.4	430	0.2	6.1	35809	17.6	7.1



a) Snort Alerts w.r.t Default Ruleset Vs Custom Signatures

b) Alert Percentage w.r.t Total Number of Packets - Snort

c) Packet Processing Time - Snort

d) Suricata Alerts w.r.t Default Ruleset Vs Custom Signatures

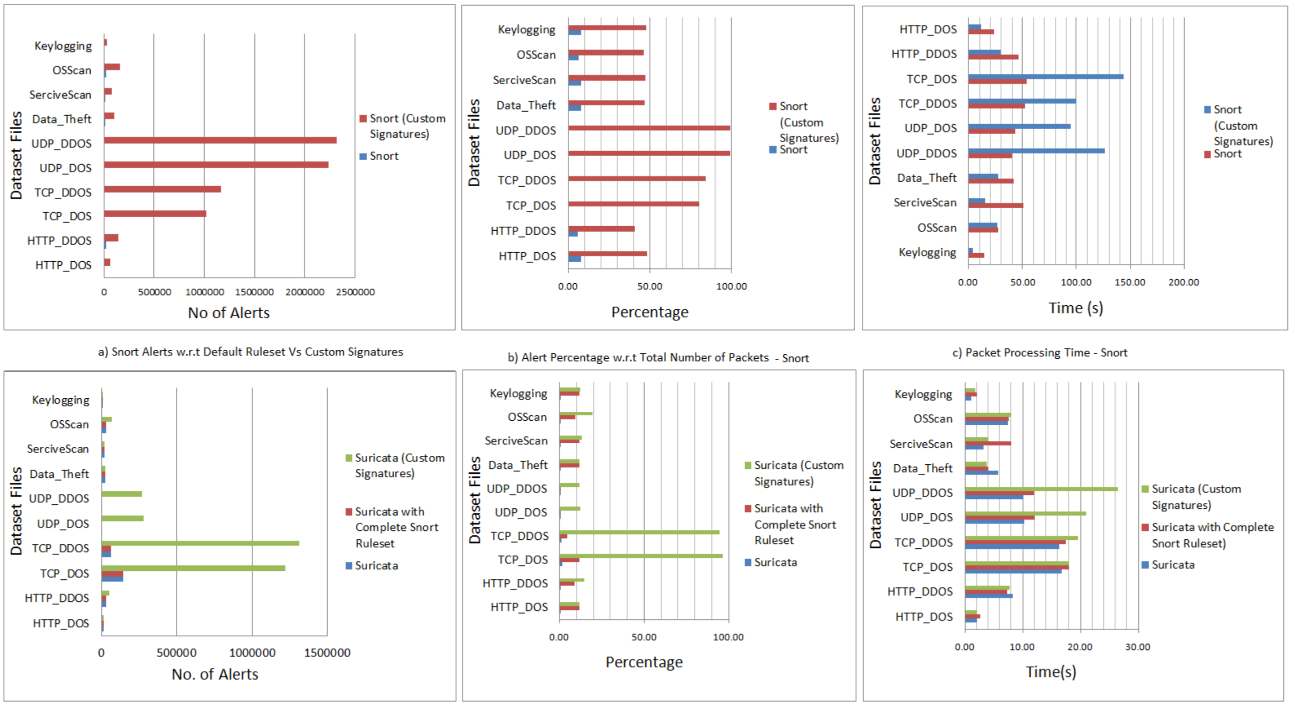
e) Alert Percentage w.r.t Total Number of Packets - Suricata

