International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 6, pp: 282-295 www.ijaem.net ISSN: 2395-5252

A Supervised Machine Learning Approach for Smart Home IoT Devices

Rutuja Patil, Ruhin Patel, Priyanka Parit, Minakshi Patil, Rajashree Ganager

Corresponding Author; Rutuja Patil

Date of Submission: 02-09-2020 Date of Acceptance: 20-09-2020

ABSTRACT: The IoT device identification includes behavior of that device in our daily routine. Although these devices simplify and automate a day tasks, they also introduce security threats. To make Iot device secure current security measures are insufficient, which breaking Iot infrastructure and unable to proctet from attackers. Paper introduce three layers of Intrusion Detection System; that uses a supervised machine learning approach to detect a multiple popular computer network based cyber-attacks on IoT devices. It consists three functions: 1) Illustrates the regular behavior of particular IoT device connected in a network, 2) identifies malicious packets on the network 3) If attack deployed, then it classifies the type of attack. The system is checked within home test architecture, it consisting of 8 commonly used devices. The system is checked by deploying attacks. Attacks chosen from few main computer network based attacks. The categories such as: Denial of Service (DoS), Reconnaissance, Man-In-The-Middle (MITM)/Spoofing, and Replay. Proposed system can automatically differentiate between IoT devices, also checks whether network packets (network activity) is malicious or benign, and detect which type of attack was deployed on which device successfully.

Index Terms—Internet of Things (IoT), Smart Homes, Networking, Security, Intrusion Detection, Anomaly Detection, Supervised Machine Learning, Classification, Heterogeneity

I. INTRODUCTION

The popularity of Internet of Things (IoT) devices has grown exponentially in recent years. This is may be due to their ubiquitous connectivity that can motivate them to act on other technologies, their judgment and their intelligence.

It provides an uninterrupted user experience that forcibly extends people's day-to-day lives, and this is evidenced by how popular such devices are today.

However, the proliferation of smart devices is not only in the home environment but also the way behind the collapse of the world based on interconnected knowledge; our economy, institutions, state systems and important national infrastructure.

(CNI) [2]. More specifically, CNI concepts like smart homes, smart cities, intelligent transport, smart grids, and health care systems are heavily hooked in to smart technologies and IoT devices. Nevertheless, although these concepts support the tasks of lifestyle, their dependency on Information Communication Technology (ICT) and IoT devices accompany tremendous security risks. A survey by Synopsys in May 2017 revealed a scarcity of confidence within the security of medical devices with 67% manufacturers believing that an attack on a medical device is probably going to occur within 12 months, and only 17% of manufacturers taking steps to stop them.

Inadequate security measures for this controversial network and lack of dedicated inconsistency detection mechanisms make data vulnerable to many types of attacks such as data leakage, spoofing, service interruptions (DOS / DDOS), energy bleeding, unsafe entrances, etc. It can organize, Catastrophic consequences; Damage to hardware, disrupt system availability, cause system blackouts and even cause physical harm to an individual [5], [6]. Therefore, it is clear that the scale of the impact of attacks on IoT networks can vary drastically. As a result, it is clear that there is a large gap between the security requirements and security capabilities of current IoT devices. The two main reasons that make these devices insecure are differences in terms and restrictions on computer power, hardware, software and protocols. The traditional IT security approach to detecting attack signatures (e.g. honey pots) may be inadequate and / or non-scalable [10]. .In addition, IoT devices operate deep within the network, helping to prevent external attacks, so traditional perimeter immunity is inadequate, But often limited attacks detect and fail to prevent attacks from devices or applications within the proposed system; In particular, Routing Attack and DOS. In this case, the goal of the proposed system is to carry out large-scale attacks with multistage attacks that represent a complex combination of attack behaviors that are more challenging to detect. The ID presented in this paper is evaluated against 12 of the 6 most common types of attacks found in the IoT domain, and also against 4 scenarios of scripted multistage attacks with complex chains of events. Secondly, the existing system does not focus on device profiling. Finding malicious traffic without profiling the 'normal' behavior of devices connected to the network is a challenging task. Therefore, in this paper, the behavior of 8 different IoT devices is shown so that abnormal behavior can be detected, and then also cyber-attacks. Third, the current IDS failed to identify the type of attack. Without this information, significant human effort is required to respond to the alert and determine the inflexibility of the attack.

However, in this paper, machine learning methods have proven that it is possible to compensate for this attack by automatically differentiating between benign and malicious network traffic, not only to find out if an attack has been deployed, but also to automatically identify the type of attack. The architecture of the proposed IDS here is novel and takes into account most of the above limitations of the existing system.

Contributions to the work presented in these systems are:

-Three-layer architecture for standalone IDS, designed for IoT devices in smart home networks.

- Inspection that best represents the packet as a feature in the context of supervised learning, so that devices type, packet is malicious or not and attack type can be automatically identified.

II. LITERATURE REVIEW

A. Signature/Event/Rule based IDSs

Several studies revolving around IoT security have attempted to style IDS systems tailored specifically for the IoT ecosystem. Stephen and Arockiam [18] suggest a light-weight, hybrid approach to detect Hello Flood and Sybil attacks in IoT networks, which use the routing protocol like Routing over Low Power and Lossy Networks (RPL). Their system is predicated on an algorithm that uses detection metrics like number of packets received and transmitted to validate the Intrusion Ratio (IR) by the IDS agent. Raza et al. [19] implemented an IDS for the IoT called SVELTE. This technique consists of a 6LoWPAN Mapper (6Mapper), intrusion detection module, and a mini firewall. It analyses the mapped data to spot any intrusions within the network. Its performance in detecting various attacks seems promising. However, it's only been tested to detect spoofed or altered information, sinkhole, and selective-forwarding attacks. Shreenivas et al. [20], [21] extended SVELTE by adding another intrusion detection module that uses an Expected Transmission (ETX) metric to spot malicious activity on the network. They also proposed a geographic hint to detect malicious nodes that conduct attacks against ETX-based networks. Their results demonstrated that the general true positive rate increases once they combine the EXT and rank-based mechanisms.

Pongle and Chavan [22] propose a centralized and distributed architecture for a hybrid IDS, which they implemented supported simulated scenarios and networks. It focuses on detecting routing attacks like the wormhole attack. Jun and Chi [23] proposed an event-processing-based IDS for the IoT. This technique is specification-based and it uses Complex Event Processing techniques for attack detection. This technique collects data from IoT devices, extracts various events, and performs security event detection by attempting to match events with rules stored during a Rule Pattern Repository. Although it's more efficient than traditional IDS, it's CPU intensive. Summerville, Zach, and Chen [24] developed IDS for IoT supported a deep packet analysis approach which employs a bit-pattern technique. The network payloads are treated as a sequence of bytes called bit-pattern, and therefore the feature selection operates as an overlapping tuple of bytes called n-grams. A match between the bit-pattern and n-grams occurs [21] when the corresponding bits matches all positions. The system is evaluated by deploying four attacks and demonstrates a really low false-positive rate.

