Machine Learning based Attacks Detection and Countermeasures in IoT

Rachid Zagrouba[1] and Reem Alhajri[2]

[1,2]College of Computer Science & Information Technology, Imam Abdulrahman Bin Faisal University, Dammam 31441, Saudi Arabia

Abstract: While the IoT offers important benefits and opportunities for users, the technology raises various security issues and threats. These threats may include spreading IoT botnets through IoT devices which are the common and most malicious security threat in the world of internet. Protecting the IoT devices against these threats and attacks requires efficient detection. While we need to take into consideration IoT devices memory capacity limitation and low power processors. In this paper, we will focus in proposing low power consumption Machine Learning (ML) techniques for detecting IoT botnet attacks using Random forest as ML-based detection method and describing IoT common attacks with its countermeasures. The experimental result of our proposed solution shows higher accuracy. From the results, we conclude that IoT botnet detection is possible; achieving a higher accuracy rate as an experimental result indicates an accuracy rate of over 99.99% where the true positive rate is 1.000 and the false-negative rate is 0.000.

Keywords: Internet of Things, Botnets, Anomaly Detection, Distributed Denial of Service, Machine learning.

1. Introduction

The increasing use of Internet of Things (IoT) is growing, and by 2020 Gartner assume that IoT devices uses will reach 20.4 billion on all over the world. This lead to use advanced technology which include using smart homes, smart cars and smart cities. This advancement led Google and Apple, as well as several major automotive industry (e.g., Ford, Honda, Toyota, General Motors, etc.), to move to develop and invent an automatic self-driven cars, also IoT devices is been used in medical as well, such as using it as wearable devices for checking vital signs, or used as a reminder for patients to the time and dosage of their medications. With the wide spread all over the world of IoT devices they have to face a cybersecurity crisis, which has become clear through the escalation of security breaches over the past five years.

Hackers launched a large-scale DDoS attack in October of 2016 using a botnet and leveraging Linux based IoT devices that had been infected with the Mirai malware. These attacks interrupt communicating to the most popular websites including Amazon, GitHub, Slack, Visa, and HBO.

Cyberattacks might cause a full control of most places using IoT devices such as controlling airline control systems, prison systems, and a lot of vulnerable IoT devices which have severe consequences [1].

We can define a botnet as a set of computers that have been connected to the internet and which have been bound together and are controlled remotely through the use of some malicious software known as bots by an intruder. Internet of things, on the other hand, can be defined as the new paradigm that connects the internet with physical objects from different aspects such as home automation and environmental monitoring. Internet of things in the recent past has tremendously developed in many parts of the world, and it is gaining momentum very rapidly from both the industry and academia. The main concept of the internet of things is to interconnect a local network of smart objects from different parts of the world using a standard internet protocol in order to access information and data with various functionalities like tracking, monitoring identifying and locating different phenomena [2]. However, this proposal intends to discuss the various ways in which internet of things has affected our day to day lives and how the introduction of botnets has affected the world of internet of things negatively and the proposed solutions to curb these malicious phenomena.

Digitalization and competition have made the organization come up with the new technology and information technology infrastructure. However, at the same time individuals, prominent organizations and authorities are afraid of the cyber-attacks. This is because when these attacks occur, they are normally associated with bringing in significant negative impacts on the organization or an individual's business. The key drivers of the increased cases of hacking are because of the availability of hacking tools which are readily available and can easily be accessed by anyone. This has many companies come up with their security operation center SOC or hiring the services from outside. The main work of the security operation center is to continuously monitor security devices of the company, like the firewalls, email servers anti-virus servers and many others [3]. IoT-based botnets, such as the Mirai DDoS malware can exploit these vulnerabilities, but no single existing system has demonstrated the capacity to detect these intrusions. Among the proposed botnet detection methods, a distinction exists between network-based detection approaches and host-based detection approaches [4] [5] [6][7] [8]. Table 3 shows dataset attributes [9].

2. Related works

2.1 IoT Security Vulnerabilities and Botnet/DDoS Attacks

There are many security vulnerabilities in the structure of IoT layers [10]. as each security vulnerabilities differ from each layer. As there are several proposed models of IoT security architectures [11][12][13], but none of them guarantee perfect security against all different types of threats. In particular, as IoT network layer is exposed to many kinds of security threats, such as Man-in-The-Middle (MiTM) attacks and DoS attacks [14]. the most famous attacks are botnets and DDoS attacks because of its potential impact, which focus on compromising the availability of information systems [15].