f) Packet Processing Time - Suricata

Figure 5: Snort and Suricata results on ISOT dataset

Table 7
Detailed results on BOTIOT dataset

BOT IOT Dataset		Snort			Suricata			Suricata with Complete Snort Ruleset			Snort (with Custom Signatures)			Suricata (with Custom Signatures)		
FileName	Packets	Alerts	%age	Time(sec)	Alerts	%age	Time(sec)	Alerts	%age	Time(sec)	Alerts	%age	Time(sec)	Alerts	%age	Time(sec)
HTTP_DoS	129369	10370	8.02	23.28	15339	11.86	2.13	15340	11.86	2.72	62199	48.08	11.45	15340	11.86	2.14
HTTP_DDoS	357961	20765	5.80	46.00	31123	8.69	8.24	31123	8.69	7.29	145548	40.66	29.73	52099	14.55	7.80
TCP_DoS	1270992	4458	0.35	53.56	146757	11.55	16.79	146619	11.54	17.99	1019580	80.22	143.29	1222328	96.17	17.99
TCP_DDoS	1391827	3190	0.23	52.00	63827	4.59	16.38	63804	4.58	17.43	1168131	83.93	99.42	1311721	94.24	19.53
UDP_DoS	2262813	10244	0.45	43.00	6264	0.28	10.24	6265	0.28	12.08	2241278	99.05	94.26	277794	12.28	20.94
UDP_DDoS	2339194	3645	0.16	40.35	5222	0.22	10.21	5223	0.22	11.95	2316398	99.03	126.36	267371	11.43	26.50
Data_Theft	218833	17052	7.79	41.47	25226	11.53	5.72	25226	11.53	4.02	102505	46.84	27.46	25388	11.60	3.80
SerciveScan	170008	13237	7.79	50.55	19347	11.38	3.29	19416	11.42	8.04	80421	47.30	15.25	22481	13.22	4.02
OSScan	352142	22513	6.39	27.39	32084	9.11	7.45	32213	9.15	7.65	163505	46.43	26.68	67963	19.30	7.98
Keylogging	65915	5244	7.96	14.16	7757	11.77	1.10	7757	11.77	2.07	31555	47.87	4.12	7867	11.94	1.76



a) Snort Alerts w.r.t Default Ruleset Vs Custom Signatures

b) Alert Percentage w.r.t Total Number of Packets - Snort

c) Packet Processing Time - Snort

d) Suricata Alerts w.r.t Default Ruleset Vs Custom Signatures

e) Alert Percentage w.r.t Total Number of Packets - Suricata

f) Packet Processing Time - Suricata

Figure 6: Snort and Suricata results on BOTIoT dataset

was measured based on their ability to detect intrusions, detection time as well as CPU and memory consumption.

All the dataset files are analyzed using Snort/Suricata using the following scenarios:

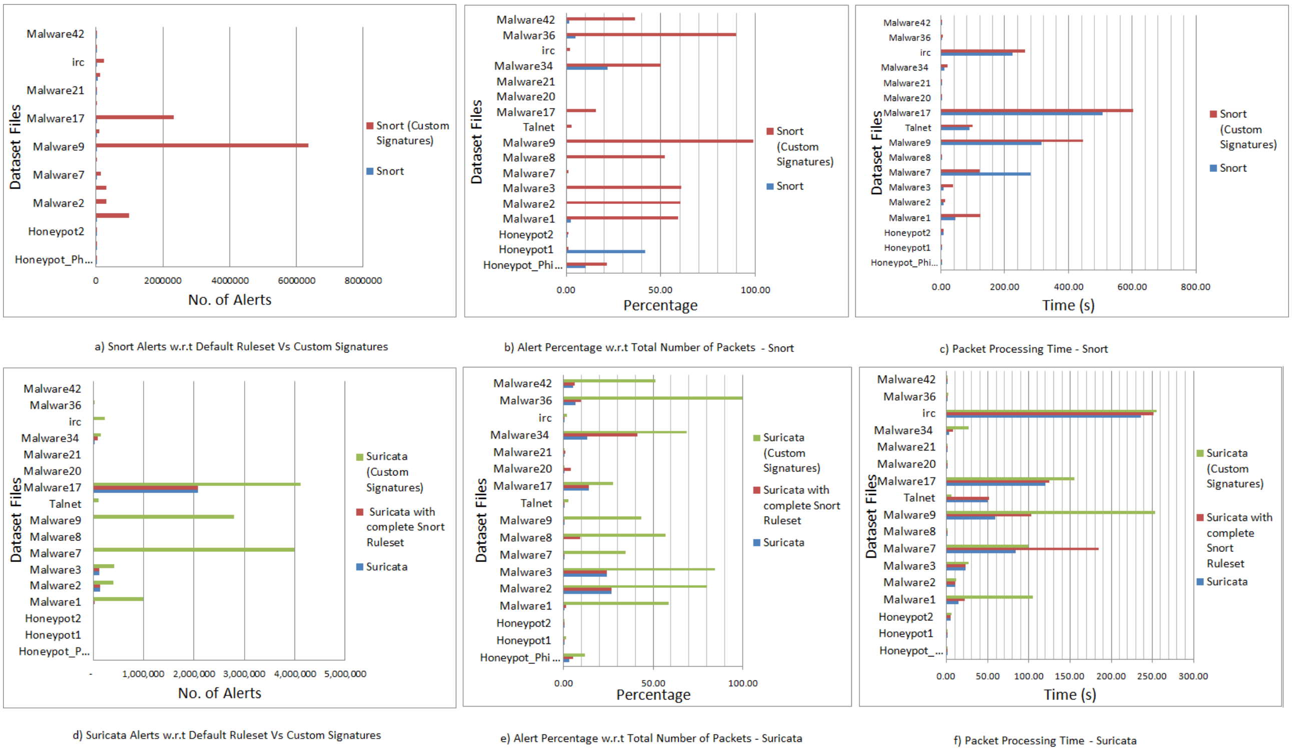
a. Snort with default ruleset

b. Suricata with default ruleset

c. Merging of Snort's signature base with Suricata's ruleset & its evaluation on Suricata.