Midi et al. [15] Proposed a knowledge-driven, adaptive, and light-weight IDS. It collects knowledge about the features and components of the network to be monitored and takes advantage of it to dynamically configure an effective set of search techniques. This is often extended to the brand of new protocol standards at the equivalent time provided. [21]. the results proved that the system had high accuracy, mainly in DOS detection and routing attacks. Furthermore, Thanigaivelan et al. [25] Proposed a hybrid IDS for IoT. During this system, each node on the network monitors its neighbors. If abnormal test architecture is found, the monitoring node will block the packet in the abnormally behaved node in the information link layer and report it to its original node. Oh et al.

[26], Implemented distributed lightweight IDS for IoT, based on algorithms matching Paget payload and attack signature. They evaluate current IDS by deploying standard attacks and by using attack signatures from previous IDSs like SNORT. The results prove that the performance of this system is promising. Finally, Ioulianou et al. [27] Proposed hybrid lightweight signature-based IDS in an effort to minimize two changes to denial of service attacks; “Hello” flood and improve version number. Although their results look promising, their system is tested in a simulated environment using cooja.

B. Machine Learning IDSs

Amouri, Alaparthy, and Morgera [28] Brought IDS to the IoT network through supervised machine learning.IDS seeks to profile the mild behavior of nodes and identify any discrepancies in network traffic. The results prove that the system is in a position to successfully isolate benign and malignant nodes. However, the performance of IDS is evaluated in a simulated network and not real test architecture. Therefore, further evaluation is needed to check the performance of their systems against large-scale attacks and tools. Doshi et al. [29], IoT networks also use machine learning algorithms to detect Distributed Denial of Service (DDOS) attacks. They show that thereby demonstrating expertise in IoTspecific network behavior (e.g. limited number and regular interval between packets) leads to higher accuracy of detecting DDOS in IoT network traffic with the transmission of machine learning algorithms. Nevertheless, they experiments solely specialize in this sort of attack. Additionally, Shukla [30] has proposed IDS that uses a combination of machine learning algorithms such as K-means and decision tree, which will detect wormhole attacks on 6LoWPAN IoT networks. However, the results of this work are promising, the assessment of the proposed IDS was supported by duplication and therefore the effectiveness of IDS has not been tested against other attacks.

Amouri, Alaparthy, and Morgera [28] Brought IDS to the IoT network through supervised machine learning. IDS attempts to profile the nodes' mild behavior and identify any discrepancies in network traffic. The results proved that the system is in a position to successfully isolate benign and malignant nodes. However, the performance of IDS is evaluated in a simulated network and not realistic test architecture.

Therefore, further evaluation is required to check the performance. Doshi et al. [29], IoT networks also use machine learning algorithms to detect Distributed Daniel of Service (DDOS) attacks. It shows that demonstrating proficiency in IoTspecific network behavior (e.g. limited number and regular interval between packets) makes it more accurate to detect DDOS in IoT network traffic. However, those experiments are entirely specialized in this type of attack. Additionally, Shukla [30]

Meidan et al. [31] and McDermott et al. [32] both specialize in finding botnets in the IoT ecosystem and use in-depth learning techniques to realize this. It is hopeful to go to both cases as they will find the botnets successfully; however, these methods are not deployed to detect various attacks and are evaluated in a simulated environment. Restuccia et al. [33] Reviewed security threats within the IoT networks and discussed potential security solutions assigned to machine learning to detect and reduce attacks using polymorphic software and hardware. However, no description of the experimental setup, implementation and subsequent evaluation of the proposed mechanism is provided. Brun et al. [34] developed an in-depth learning-based approach using dense random neural networks to detect network attacks. Although attacks are often successfully identified from this perspective, the system was only evaluated on test architecture with 3devices and simple cyber attacks. In addition, packet features were associated with specific attacks, for example, a DOS attack, limiting the frequency of packets in your chosen period of time, the space of the attack.