2.2 Machine Learning Detection of IoT Botnets And DDoS

Botnet detection methods can be either specific operational steps or utilization of detection approaches. for botnet detection operational steps such as using Software Defined Networking (SDN) collaborative schemes [16]. it focus on the early stages of attack propagation and execution within the C&C server. The proposed solution to this problem is to focus on IoT botnet operations and make more steps for these operations by depending on using network-based botnet detection methods. In [17], researchers adopted a model for classifying methods of detecting network-based botnet through using detection techniques, sources of detection and algorithms used for detection. The used detection technique can be either anomaly-based detection or fingerprint-based detection methods. For the sources of detection, might be depending on normal sources from real network or botnet sources that can be from honeypots or simulation solutions. Lastly, for the detection algorithms it can be using instance-based learning, supervised learning, semi-supervised learning, unsupervised learning, use of heuristic rules, and signal processing. A growing body of literature focuses on comparing the performance of various ML-based detection methods in IoT environments. The comparison was between the detection accuracies of three supervised ML classifiers designed to mitigate DDoS attacks in IoT networks: Naïve Bayes, Decision Tree, and Linear Discriminate Analysis as shown in Table 1.

2.3 Anomaly Detection of IoT Botnets Using Auto-Encoders

In [19] authors presented auto-encoder neural networks for anomaly detection in Wireless Sensor Networks (WSNs) which is embedded in IoT environments. The detection algorithm contains two components; one placed within sensors and the other one is added in the IoT cloud. The evaluation shows that the unsupervised learning features of the auto-encoder neural network allowed accepting unexpected changes in IoT networks.

Based on Table 2 the approaches proposed by [20] and [21] present the most promising results of the studies reviewed. These methods use deep auto-encoders to automatically extract features and training data, which significantly increases the accuracy of the detection. In addition, auto encoder s can be device-based and/or network-based implying that the features can be extracted from device or traffic data. Since most IoT devices have low memory capacity, it is possible to deploy the model's deep learning capabilities in a cloud environment and therefore only implement lightweight models on the devices and network for detecting anomalies. As a result, it is easy to maintain continuous monitoring of the devices and network. In addition, it is relatively fast to detect devices and networks that have been compromised due to low burden on device memory.

3. Methodologies

This section focuses on the methods used to answer the question raised by the research proposal, providing the main reasons why choosing this research topic. This thesis is based on studying and reading a set of literature review for recent research described in the previous section, then in this section it is providing the testing, measuring and acquiring results using ML techniques. The purpose of this research is to explore the opportunities that ML techniques will present in the detection of IoT botnets.

There are two types of approaches for research: Inductive and deductive. A deductive approach usually starts with a hypothesis from an existing theory where data are collected to check the reality of that hypothesis or falsity of it, while an inductive approach will usually use research questions to narrow the scope of the study. The methods used to carry out this research include deductive approach whereby known figures and tests were carried out and explored in order to come up with clear information and final deductions to be made. different materials were used to give these deductions on machine learning especially in the field of IoT. Our thesis will focus on starting with general concepts and theories, then applying these concepts to the real applications [8].

3.1. Deductive Research Approach

Deduction research starts using a general idea and moves on until it ends up with a specific one, this type of research is using statistical analysis to make a relation between known data and learned data through research. Collecting and analyzing data need to understand the relationships for used variables using either descriptive or inferential statistics.

3.2. Flow Diagram

As shown in Fig. 1 it describes the workflow and the methodology used in this research.

- Literature Review. Based on the selection of the research topic we first start reviewing the literature review of recent studies and find out the gaps in it.
- Get security logs dataset. When the logs are obtained from the internet, it is important to categorize the logs and to select the utility to be used for them.
- Simulation. In this thesis, software WEKA was used to simulate machine learning (Random Forest algorithm, Decision tree) in testing phase.

3.3. Data Collection

Security logs are required for testing the algorithm. There is one publicly available data source from the cloudstor. The cloudstor security logs datasets have been widely used for testing IoT products. They contain extensive examples of attacks and background traffic [8].