Table 8
Detailed results for IoT23 dataset

IOT23		Snort			Suricata			Suricata with complete Snort Ruleset			Snort (With Custom Signatures)			Suricata (With Custom Signatures)		
FileName	Packets	Alerts	%age	Time	Alerts	%age	Time	Alerts	%age	Time	Alerts	%age	Time	Alerts	%age	Time
Honeypot_Philips	8,573	869	10.14	1.29	281	3.28	0.27	457	5.33	0.27	1,822	21.25	1.30	1,015	11.84	0.42
Honeypot1	21,664	9052	41.78	2.51	14	0.06	0.35	117	0.54	0.42	203	0.94	1.20	256	1.18	0.34
Honeypot2	398,312	2840	0.71	6.20	69	0.02	4.85	103	0.03	4.98	4,640	1.16	6.50	788	0.20	5.26
Malware1	1,686,291	36676	2.17	43.52	3,393	0.20	13.88	24,063	1.43	21.76	993,998	58.95	122.75	987,039	58.53	104.66
Malware2	501,356	-	-	8.18	133,444	26.62	10.20	133,444	26.62	10.90	302,167	60.27	13.27	401,377	80.06	11.11
Malware3	496,959	-	-	7.10	120,026	24	22.70	120,026	24	22.70	302,167	61	36.26	418,885	84.29	27.12
Malware7	11,508,430	7287	0.06	278.88	49	0	84.08	7,554	0.07	185.10	133,708	1.16	119.28	3,997,690	34.74	99.00
Malware8	23,623	-	-	1.64	-	-	0.24	2,162	9.15	0.45	12,334	52.21	1.17	13,440	56.89	1.06
Malware9	6,437,837	3975	0.06	312.97	820	0.01	59.11	5,197	0.08	103.16	6,363,939	98.85	445.20	2,792,779	43.38	253.19
Talnet	3,793,326	944	0.02	88.10	2,560	0	49.77	3,001	0	51.10	100,048	3	98.70	101,773	2.68	6.12
Malware17	15,000,000	3302	0.02	505.11	2,079,180	14	120.00	2,080,223	14	125.00	2,316,695	15	602.00	4,118,059	27.45	155.00
Malware20	50,156	1	0	1.45	5	0.01	0.50	2,020	4.03	0.61	30	0.06	1.20	34	0.07	0.46
Malware21	50,277	68	0.14	1.42	5	0.01	0.67	527	1.05	0.61	131	0.26	1.49	97	0.19	0.57
Malware34	233,865	50559	21.62	9.12	30,951	13.23	2.99	95,948	41.03	7.49	117,321	50.17	18.70	160,710	68.72	26.72
Malware36	13670225	377	0	224.10	12	0.00	236.60	914	0.01	251.45	235,019	1.72	262.30	236,178	1.73	255.14
irc	36,797	1752	4.76	1.37	2,501	6.80	0.38	3,587	9.75	0.73	33,041	89.79	4.12	36,601	99.47	2.18
Malware42	24,484	377	2	0.67	1,271	5.19	0.46	1,564	6.39	0.53	8,877	36.26	1.00	12,539	51.21	0.76



a) Snort Alerts w.r.t Default Ruleset Vs Custom Signatures

b) Alert Percentage w.r.t Total Number of Packets - Snort

c) Packet Processing Time - Snort

d) Suricata Alerts w.r.t Default Ruleset Vs Custom Signatures

e) Alert Percentage w.r.t Total Number of Packets - Suricata

f) Packet Processing Time - Suricata

Figure 7: Snort and Suricata results for IOT23 dataset

d. Evaluation of Snort & Suricata by creating custom signatures.

5.1. Performance Analysis

This section presents a detailed discussion on the results to evaluate the suitability of Snort & Suricata for our botnet detection framework. The Table 5 shows the summary of results and Figure 4 shows comparison of results for both the signature detection engines i.e. Snort and Suricata in terms of number of alerts, accuracy with respect to total number of packets analyzed and detection time. The Tables 6,7, 8 and Figures 5, 6, 7 show detailed evaluation results of ISOT, BOTIoT, and IoT23 datasets respectively.