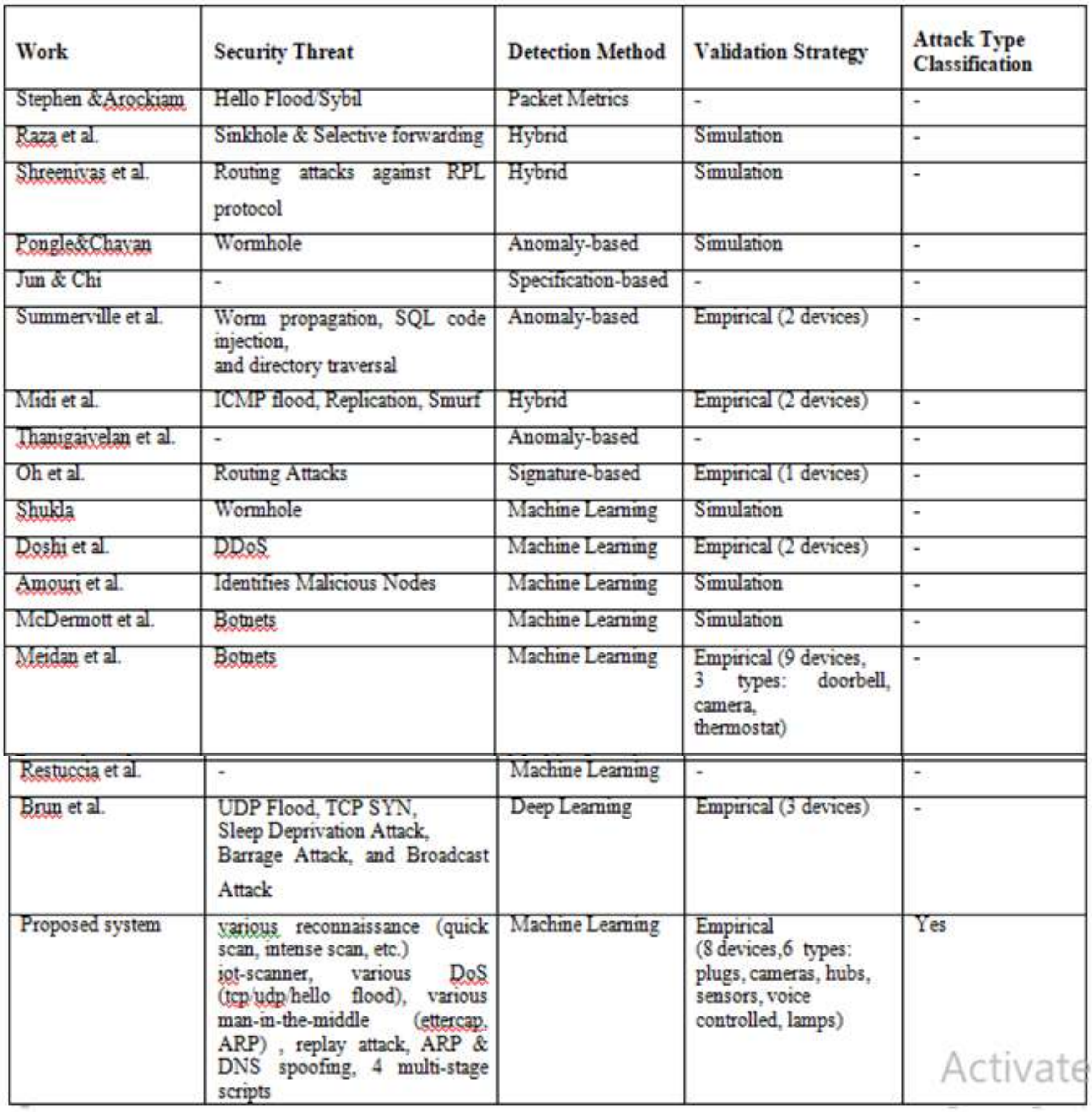
C. Attack Type Classification

There are currently a few approaches to the types of classification attacks. However, such methods have only been used in traditional networks and have been evaluated. Therefore, these methods are challenging to use in such an environment as they are not designed to rely on the exact requirements of IoT and computing capabilities. Bolozoni et al. [35] proposed a machine learning approach to classifying different types of cyber-attacks discovered by Alert Based Systems (ABS). To find out, byte sequences were extracted from the alert payload triggered by a specific attack. Compared sequences with the previous alert data. Although this system is effective in traditional systems, such approach relies on the alerts produced by the ABS, which aren't effective in IoT environments.

Although this system is effective in traditional systems, however, this approach relies on alerts created by ABS, which are not effective in IoT environments. (Eg. DOS) not found. Subba et al. [36] implemented a model that uses feed forward

and therefore back production algorithms to detect and classify cyber-attacks in desktop networks. However, they used the NSL-KDD dataset to impart knowledge to their system tried to classify DOS, User to Root and Remote to User Attack.
However, there is no evidence that this technique would be as effective if deployed in a heterogeneous IoT environment, which has many more protocols, devices, and network behaviors.

To summarize these methods, Table I shows the existing IDS for IoT and classifies them by search method, security risk, validity policy, and attack type. As a result, it is clear that previous IDS proposals dedicated to the IoT ecosystem are still in the early stages of development. Several approaches have used data from network simulations or have evaluated the system on a small array of IoT devices, which may compellingly decline from a realistic environment. Many methods have used data from network simulations or evaluated systems on a small array of IoT devices, which can be forcibly reduced from a realistic environment. This is an important feature of IDS, as it can be used for specific resistant specific attack types.



Work	Security Threat	Detection Method	Validation Strategy	Attack Type Classification
Stephen &Arockiam	Hello Flood/Sybil	Packet Metrics	-	-
Raza et al.	Sinkhole & Selective forwarding	Hybrid	Simulation	-
Shreenivas et al.	Routing attacks against RPL protocol	Hybrid	Simulation	-
Pongle&Chavan	Wormhole	Anomaly-based	Simulation	-
Jun & Chi	-	Specification-based	-	-
Summerville et al.	Worm propagation, SQL code injection, and directory traversal	Anomaly-based	Empirical (2 devices)	-
Midi et al.	ICMP flood, Replication, Smurf	Hybrid	Empirical (2 devices)	-
Thanigaivelan et al.	-	Anomaly-based	-	-
Oh et al.	Routing Attacks	Signature-based	Empirical (1 devices)	-
Shukla	Wormhole	Machine Learning	Simulation	-
Doshi et al.	DDoS	Machine Learning	Empirical (2 devices)	-
Amouri et al.	Identifies Malicious Nodes	Machine Learning	Simulation	-
McDermott et al.	Botnets	Machine Learning	Simulation	-
Meidan et al.	Botnets	Machine Learning	Empirical (9 devices, 3 types: doorbell, camera, thermostat)	-
Restuccia et al.	-	Machine Learning	-	-
Brun et al.	UDP Flood, TCP SYN, Sleep Deprivation Attack, Barrage Attack, and Broadcast Attack	Deep Learning	Empirical (3 devices)	-
Proposed system	various reconnaissance (quick scan, intense scan, etc.) iot-scanner, various DoS (tcp/udp/hello flood), various man-in-the-middle (ettercap, ARP) , replay attack, ARP & DNS spoofing, 4 multi-stage scripts	Machine Learning	Empirical (8 devices,6 types: plugs, cameras, hubs, sensors, voice controlled, lamps)	Yes

III. PROPOSED SYSTEM

The simplest of our knowledge, architecture
the IDS proposed here is a novel and addresses the majority the above limitations of prevailing systems.

• Three-layer architecture for light weight, independent
ID created according to IoT devices in a sensible home
Network

•Investigate which attributes are best represented packets as features in supervised context In this order of tools, malice and attacks can be identified automatically.

• Research resources that will help with further research
Automating IoT-based cyber-attacks, like mild and malicious networks.

METHODODLOGY

A. System Overview

The proposed system is in architecture three levels. The primary level of the tool will scan Network, support connected IoT devices. Their MAC addresses and their classification supported them network behavior. In the second layer, the packet whether such devices are classified as such mild or hateful. Finally, if there are malicious packets If found in the second layer, the third layer will be found classify this malicious packet with 4 main attacks type. As a result, in case of attack, output the system is as follows: 1) MAC address of the internal device fire, 2) whether the packet is malicious and3) sortthere has been an attack, which is one of four the main category that was trained on the model.

B. IoT Smart Home Test architecture

According to Cisco's VNI report in 2017 [37], North America, Western Europe, and 8, 5.4, on average in Central and Eastern Europe and 2.5 smart devices, respectively. Support the experiments provided during this paper without test bed. There are 8 commercially popular IoT devices; And so on could be a typical example of a certified smart homepage. Such devices included the Belkin net cam camera, TP-Link NC200 Camera, TP-Link Smart Plug, Samsung Smart Things Hub, Amazon Echo Dot, British Gas Hive connected to 2 sensors: one motion sensor and a Window / Door Sensor and Lifax Lamp. In addition, a Laptops were also connected to the network for two functions Functions: 1) Record continuous network traffic and automatically generate and save log files and 2) Deploy network based attacks. Figure 2 shows the architecture of the smart home test bed.

IoT device	Type	Protocol(s)
Amazon Echo Dot	Multimedia	Ethernet
BelkinNetCam	Multimedia	Wi-Fi
TP-Lik NC200	Multimedia	Wi-Fi
Hive Hub	Sensors	Ethernet &ZigBee
Samsung Smart Things Hub	Sensors	Ethernet & BLE
TP-Link Smart Plug	Sensors	Wi-Fi
Apple TV	Multimedia	Wi-Fi
Lifx Smart Lamp	Lamp	WiFi&ZigBee

TABLE II: IoT devices included in the smart home test architecture

IoT test bed, TCP Dump was scheduled to run at the entrance point (p1) as shown in the same figure. Sphere PCAP logs were then transferred and stored in syslog server.