Using tshark tool raw network packets (Pcap files) of the BoT-IoT dataset was created which has a combination of normal and abnormal traffic in the Cyber Range Lab of the Australina Center for Cyber Security (ACCS). Using Ostinato tool and Node-red (for non-IoT and IoT respectively) simulated network traffic was generated. The dataset's source files are provided in different formats, such as the original pcap files, the generated argus files and finally in CSV format [8].

The files were separated, based on attack category and subcategory as following:

- DDoS: HTTP, TCP, UDP
- DoS: HTTP, TCP, UDP
- SCAN: OS, SERVICE

Dataset is divided into the following:

- 10-best features
    - 10-best Training-Testing split
- All features

Our dataset focus on using 10-best features testing dataset.

4. Simulation

Waikato Environment for Knowledge Analysis (WEKA) is used in our thesis evaluation machine learning tool which is written in Java Language by University of Waikato, New Zealand. This software is used for data mining with many machine learning algorithms, it contains many tools such as for virtualization, classification, regression, clustering, association rules, and data pre-processing [22].

4.1. Data Analysis

We can analyze network connectivity by checking the network flow information provided using testing tools that will show us a summary of the connectivity between two hosts. A network flow can provide the following details: source and destination IP addresses, source and destination port numbers, and protocol. etc., it can provide a good information about activities handled through network during the flow from the network traffic. In our research, we get a dataset from cloud store that extract features from network traces. Overall, 17 features are obtained. For each flow, a feature vector is constituted by the features listed in Table 2 [8].

4.2. Results and Discussions

In this section, we focus in achieving high level of accuracy for detection of botnet traffic. In the first phase, we start analyzing of network traffic in the dataset to classify and detect botnet, and then the second phase is based on a comparison done between Random Forest (RF) and Decision Tree (DT) results with previous related work. Our analysis results of the dataset are shown below.

- Test mode: user supplied test set.
- Classifier model (full training set).
- Used algorithm: Random Forest, Decision tree.
- Bagging with 100 iterations and base learner.
- Weka classifiers:
    - Misc.: Input Mapped Classifier -I -trim -W
    - Trees:
        - Random Forest -- -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1.
        - J48 -- -C 0.25 -M 2.

Time taken to build a model in RF was 281.19 seconds, where it takes about 16.79 seconds for DT. Compared with the related work shown in Table 8, it shows that the accuracy and precision of using RF are higher than used with DT or any other algorithms.

Below it indicates that of the 733705 instances of botnet activities propagating DoS, DDoS, Scan and Theft attacks, the Random Forest algorithm accurately identifies 99.9238% compared to the 99.8982% detected by the Decision Tree (DT) algorithm. The RF classification is more accurate as it incorrectly classified fewer instances (559) compared to the (747) errors of DT.

Furthermore, in Table 5 below indicates that RF accuracy of detecting and classifying classes is accurate relative to the normal. For instance, it has flawless identification of TCP, UDP categories.

Where in Table 6 below, it indicates that RF gradually identifies classes over various metrics during repeated but random analysis of the data set. Notably, outliers from the data set are minimal to insignificant.

Table 7 indicates a degree of confidence in DT's accuracy of detecting and classifying classes relative to the normal, with the latter being close to ideal while DT has flawless identification of TCP and UDP categories.

Table 8 below indicates that RF gradually identifies classes over various metrics during repeated but random analysis of the data set. However, there are some uncaptured deviations; for instance, some categories class not categorized in some cases.

In this paper, we have analyzed the data resulting from detecting and classifying distributed denial of service (DDoS) attacks on the Internet of Things (IoT) using the Random Forest (RF) and Decision Tree (DT) algorithms. The data indicate that the RT algorithm is more accurate than the DT algorithm in capturing various indicators of a DDoS attack. For instance, Table 3 indicates that RF correctly classified instances of possible malicious intent with an accuracy rate of 99.9238%, which is greater than DT's 99.8982%. A failure by any system to detect a DDoS attack would result in the target system's services being unavailable to legitimate users. While there is a marginal difference between the HTTP details between the two algorithms, this gap would be of significance considering the multitude of vectors used by DDoS attacks. Comparing the results in Table 5 and Table 8, RF has seen having better recall HTTP with 99.4% detection and classification rate compared to DT's 98.2%. This conclusion further reinforced by analyzing the results of Table 5 and Table 8 for RF and DT, respectively; it yields more precise results with a lower false-negative rate.