Testing of Snort & Suricata with default rulesets: As shown in the Figure 4 and Table 5, Snort shows a percentage/number of alerts 3.39% (0.34M alerts), 1.29% (0.11M alerts) and 0.22% (0.118M alerts) whereas Suricata with the default signature base generates higher number of alerts i.e. 23.13% (2.37M alerts), 4.12% (0.353M alerts) and 4.4% (2.37M alerts) for ISOT, BoTIoT, and IoT23 datasets. However, the initial experiment yields that both Snort and Suricata with their default signature base show extremely poor results and generates minuscule number of alerts. Suricata has relatively better number of alerts but still shows poor performance to detect botnet attacks. The results also show that Suricata has in general (specifically for BoTIot and Iot23 datasets) lower processing time than its counterpart. This is due to its multi-threaded nature whereas the Snort contains a single-threaded processing engine and utilizes only a single core of the processor. Moreover, Suricata also has a higher processing time in the ISOT dataset, this is probably due to the processing of a higher number of alerts.

Merging of rulesets of both tools and its evaluation in Suricata: The initial experiments with the default dataset reveals that the Suricata performs better than Snort both in terms of number of alerts and detection time. Therefore, the experimentation has been extended by adding Snort's Signature base (aka ruleset) to the default Suricata's signature base. This shows further improvements in detection which has improved from 23.13% (2.37M alerts) to 24.85% (2.54M alerts) for ISOT and from 4.4% (2.37M alerts) to 4.6% (2,48M alerts) for BotIoT dataset but with increased processing time. This is obvious since the detection engine needs to compare more signatures while processing the packets. Moreover, merging of the signature base does not significantly improve the results on the IoT23 dataset.

Evaluation of Snort & Suricata using custom signatures: Although the results are improved in previous experiments when the signature-base of both the tools is combined and executed through Suricata. However, these results need further improvements in order to achieve a decent detection rate. Consequently, the experiments have been extended by creating custom signatures keeping in view of the botnet's behaviour to find out the following abnormal activities:

- TCP/UDP Flooding attempts
- Ping of Death attempts (ICMP Flooding)
- Syn Flooding attempts
- IRC Communication
- File Download attempt

- Brute Force attack

The evaluation results on ISOT dataset (see Table 6 and Figure 5 for details) using Snort show significant improvements with a peak alert of 1.5 million which was more than the total alerts generated (around 0.34 million) with default ruleset with maximum processing time of 298.5s. Suricata performs better and reaches a peak alert of 2.0M with a maximum processing time of 258.3 seconds. For BoTIoT (see Figure 6 and Table 7 for details), both the engines performs better for TCP and UDP based Dos/DDoS attacks with a peak alert percentages in 90s. Figure 7 and Table 8 show detailed results on IoT23 dataset, both engines show decent improvements with peak alerts of more than 6.0m and 4.0M respectively. Moreover, the Suricata appears to be more efficient in terms of less processing time to generate these alerts.