C. Data Collection

1) Benign Network Data: to develop for other comparisons research (e.g. [38]), weeks of mild price data and a few weeks of malicious data were collected from Iot test bed. There were test beds described in Section III-B designed and executed were networks (local-to local or local-to-remote) caught. All inbound and outbound traffic in the catch smart devices were captured using the tcp dump tool, which was constantly running at the access point (shown in Figure 2 with red circular marker) the data collection process was automated using Cron Job and Bash scripts. Data frames were captured continuously and saved to syslog server during PCAP formatting. Files was generated at one minute intervals and accessed to attach remotely using Secure Shell (SSH) Sislog server.For benign data collection requirements, pcap files were automatically transferred and merged into it syslog server using Cron Jobs on which the series begins bash scripts.

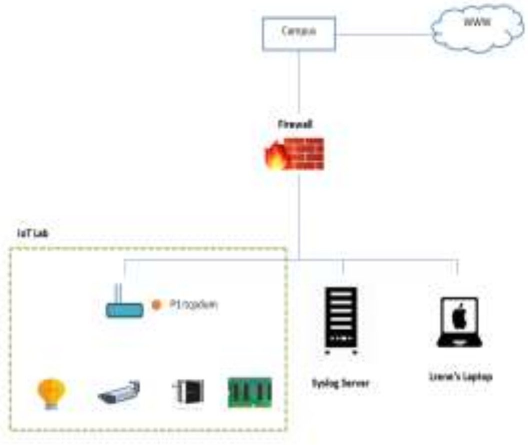


Fig. 2: IoT smart home test architecture network architecture.

2) Cyber-attacks in the IoT environment: multiple studies (E.g. [22], [40], [41], and [15]) IoT devices are being sensitive to a good range of attacks network attacks, physical attacks, firmware attacks and data leaks. Explanation of what such devices are unsafe in: Limitations computer power, lack of transport encryption, unsafe web interface, lack of authentication /authority systems and their abundance which implements a uniform security mechanism extremely challenging [42].As a result, many IoTs attack categories have emerged:

- **Denial of Service (DoS):** The purpose is to create Iot devices unavailable by its intended users interrupting them temporarily or indefinitely service [43].
- **Distributed Denial of Service (DDoS)/ Botnets:** The target of the attack was an outside compromise. Hence the number of insecure IoT devices ready to deploy significantly more severe DOS or other attacks [44].
- **Man-In-The-Middle:** Compromises with Chanel IoT devices and so on his data recipient wanted. Once the connection is compromised, the attacker is in a status of acting as a proxy and thus reading, insert, and modify transmitted data [45].
- **Spoofing:** Deals with fake identification compromise with the effectiveness of the IoT device by introducing outward to action as legal nodes [8].
- **Unsafe Firmware:** Compromises User Data take control of the IoT device and launch attacks against it other tools [42].
- **Data Leakage:** Many IoT devices suffer lack of transport encryption. It will end measuring data loss and leaked information; this can complete the compromise of the device or user accounts [42].

Must take 3 weeks of malicious activity planning and deployment of malicious variety attacks. The malicious machine attacker was a Lenovo ThinkPad that was configured to run Linux OS [46].Although many IoT devices are connected to the web via WiFi, they also support others Ethernet, IEEE 202.15., Communication Protocol Bluetooth, ZigBee, Z-Wave, Loravan and Cellular (GPRS / 2G / 3G / 4G). However, during this paper, WiFi and Ethernet communications are used. Table III demonstrated all attacks and therefore used the tools in this work.

To make sure the ID is tested properly, getting a broader data set was important, the Representatives of the attacks carried out. In particular, it was necessary to introduce something randomly deployed attacks to avoid compensating models. Because of this, bash scripts were implemented to automate and make malicious attacks random. Randomization was achieved by applying the launched timer, random attack (between 5 seconds and 20 minutes) for a random period of your time.

There was idle time between the launches of each attack also done randomly using the equivalence principle. Intensity of attacks, similar to iot-toolkit toggle attacks for some attack (e.g. Amount of malicious packets sent) was also random. Furthermore, there were four automated multilevel malicious situations enabled, and deployed over the network. This is often to increase the complexity of the attack, but also to represent when a real enemy will follow the next steps is attacking the device.

Attack Category	**Method**
Reconnaissance	Nap (Quick Scan, Intense Scan, etc.), iot-scanner
DoS/DDoS	TCP Flood/UDP Flood, Hello flood attacks
MITM	Ettercap, SSL Strip, Burp suit
Replay	mitmframework suite
Spoofing	DNS, ARP

TABLE III: Cyber-attacks that were deployed on the IoTtest architecture

1) Scenario 1: network scanning

The attacker does either a quick scan or two scan, try to do magic with one more in-depth and targeted. The script will present a second attack with a probability of 0.5. The argument for this is that the attacker will usually start them. Then attack with a quick scan to work the available hosts decides whether or not to proceed to look more complex for insecurity if needed.

2) Scenario 2: network scanning & Denial(s) of Service

This scenario involves faster scanning, but the attacker also performs one or more important function on the target network. Attacks up to six DoS are often carried out in a row. The duration between random attack attacks is also random wait timeshare used. The situation is targeting a random IP address recognized on a default network.

3) Scenario 3: network scanning & MITM

This scene is a fast-paced magical, but after the MITM attack by ARP spoofing, Packet injection either with passive monitoring or using this (selected at a probability of 0.5). Random attack times, wait times, as well as random numbers injected packets are selected automatically. MITM is usually set between the access point and one of the IP addresses present on the network (initially identified at the start of the script).

4) Scenario 4: complete attack with iot-toolkit

End-to-end automation of previously described framework IoT-toolkit attacks. It targets TPLink the tools for nicknames and the toggle / gate function information on the TPLink smart plug. Again, random duration, intensity, and wait time are automatically selected.

Meanwhile, another important concept was considered to be the program of the script to get all kinds of logs and the type of attack that happened. This required further labeling tasks for the supervising machine, needless to say about learning and attack work authentication. Ordinary logs were made to supply. An overview of the dates and attacks. In addition, logs of all output generated in between attacks occur (with output returned by the device) created for the purpose of debugging.

D. Feature Selection

The main requirements to consider when developing machine learning based IDS for IoT:

- Lightweight: No significant computer processing, energy required.
- Stand Alone: without access to other software or warning systems.
- Fast: Malicious activity should be detected in almost real time to reverse the back effect.
- To work over encrypted traffic: Many commercial IoT devices use transport encryption.

Given the above requirements, it was initially decided to investigate whether it is possible to detect malicious test architectures from a single packet. The reason behind this method is that, since single packet networks are the smallest piece of information, they are faster to process and then improve the speed of detecting malicious actions.

Raw PCAP files containing network packets were initially converted and represented in the form of Packet Description Glossary (PDML) [47].PDML complies with XML standards and contains packet segmentation / layer information. As a result, it allows access to all or any of the packet attributes used as a feature. Network packets consist of a series of layers (physical, data link, network, transport and) application, each layer is a small child of the previous layer and is formed from the lower layer of rock [] 48] (see figure).Each layer provides information and payloads of different fields in its own header. For the classification experiments discussed during this work, all the fields constructing all the above layers were removed, in order to research which of them is most relevant in finding benign and malignant test architecture IoT.

In addition to those features, other fields were included, such as: frame information [39] and packet type - which specified whether the information packet for the IoT device on the test architecture was inbound or outbound. In addition, the features captured by the network test architecture features instead of the network properties to ensure that the model is not hidden in the specific network configuration (properties IP address, time, packet ID) features that were deleted. Finally, since network traffic is encrypted, the payload information feature on the appliance layer is not considered. In total, 121 features were extracted from each packet and shown as feature vectors.

E. Data Labeling

Supervised machine learning requires labeled training data. For each dataset: 3 classification experiments were performed: (1) device type classification, (2) malicious packet detection classification, and (3) attack type classification.

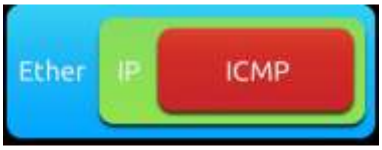


Fig. 3: An example of how layers are structured within a packet.

For (1) and (2), it has been found that the IP address of IoT devices on the test architecture will be changed frequently from specific attacks. There was no proper indicator to add category labels to the packet as IP addresses. Such devices were not accustomed to associating packets as MAC addresses. For (3), as the attacks occurred systematically, the packets were labeled as their attack type when completed. To ensure that the labeling of malicious packets is applied as accurately as possible, two parameters were considered: the launch time of the attack and therefore the MAC address of the attacker's machine. As a result, we recorded the exact time when the attack started and did not link it to the laptop's MAC address. Therefore, any packet with a time-stamp in the selected attack time-frame, including the attacker's MAC address, was labeled malicious. Finally, applications such as the attacker's machine services / mail and web browsers were disabled, so that any mild packets on the equivalent machine would not be mistaken for malicious. The category labels for each classification experiment are as follows:

(1): Amazon Mazon Echo Dot, Belkin Net, TP-Lick NC200, HV Hub, Samsung Smart Things Hub, TP-Link Smartplug, Lifts Smart Lamp, Firewall, Access Point.

(2): If a packet is collected during an attack targeting a device on the Iotest architecture, it is labeled malicious. Otherwise the packet was labeled as mild.

(3): DOS, MITM, Scanning, IoT-Toolkit.

Figures 4 - 6 show the distribution of packets in all classes for each experiment.

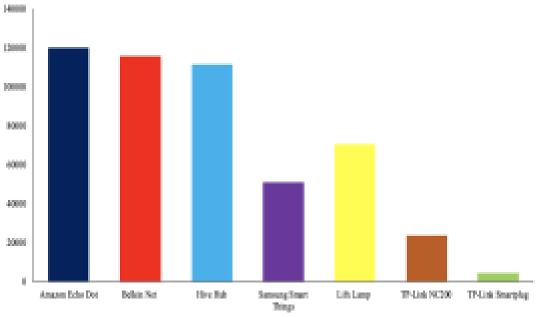


Fig. 4: Distribution of packets across IoT devices

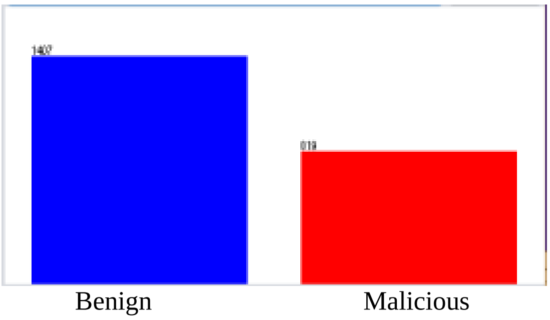


Fig. 5: Distribution of packets across attack detection

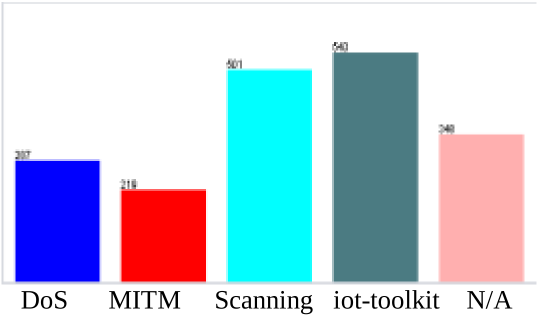


Fig. 6: Distribution of packets across attack types

F. Class Balancing and Sample Size Reduction

The uneven balance of sophisticated labels in each classification experiment (Figures 4 - 6) has the potential to have a negative impact on classification performance. In addition, applying a machine learning algorithm to a dataset with a large number of packets like the one created here requires high computing power and time duration.

Weka [49], a set of machine learning software, did not support classification experiments. Many unequal balances are given in the dataset and hence a large number of packets will be sorted, to obtain random packet samples of the scattering sub-sample and balancing filters available in Weka and then to balance the distribution of classes in those samples. For device type classification, sample size was obtained randomly from two 004,657 packets. The final sample size was 10,000 packets, 1000 packets in each device. To find out if the attack was malicious, random samples were generated from 220,785 packets of the dataset containing 80,000 packets (40,000 and 40,000 mild and malicious packets, respectively).Finally, to classify the attack sort, the final sample size was set to collect a total of fifty packets (10,000 packets per attack) of 220,785 packets.

IV. ALGORITHM SELECTION AND CLASSIFICATION EXPERIMENTS

Classification algorithms can learn how to profile IoT devices on a network, how to detect wireless attacks, and how to classify such attacks, supervised machine learning performance without the habit of training and evaluating relevant network activity data classification models.

- True Positive (TP) - When the packets become really malicious, they are presumed to be malicious.
- True Negative (TN) - Packets are supposed to be mild when they are really mild.
- False positives (FP) - Packets are considered extremely bad, when in fact they are mild.
- False Negative (FN) - In fact, packets are supposed to be mild when they are malicious.