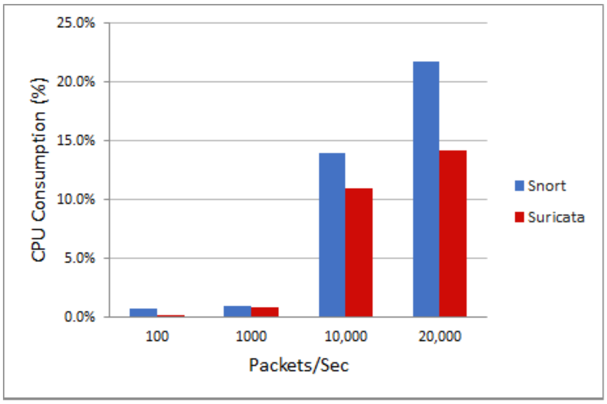


Figure 8: Snort and Suricata peak CPU usage

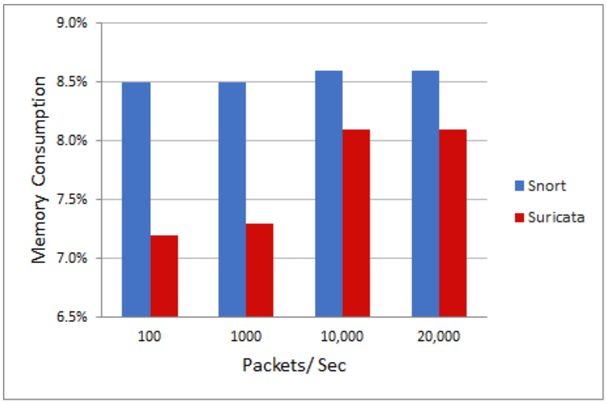


Figure 9: Snort and Suricata peak memory usage

5.2. Memory & CPU Overheads

Since memory and CPU consumption are two factors that are of prime importance, especially in a typical resource-constrained environment, therefore, experiments were performed to assess CPU and memory consumption to find out the suitability of Snort and Suricata for the proposed botnet detection framework. Figures 9 & 8 show the evaluation results of memory and CPU usage of Snort and Suricata respectively. The experiments were performed by evaluating both IDS with different packet rates (i.e. 100, 1000, 10,000 & 20,000 Pkt/s). The results in Figure 9 demonstrate that the increase in the packet rate has a minuscule effect on memory consumption and Suricata consumes slightly less memory than Snort. In terms of CPU consumption, as shown in Figure 8, the Snort hit the peak CPU usage of 21.7% while Suricata consumes almost 14.2% of CPU resources when the packet rate reaches 20k per second. The CPU and memory consumption of both tools shows their suitability in a resource-constrained IoT environment. Moreover, the Suricata is more suitable for multithreaded and multi-core CPU environments due to its support whereas Snort only supports processing the packets utilizing a single core of the CPU (processor).

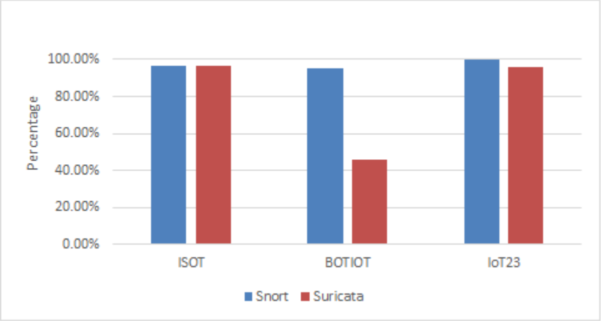


Figure 10: Snort and Suricata peak Accuracy

5.3. Accuracy

Figure 10 depicts the peak accuracy of both snort and suricata on three different datasets. The results showed that for the ISOT dataset both suricata achieves a peak accuracy of 96.53%. However, when evaluating the BotIoT dataset, Snort showed better performance, with a peak accuracy of 94.98% compared to Suricata's 46%.In the IoT23 dataset, Snort demonstrated the highest peak accuracy of 99.95%, surpassing Suricata's 95.59% accuracy.

A comparison of accuracy between our research and previous studies is presented in Table 9, which clearly illustrates the dominance of our study over the others. While previous studies either relied solely on simulation, evaluated a single attack, or used outdated datasets. Out of a number of surveyed studies only one study [31] achieved 100% accuracy. However, the study evaluated a host-based technique, not a network-based intrusion detection system, Furthermore, other studies like [43] and [44] focused on ram overheads and [35] evaluated CPU cycles. On the other hand, this study evaluated several key performance indicators, including accuracy, number of intrusion alerts, time consumption, CPU usage, and memory consumption.

5.4. Discussion and Future work

This paper mainly focuses on the implementation of device-level intrusion detection system by evaluating the suitability

Table 9
Comparison of Peak Accuracy of other research with this study

Paper ID	Peak Accuracy
[24]	96.8%
[25]	91.65%
[26]	90%
[27]	99.41%
[28]	99.02%
[29]	98.8%
[31]	81%
[33]	91.77%
[36]	98.27%
[38]	97.81%
[39]	99%
[41]	88.92%
[42]	100%
This Study	99.95%

of Snort and Suricata in IoT environment both in terms of resource consumption (i.e. CPU and memory consumption) as well as efficiency (in terms of detection time and number of alerts generated). We have also created custom signatures to strengthen the capabilities of both tools in order to increase their efficiency against botnet attacks.

- The evaluation results show the potential of both Snort and Suricata to be applied in an IoT node as device-level BDS both in terms of efficiency and memory/CPU usage. The Snort is more suitable in a single core CPU based machine whereas Suricata is more suitable for devices having multi-core CPUs due to its support-ability of multi-threaded processing environments. Moreover, the evaluation results are significantly improved after adding custom signatures in some cases.

- Our experimental study also reveals the requirement of evaluation of ML algorithms to thoroughly analyze the labeled CSV files available with two of the three selected datasets and generation of signatures for efficient detection of botnets. Since two of these three datasets (including BotIot and Iot23) are available in both PCAP and CSV formats (see Table 1), therefore the signatures generated by ML-based algorithms can further be applied to the original PCAP files to evaluate their performance.