There are many measures that can evaluate the performance of a classification. Our goal is to increase the target of all measures from 0 to 1. Therefore higher values are related to classification efficiency. The most common measures are accuracy, recall, f-measurement and accuracy. Precision (P) accurately calculates the amount of malicious packets, while Reckel (R) specifies how to accurately identify the amount of malicious packets. The two measures are usually used together in F-measurement (F) which remembers and recalls accuracy. Metrics for measuring general classification performance. In Equation 1 many solutions are calculated using the equation.

$$P = TP / TP + FP,$$
$$R = TP / TP + FN,$$
$$F = (2 .PR) / P + R \quad (1)$$

Others use accuracy to measure performance. Accuracy Measures the size of an accurately classified packet. However, the thing about using accuracy to survive for the classifier's performance is that if the classifier is always predicting a particular class, such a technique loses the purpose of classification, which will achieve higher accuracy. For the classification experiment, a random subset of 60% of each balanced dataset was selected for training, with the remaining 40% used for testing. The "free lunch" theorem suggests that the best learning algorithm is not everywhere [] 0]. In other words, the choice of a particular algorithm should support its functionality for that particular problem, and therefore the properties of knowledge that characterize that subject. In this case, the propagation of the distributed classification was evaluated as part of the weka.Clarifying other IDS using machine learning techniques to detect cyber-attacks in traditional and IoT networks (e.g. [] 1], [] 2]), class classifiers classify their multi-category classification, high-dimensional feature location, and therefore unseen data. The time required for the classification model. Classification involves creating models that consider conditional dependence on the dataset or assume conditional independence. (E.g. Bayesian network, navy bias) and discriminatory models that maximize retrieval of information or data in the relevant class without modeling any basic probabilities or structures. Yes, creating a live map. Information (e.g. J48 decision tree, support vector machine). In addition, they create classification models because their classification results can be better grasped so their classification models will be easily created.

V. RESULTS AND DISCUSSION

Summary IV reports the overall weighted-average performance of all 3 classifiers. Overall, the implementation of the 48 decision tree method [] 53] with Weka pruning resulted in excellent performance, resulting in experiment .6 64%, 100.0%, and.0 seconds and 0. Seconds we have performed additional experiments to ensure that the classification time of seconds does not exceed 48 classifiers, which will not cause any change in classification performance:

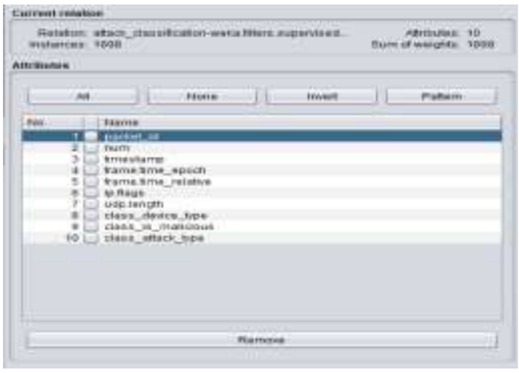
- Classification using Pr unpublished decision tree.
- Not all packets can be related to features because the feature space is too large. Two main feature selection methods were used to identify the most relevant features; Correlation properties evaluation filters and gain ratio evaluation filter. Evaluates the values of specificity by measuring the class and its interrelationships and its properties by obtaining information about the former class. The results showed that 10 out of 121 features are ranked as having the most correlation in the feature distance. Further classification was made using only high correlation features that

exist.

- Evaluates the latter value of an attribute

Figures 7 and 8 show the features in the top 10 that affect decision trees: ICMP field IP and TCP flags, packet and frame lengths, and TCP destination port. Specifically, if premitted, ICMP code options such as Fragment Protection and Packet Protection may indicate one

DOS attack. Furthermore, scanning methods and DOS (e.g. Sion flood) mostly include modified TCP flags in invalid or incorrect settings. In addition, specific TCP flags such as the TCP SYN probe and the TCP SQ probe may indicate a MITM attack. As a result, various combinations of flags are important indicators of malicious actions IP flags are indicative of an IP fragmentation attack and can take many forms, such as UDP (attack against IoT) or ICMP packet transmission. Due to the unavailability of this device, it may eventually be considered a form of DOS. The packet's destination Ports have another useful feature for detecting activities like port scanning, which usually consists of multiple on probes one or more ports.



Relation: attack_classification-weka.filters.supervised...
Instances: 1008
Attributes: 10
Sum of weights: 1008

No.		Name
1	□	packet_id
2	□	num
3	□	timestamp
4	□	frame.time_epoch
5	□	frame.time_relative
6	□	ip.flags
7	□	udp.length
8	□	class_device_type
9	□	class_is_malicious
10	□	class_attack_type

Fig 7. Top 10 features following correlation attribute filtering

The length of the packet is also indicative of malicious behavior, especially when the packet is forcibly larger than normal or smaller.

To gain a better understanding of classification performance in experiments, Confusion matrix in Table VI, which shows how predictive classes actually compare for individual packets, was analyzed. When profiling devils, the classifier showed a high percentage of accurate estimates, thus the devices less often classify incorrectly. Elevators, smart lamps, Samsung Smart Things Hub and Belkin Net, for example, showed a bit confused and generally properly classified. This can be explained by the fact that such tools are different.

- For class Classification means classification, classification means confusion matrix is less often a device in the wrong category. Best classification results in all three experiments.

Classifier	Device Identification		Detect Wireless Attacks		Attack Type Classification	
	Accuracy	Time taken for execution	Accuracy	Time taken for execution	Accuracy	Time taken for execution
J48	95.64%	0.79	100%	0.29	99.52%	0.4476
Bayesian Network	9.165%	25.45	63.108%	3.52	17.196%	5.67
K-nearest	87.49%	6.20	99.31%	2.54	98.56%	2.89

Table IV: Performance of all 3 classifiers

In this case, the features may be in some packets in one device but are missing in the packet in others. For example, the test architecture of the TP-Link NC200 is particularly different from the test architecture of the TP-Link Smart Plug because the functions they perform are different. In this case, features in the TPLink Connectionless protocol in NC200 packet, User Datagram Protocol (UDP), while TP-Link Smart Plug uses the Packet Transmission Control Protocol (TCP). However, in some cases, confusion often occurs when the belkin net and hive hub were incorrectly classified. This confusion can be explained by the concept that the network behavior of such devices may be similar when compiling data, such as when firmware updates were assigned. Determining whether network packets are malicious or mild and identifying the types of wireless attacks show even less confusion. The attacks can be explained by that fact carried out during data collection come from shelf attacks, i.e. resources that include freely available attacks, such as HPing, NMAP, IoT-Toolkit, etc., and are unpleasant. In this case, the characteristics of the malignant and mild packets are different and thus, caused some confusion in the classification. For example, a malicious packet may contain a different flag that indicates that an attack has occurred as previously reported.

To conclude, the main objectives of these results are:

- Trees Decision Trees (especially J48) seems to be the best algorithm in this work as it works.
- IP and TCP flags are the most important features.
- To detect malicious packets of lic, the confusion matrix indicates The classifier also shows even less confusion
- The high accuracy of the classification can be explained by the fact that the deployed attacks were not sophisticated and were not deployed using their own tools. Resulting traffic and networks test architecture changes during this period class.
- When unseen authentication dataset Performance, accuracy of device type classification and detection of malicious packets is particularly omitted.