- The proposed framework also necessitates an anomaly detection and signature creation component as shown in Figure 2 that will utilize an edge device for anomaly detection as well as automated signature generation to update the underlying blockchain ledger. This component leverages the benefits of Machine Learning classification algorithms for detection of novel attacks and will automatically create automatic signatures. This consequently leads to efficient detection of this type of attack in future via signature detection component.

Our future work involves the implementation and evaluation of a complete botnet detection model with decentralized trust management scheme, underlying blockchain ledger, and optimized anomaly detection engine to create automated botnet signatures as shown in Figure 2. Moreover, the available labeled datasets (as shown in Table 1) will be used to train the anomaly detection module after evaluating the efficiency in terms of accuracy, F-Measure, FAR etc., to be used in anomaly detection as well as automated signature creation process. The work will also involves testing of complete botnet detection model in various scenarios (using threat models) as discussed in Section 4.3

6. Conclusions

Adoption of IoT devices within widespread domains has made them a lucrative target for adversaries - botnets are one the of most prominent threats to such systems. This paper introduced our efforts to establish a collaborative intrusion detection for IoT systems whilst focusing on device-level detection scheme to address this challenge. We presented a critical review of the existing work within device-level intrusion detection for IoT, highlighting challenges which require further work. We presented the design and implementation of device-level intrusion detection scheme which utilises signature-based detection (Snort and Suricata) to achieve light-weight operation whilst addressing limitations of such systems through the use of signature update mechanism to ensure protection against emerging attacks. Evaluation with three IoT botnet datasets (ISOT, BoTIoT, and IoT23) shows that the proposed device-level detection scheme is able to achieve 99% detection accuracy for TCP-based DoS and DDoS attacks. We envisage working on the challenge of trustworthy attack signature sharing among IoT nodes by developing a robust decentralised reputation system which can ensure protection against tampering of attack signatures.

References

[1] Polly Wainwright and Houssain Kettani. An analysis of botnet models. In *Proceedings of the 2019 3rd International Conference on Compute and Data Analysis*, pages 116–121, 2019.

[2] MohammadNoor Injadat, Abdallah Moubayed, and Abdallah Shami. Detecting botnet attacks in iot environments: An optimized machine learning approach. In *2020 32nd International Conference on Microelectronics (ICM)*, pages 1–4. IEEE, 2020.

[3] Robin Taylor, David Baron, and Daniel Schmidt. The world in 2025-predictions for the next ten years. In *2015 10th International Microsystems, Packaging, Assembly and Circuits Technology Conference (IMPACT)*, pages 192–195. IEEE, 2015.

[4] Smita Dange and Madhumita Chatterjee. Iot botnet: the largest threat to the iot network. In *Data Communication and Networks*, pages 137–157. Springer, 2020.

[5] Sunny Behal and Krishan Kumar. An experimental analysis for malware detection using extrusions. In *2011 2nd international Conference on Computer and Communication Technology (ICCCT-2011)*, pages 474–478. IEEE, 2011.

[6] Yan Naung Soe, Yaokai Feng, Paulus Insap Santosa, Rudy Hartanto, and Kouichi Sakurai. Implementing lightweight iot-ids on raspberry pi using correlation-based feature selection and its performance evaluation. In *International Conference on Advanced Information Networking and Applications*, pages 458–469. Springer, 2019.

[7] Wenjuan Li, Steven Tug, Weizhi Meng, and Yu Wang. Designing collaborative blockchained signature-based intrusion detection in iot environments. *Future Generation Computer Systems*, 96:481–489, 2019.

[8] Philokypros P Ioulianou and Vassilios G Vassilakis. Denial-of-service attacks and countermeasures in the rpl-based internet of things. In *Computer Security*, pages 374–390. Springer, 2019.

[9] Sherif Saad, Issa Traore, Ali Ghorbani, Bassam Sayed, David Zhao, Wei Lu, John Felix, and Payman Hakimian. Detecting p2p botnets through network behavior analysis and machine learning. In *2011 Ninth annual international conference on privacy, security and trust*, pages 174–180. IEEE, 2011.

[10] Nickolaos Koroniotis, Nour Moustafa, Elena Sitnikova, and Benjamin Turnbull. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Generation Computer Systems*, 100:779–796, 2019.

[11] S Garcia, A Parmisano, and MJ Erquiaga. Iot-23: A labeled dataset with malicious and benign iot network traffic. *Stratosphere Lab., Praha, Czech Republic, Tech. Rep*, 2020.

[12] Muhammad Nasir Mumtaz Bhutta, Amir A Khwaja, Adnan Nadeem, Hafiz Farooq Ahmad, Muhammad Khurram Khan, Moataz A Hanif, Houbing Song, Majed Alshamari, and Yue Cao. A survey on blockchain technology: evolution, architecture and security. *IEEE Access*, 9:61048–61073, 2021.

[13] Muhammad Hassan Nasir, Junaid Arshad, Muhammad Mubashir Khan, Mahawish Fatima, Khaled Salah, and Raja Jayaraman. Scalable blockchainsâ€"a systematic review. *Future Generation Computer Systems*, 126:136–162, 2022.

[14] Andrew West, Sampath Kannan, Insup Lee, and Oleg Sokolsky. An evaluation framework for reputation management systems. *Departmental Papers (CIS)*, 10 2012.

[15] Sarah Ali Siddiqui, Adnan Mahmood, Quan Z Sheng, Hajime Suzuki, and Wei Ni. A survey of trust management in the internet of vehicles. *Electronics*, 10(18):2223, 2021.

[16] Guillaume Arcas. French Chapter Status Report 2012 â€“ The Honeynet Project.

[17] Géza Szabó, Dániel Orincsay, Szabolcs Malomsoky, and István Szabó. On the validation of traffic classification algorithms. In *International conference on passive and active network measurement*, pages 72–81. Springer, 2008.

[18] LBNL/ICSI Enterprise Tracing Project - Project Overview.

[19] Yair Meidan, Michael Bohadana, Yael Mathov, Yisroel Mirsky, Asaf Shabtai, Dominik Breitenbacher, and Yuval Elovici. N-baiotâ€"network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, 17(3):12–22, 2018.

[20] Joel Margolis, Tae Tom Oh, Suyash Jadhav, Young Ho Kim, and Jeong Neyo Kim. An in-depth analysis of the mirai botnet. In *2017 International Conference on Software Security and Assurance (ICSSA)*, pages 6–12. IEEE, 2017.

[21] Artur Marzano, David Alexander, Osvaldo Fonseca, Elverton Fazzion, Cristine Hoepers, Klaus Steding-Jessen, Marcelo HPC Chaves, Ítalo Cunha, Dorgival Guedes, and Wagner Meira. The evolution of bashlite and mirai iot botnets. In *2018 IEEE Symposium on Computers and Communications (ISCC)*, pages 00813–00818. IEEE, 2018.

[22] Imtiaz Ullah and Qusay H Mahmoud. A scheme for generating a dataset for anomalous activity detection in iot networks. In *Canadian Conference on Artificial Intelligence*, pages 508–520. Springer, 2020.

[23] Eirini Anthi, Lowri Williams, Małgorzata Słowińska, George Theodorakopoulos, and Pete Burnap. A supervised intrusion detection system for smart home iot devices. *IEEE Internet of Things Journal*, 6(5):9042–9053, 2019.

[24] Sarumathi Murali and Abbas Jamalipour. A lightweight intrusion detection for sybil attack under mobile rpl in the internet of things. *IEEE Internet of Things Journal*, 7(1):379–388, 2019.

[25] Ayyaz-Ul-Haq Qureshi, Hadi Larijani, Nhamoinesu Mtetwa, Abbas Javed, Jawad Ahmad, et al. Rnn-abc: A new swarm optimization based technique for anomaly detection. *Computers*, 8(3):59, 2019.

[26] Qiao Tian, Jingmei Li, and Haibo Liu. A method for guaranteeing wireless communication based on a combination of deep and shallow learning. *IEEE Access*, 7:38688–38695, 2019.

[27] Mengmeng Ge, Xiping Fu, Naeem Syed, Zubair Baig, Gideon Teo, and Antonio Robles-Kelly. Deep learning-based intrusion detection for iot networks. In *2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC)*, pages 256–25609. IEEE, 2019.

[28] Yazan Otoum, Dandan Liu, and Amiya Nayak. Dl-ids: a deep learning–based intrusion detection framework for securing iot. *Transactions on Emerging Telecommunications Technologies*, page e3803, 2019.

[29] Joshua Bassey, Damilola Adesina, Xiangfang Li, Lijun Qian, Alexander Aved, and Timothy Kroecker. Intrusion detection for iot devices based on rf fingerprinting using deep learning. In *2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*, pages 98–104. IEEE, 2019.

[30] Shalaka Satam, Pratik Satam, and Salim Hariri. Multi-level blacktooth intrusion detection system. In *2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA)*, pages 1–8. IEEE, 2020.