A. Use Case

The main use case of the proposed IDS in this paper is the case identifies real-time malicious behavior in smart home IoT devices and to identify what type of attack occurred. However, IoT itself this is a large concept consisting of a significant number of odd elements devices. Large networks, along with Other IoT devices, traditionally divided into sub-networks, each have a set of devices. In this case, considering increasing the number of IDs proposed in this paper, IDS can be deployed on each sub-network to detect malicious activity in the environment with more equipment. Numerous instances of IDS eventually make it possible to share network activity data across each sub-network. Data from a sub-network Data from a sub-network with different devices When connected to new sub-networks, they can be used to train IDS to detect malicious activity in such devices.

VI. FUTURE WORK

Given the positive results of the initial study, the next step is to implement this technique in real time, so that it can be deployed in very real, very large, heterogeneous IoT and even industrial IoT environments. This allows for more evaluation of the system for more complex and sophisticated attacks. In addition, to leave out the detailed requirements of feature engineering and date labeling, the use of DOP learning techniques can automatically determine which packet features have an impact in identifying malicious activity in the IoT environment.

VII. CONCLUSION

In this project, three layers represent intelligent IDs. To meet the above limitations of existing System, the ID presented here includes three main functions: 1) to classify the general behavior and types Of every IoT device connected to the network, 2) wireless attacks deployed on the connected IoT device and))attack Classify Type of banana. To evaluate the efficiency of implementing supervisory machine learning approaches to automate each task, network activity data were collected from actual test architectures consisting of commercially available and popular IoT devices. The two main functions are System performance for each experiment 95.64%, 100.0%, and 99.52% and a classification time of 0.8 seconds, 0.3 seconds, and 0.4 seconds for each experiment respectively. This Indicates whether the proposed architecture can successfully differentiate between IoT devices on the network whether there is activity on the network is malicious or benign. In addition to this project, resources are available that will support research to automate IoT-based cyber-attack detection. Such sources Includes raw PCAP files and flow information for malicious and malicious network activities.

REFERENCES

[1]. Rafiullah Khan, SarmadUllah Khan, Rifaqat Zaheer, and Shahid Khan(2012). Future internet: the internet of things architecture, possible applications and key challenges.In Frontiers of Information Technology (FIT),

2012 10th International Conference on, pages 257–260. IEEE.

[2]. Tobby Simon. Chapter seven(2017): Critical infrastructure and the internet of things. Cyber Security in a Volatile World, page 9.

[3]. EiriniAnthi, Lowri Williams, and Pete Burnap (2018). Pulse: adaptive intrusion detection for the internet of things.

[4]. Cyber security executive: Medical devices a ’bullseye’forcyberattacks.https://www.digitalhealth.net/2017/12/medicaldevicefunctionalityvscybersecurity/.(Accessedon02/05/2018).

[5]. EiriniAnthi, Amir Javed, Omer Rana, and George Theodorakopoulos (2017). Secure data sharing and analysis in cloud-based energy management systems. In Cloud Infrastructures, Services, and IoT Systems for Smart Cities, pages 228–242. Springer.

[6]. Cyber hackers can now harm human life through smart meters smart grid awareness. https://smartgridawareness.org/2014/12/30/ hackers-can-now-harm-human-life/. (Accessed on 02/05/2018).

[7]. Securing the internet of things: A proposed framework cisco. https://www.cisco.com/c/en/us/about/security-center/secure-iot-proposed-framework.html. (Accessed on 07/13/2018).

[8]. Liang Xiao, Xiaoyue Wan, Xiaozhen Lu, Yanyong Zhang, and Di Wu(2018). Iot security techniques based on machine learning. arXiv preprint arXiv:1801.06275,.

[9]. EiriniAnthi, Shazaib Ahmad, Omer Rana, George Theodorakopoulos, and Pete Burnap(2018). Eclipseiot: A secure and adaptive hub for the internet of things. Computers & Security, 78:477–490.

[10]. Tianlong Yu, VyasSekar, SrinivasanSeshan, YuvrajAgarwal, and ChenrenXu(2015). Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the internet-of-things. In Proceedings of the 14th ACM Workshop on Hot Topics in Networks, page 5. ACM.

[11]. MichaelV¨ogler, Johannes Schleicher, Christian Inzinger, Stefan Nastic, Sanjin Sehic, and Schahram Dustdar (2015). Leonore – large - scale provisioning of resource - constrained ioteployments. In Service - Oriented System Engineering (SOSE), 2015 IEEE Symposium on, pages 78–87. IEEE.

[12]. The limit does not exist: Why defending the perimeter is not feasible in the iot - 2018-03-04-page1-rfidjournal. http://www.rfidjournal.com/articles/view?16805.(Accessedon 03/29/2018).

[13]. Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot)(2013): A vision, architectural elements, and future directions. Future generation computer systems, 29(7):1645– 1660.

[14]. Martin Roesch et al(1999). Snort: Lightweight intrusion detection for networks. In Lisa, volume 99, pages 229–238.

[15]. Daniele Midi, Antonino Rullo, Anand Mudgerikar, and Elisa Bertino (2017). Kalisa system for knowledge-driven adaptable intrusion detection for the internet of things. In Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on, pages 656–666. IEEE.

[16]. Gartner says 6.4 billion connected ”things” will be in use in 2016, up 30 percent from 2015. https://www.gartner.com/newsroom/id/3165317. (Accessed on 07/13/2018).

[17]. Bruno Bogaz Zarpelao, Rodrigo Sanches Miani, Cl´audio Toshio Kawakani, and Sean Carlisto de Alvarenga et al (2017). A survey of intrusion detection in internet of things. Journal of Network and Computer Applications, 84:25–37.

[18]. R Stephen and L Arockiam(2017). Intrusion detection system to detect sinkhole attack on rplprotocolin internet of things. International Journal of Electrical Electronics and Computer Science, 4(4):16–20.

[19]. ShahidRaza, Linus Wallgren, and Thiemo Voigt. Svelte (2013): Real-time intrusion detection in the internet of things. Ad hoc networks, 11(8):2661– 2674.

[20]. Dharmini Shreenivas, Shahid Raza, and Thiemo Voigt(2017). Intrusion detection in the rpl-connected 6lowpan networks. In Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security, pages 31–38. ACM.

[21]. Leonel Santos, Carlos Rabadao, and Ramiro Gonc¸alves (2018). Intrusion detection systems in internet of things: A literature review. In 2018 13th Iberian Conference on Information Systems and Technologies (CISTI). IEEE.

[22]. Pavan Pongle and Gurunath Chavan (2015). Real time intrusion and wormhole attack detection in internet of things. International Journal of Computer Applications, 121(9).

[23]. Chen Jun and Chen Chi (2014). Design of complex event-processing ids in internet of things. In Measuring Technology and Mechatronics Automation (ICMTMA), 2014 Sixth International Conference on, pages 226–229. IEEE.

[24]. Douglas H Summerville, Kenneth M Zach, and Yu Chen (2015). Ultra light weight deep packet anomaly detection for internet of things devices. In Computing and Communications Conference (IPCCC), 2015 IEEE 34th International Performance, pages 1–8. IEEE.

[25]. Nanda Kumar Thanigaivelan, Ethiopia Nigussie, Rajeev Kumar Kanth, Seppo Virtanen, and JouniIsoaho et al(2016). Distributed internal anomaly detection system for internet-of-things. In Consumer Communications & Networking Conference (CCNC), 2016 13th IEEE Annual, pages 319–320. IEEE.

[26]. Doohwan Oh, Deokho Kim, and Won Woo Ro(2014). A malicious pattern detection engine for embedded security systems in the internet of things. Sensors, 14(12):24188–24211.

[27]. PhilokyprosIoulianou, Vasileios Vasilakis, Ioannis Moscholios, and Michael Logothetis (2018). A signature-based intrusion detection system for the internet of things. Information and communication Technology Form.

[28]. Amar Amouri, Vishwa T Alaparthy, and Salvatore D Morgera (2018). Cross layer-based intrusion detection based on network test architecturefor iot. In Wireless and Microwave Technology Conference (WAMICON), 2018 IEEE 19th, pages 1–4. IEEE.

[29]. Rohan Doshi, Noah Apthorpe, and Nick Feamster (2018). Machine learning ddos detection for consumer internet of things devices. arXiv preprint arXiv:1804.04159.

[30]. PrachiShukla. Ml-ids (2017): A machine learning approach to detect wormhole attacks in internet of things. In Intelligent Systems Conference (IntelliSys), 2017, pages 234–240. IEEE.

[31]. Yair Meidan, Michael Bohadana, Yael Mathov, YisroelMirsky, AsafShabtai, Dominik Breiten bacher, and Yuval Elovici (2018). N-baiotnetworkbased detection of iot botnet attacks using deep autoencoders. IEEE Pervasive Computing, 17(3):12–22.

[32]. Christopher D McDermott, FarzanMajdani, and Andrei V Petrovski(2018). Botnet detection in the internet of things using deep learning approaches. In 2018 International Joint Conference on Neural Networks (IJCNN), pages 1–8. IEEE.

[33]. Francesco Restuccia, Salvatore DOro, and Tommaso Melodia (2018). Securing the internet of things in the age of machine learning and software-defined networking. IEEE Internet of Things Journal, 5(6):4829–4842.

[34]. Olivier Brun, Yonghua Yin, and Erol Gelenbe (2018). Deep learning with dense random neural network for detecting attacks against iot-connected home environments. Procedia computer science, 134:458–463.

[35]. DamianoBolzoni, SandroEtalle, and Pieter H Hartel(2009). Panacea: Automating attack classification for anomaly-based network intrusion detection systems. In International Workshop on Recent Advances in Intrusion Detection, pages 1–20. Springer.

[36]. BasantSubba, Santosh Biswas, and Sushanta Karmakar (2016). A neural network based system for intrusion detection and attack classification. In 2016 Twenty Second National Conference on Communication (NCC), pages 1–6. IEEE.

[37]. Cisco visual networking index(2017): Forecast and trends, 20172022 white paper - cisco. https://www.cisco.com/c/en/us/solutions/collateral/ service-provider/visual-networking-index-vni/white-paper-c11-741490. html. (Accessed on 03/26/2019). [38] Arunan Sivanathan, Daniel Sherratt, Hassan Habibi Gharakheili, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman. Characterizing and classifying iot traffic in smart cities and campuses. In Proc. IEEE INFOCOM Workshop SmartCity, Smart Cities Urban Comput., pages 1–6.

[38]. Wiresharkgodeep.https://www.wireshark.org/.(Accessedon07/18/2018).

[39]. TariqahmadSherasiya and HardikUpadhyay. Intrusion detection system for internet of things. International Journal of Advance Research and Innovative Ideas in Education, 2(3).

[40]. Prabhakaran Kasinathan, Claudio Pastrone, Maurizio A Spirito, and Mark Vinkovits (2013). Denial-of-service detection in 6lowpan based internet of things. In Wireless and Mobile Computing, Networking and Communications (Wi

Mob), 2013 IEEE 9th International Conference on, pages 600–607. IEEE.

[41]. Owasp internet of things project - owasp. https://www.owasp.org.(Accessedon05/31/2018).

[42]. Ioannis Andrea, Chrysostomos Chrysostomou, and George Hadjichristofi (2015). Internet of things: Security vulnerabilities and challenges. In Computers and Communication (ISCC), 2015 IEEE Symposium on, pages 180–187. IEEE.

[43]. Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, and Jeffrey Voas(2017). Ddos in the iot: Mirai and other botnets. Computer, 50(7):80–84.

[44]. Luigi Atzori, Antonio Iera, and Giacomo Morabito (2010). The internet of things: A survey. Computer networks, 54(15):2787–2805.

[45]. Kali linux - penetration testing distribution - documentation. https:// docs.kali.org/. (Accessed on 02/15/2018).

[46]. Pdml - the wireshark wiki. https:// wiki.wireshark.org/ PDML. (Accessed on 03/27/2019).

[47]. Scapy p.04 looking at packets — thepacketgeek. https://thepacketgeek. com/ scapy-p-04-looking-at-packets/. (Accessed on 05/14/2019).

[48]. Weka 3 - data mining with open source machine learning software in java. https:// www.cs.waikato.ac.nz/ml/weka/. (Accessed on 06/03/2018).

[49]. David H Wolpert(1996). The lack of a priori distinctions between learning algorithms. Neural computation, 8(7):1341–1390.

[50]. Chih-Fong Tsai, Yu-Feng Hsu, Chia-Ying Lin, and Wei-Yang Lin. Intrusion detection by machine learning: A review. expert systems with applications, 36(10):11994–12000, 2009.

[51]. Mahesh kumar Sabhnani and G¨ursel Serpen (2003). Application of machine learning algorithms to kdd intrusion detection dataset within misuse detection context. In MLMTA, pages 209–215.

AUTHORS PROFILE

Rutuja Patil
BE in Computer Science & Engineering from Sanjay Ghodawat Institute of Engineering, Kolhapur.

Priyanka Parit
BE in Computer Science & Engineering from Sanjay Ghodawat Institute of Engineering, Kolhapur.

Ruhin Patel
BE in Computer Science & Engineering from Sanjay Ghodawat Institute of Engineering, Kolhapur.

Minakashi Patil
BE in Computer Science & Engineering from Sanjay Ghodawat Institute of Engineering, Kolhapur.

Rajashree Ganager
BE in Computer Science & Engineering from Sanjay Ghodawat Institute of Engineering, Kolhapur.