[31] Dominik Breitenbacher, Ivan Homoliak, Yan Lin Aung, Nils Ole Tippenhauer, and Yuval Elovici. Hades-iot: A practical host-based anomaly detection system for iot devices. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, pages 479–484, 2019.

[32] Joshua Bassey, Damilola Adesina, Xiangfang Li, Lijun Qian, Alexander Aved, and Timothy Kroecker. Intrusion detection for iot devices based on rf fingerprinting using deep learning. In *2019 Fourth International Conference on Fog and mobile edge computing (FMEC)*, pages 98–104. IEEE, 2019.

[33] M Jagadeesh Babu and A Raji Reddy. Sh-ids: Specification heuristics based intrusion detection system for iot networks. *Wireless Personal Communications*, 112(3):2023–2045, 2020.

[34] Gunasekaran Raja, Aishwarya Ganapathisubramaniyan, Gokul Anand, et al. Intrusion detector for blockchain based iot networks. In *2018 Tenth International Conference on Advanced Computing (ICoAC)*, pages 328–332. IEEE, 2018.

[35] Sunwoo Ahn, Hayoon Yi, Younghan Lee, Whoi Ree Ha, Giyeol Kim, and Yunheung Paek. Hawkware: Network intrusion detection based on behavior analysis with anns on an iot device. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2020.

[36] Abebe Abeshu Diro and Naveen Chilamkurti. Distributed attack detection scheme using deep learning approach for internet of things. *Future Generation Computer Systems*, 82:761–768, 2018.

[37] Wenjuan Li, Weizhi Meng, and Man Ho Au. Enhancing collaborative intrusion detection via disagreement-based semi-supervised learning in iot environments. *Journal of Network and Computer Applications*, 161:102631, 2020.

[38] Geethapriya Thamilarasu, Adedayo Odesile, and Andrew Hoang. An intrusion detection system for internet of medical things. *IEEE Access*, 8:181560–181576, 2020.

[39] Chao Liang, Bharanidharan Shanmugam, Sami Azam, Mirjam Jonkman, Friso De Boer, and Ganthan Narayansamy. Intrusion detection system for internet of things based on a machine learning approach. In *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, pages 1–6. IEEE, 2019.

[40] Nandita Sengupta and Jaya Sil. *Intrusion Detection: A Data Mining Approach*. Springer Nature, 2020.

[41] Vikash Kumar, Ditipriya Sinha, Ayan Kumar Das, Subhash Chandra Pandey, and Radha Tamal Goswami. An integrated rule based intrusion detection system: analysis on unsw-nb15 data set and the real time online dataset. *Cluster Computing*, 23(2):1397–1418, 2020.

[42] Robin Gassais, Naser Ezzati-Jivan, Jose M Fernandez, Daniel Aloise, and Michel R Dagenais. Multi-level host-based intrusion detection system for internet of things. *Journal of Cloud Computing*, 9(1):1–16, 2020.

[43] Junaid Arshad, Muhammad Ajmal Azad, Mohammad Mahmoud Abdellatif, Muhammad Habib Ur Rehman, and Khaled Salah. Colide: A collaborative intrusion detection framework for internet of things. *IET Networks*, 8(1):3–14, 2019.

[44] Junaid Arshad, Muhammad Ajmal Azad, Muhammad Mahmoud Abdeltaif, and Khaled Salah. An intrusion detection framework for energy constrained iot devices. *Mechanical Systems and Signal Processing*, 136:106436, 2020.

M. Hassan Nasir is a Ph.D. scholar in the field of Cybersecurity at NED UET, Karachi, Pakistan. He received his Bachelor's degree in Computer System Engineering and Masters of Engineering degree in Communication Systems & Networks from Mehran UET Jamshoro, Pakistan in the year 2006 and 2012 respectively. His research interests includes Blockchain, Intrusion detection, Swarm Intelligence, and IoT Security.

Junaid Arshad is an Associate Professor at the School of Computing and Digital Technology, Birmingham City University, UK. He received his PhD in Computer Security from the University of Leeds, UK in 2011. His research interests include investigating security challenges for diverse computing paradigms such as distributed computing, cloud computing, IoT, and distributed ledger technologies.

Muhammad Mubashir Khan is a Professor in the Department of Computer Science and Information Technology at NED University of Engineering and Technology, Karachi Pakistan. He received his Ph.D. degree in Computing from University of Leeds, UK in 2011. He is serving as the Co-Principal Investigator of the National Center for Cybersecurity at NED University. He is also leading the Masters and PhD programmes in the Computer Science & IT Department at NED University. His research interests include Network and Information Security, Cybersecurity, Blockchain and Quantum Key Distribution in which he has published several research articles in prestigious research journals.