Analysis of the data reveals that RF is superior to DT in detecting botnet activities that propagate DDoS attacks. Notably, RF combines several DTs that randomized to avoid bias and missing new data. Therefore, this makes RF more accurate as the diversity of the data set classified captured accurately. The RF algorithm handles cumulative data better with an increase in performance and the accuracy of the results. This feature makes RF algorithms suitable for identifying DDoS attack activity data, which is rapidly dynamic, a feature that can capture in a forest of DTs, thereby triggering remediation steps in time.

5. Common Attacks in IoT and their Countermeasures

5.1. Common attacks in IoT

There are various models for classifying attacks in IoT. In [23], the security attacks in IoT can be classified into five categories depending on the component targeted or involved in the attack. The first category involves physical attacks, which target the hardware components of IoT networks. Physical attacks are difficult to execute in IoT environments due to the massive resources required to implement the attacks. The common attacks include layout reconstruction attacks, de-packaging attacks, micro probing, and particle beam techniques. The second category of attacks are side-channel attacks, which use side channel information retrieved from the encryption device. The adversaries use the retrieved information to recover the key that the devices utilize. Some of the common examples of this type of attack include environmental attacks, timing attacks, fault analysis attacks, power analysis attacks, and electromagnetic attacks. The third category of attacks comprises of cryptanalysis attacks, which target the cipher-text as a strategy to compromise the encryption mechanism and obtain the plaintext. Examples of this type of attack include cipher-text only attacks, chosen-plaintext attacks, known-plaintext attack, and Man-In-The-Middle (MITM) attacks. Fourthly, software attacks occur in the IoT exploiting software security vulnerabilities in any of the core or embedded systems. Mostly, these attacks exploit vulnerabilities in the communication interfaces and entail using buffer overflow techniques and injection of malicious codes. The fifth category of attacks in IoT entails network attacks, which typically exploit vulnerabilities in wireless communications. These attacks are classified into two categories: active attacks and passive attacks. Passive attacks include eavesdropping, traffic analysis, and camouflage attacks. Active attacks include DoS, node capture and malfunction, as well as message routing attacks, and node subversion.

In [23] researcher describe another approach to classification of IoT device attacks, with four distinct types of attacks depending on the layer of the IoT affected: physical, network, software, and encryption attacks. Physical attacks occur when the adversary is physically close to the IoT device. Network attacks entail manipulation of the IoT networks to cause damage to the underlying systems. Software attacks involve exploitation of vulnerabilities in the applications. Lastly, encryption attacks involve compromising the encryption system. However, encryption attacks target multiple layers of the IoT, with the adversary seeking to find and exploit any vulnerability.

5.2. Countermeasures

Classification of attached are presented in Table 9. This section presents a description of the countermeasures for the IoT attacks based on the classification presented in Table 10.

5.2.1 Physical security countermeasures

Physical layer security measures in the context of IoT fall into five categories:

1) Secure booting: the first countermeasure is to authenticate the software running on IoT devices to verify the integrity of cryptographic hash algorithms. However, the low processing power on most IoT devices means that implementation of cryptographic hash functions is limited to lightweight solutions. In [24], proposed IoT Sense, a method for device fingerprinting using ML techniques to extract device behavior features from the network traffic. Evaluation of the method demonstrates its capability to identify IoT devices and establishing strong authentication.
2) Device authentication: all new IoT devices introduced into the network must be authenticated before transmission data. One approach is to use ML technique to detect unauthorized IoT devices in a network. In [25] presented a supervised ML algorithm, using Random Forest to extract network traffic features to identify all IoT devices from a whitelist. The method demonstrated the capability to detect unauthorized IoT devices. In [26], IoT Sentinel, a similar approach was presented which identifies all the devices connected in an IoT network, with minimal performance overhead.
3) Data integrity: each device should have an error detection mechanism to prevent tampering of privacy-sensitive data. The most appropriate techniques for the IoT context include low power consumption methods, such as Checksum, Cyclic Redundancy Checks (CRC), and Parity Bits. In [27] they presented a novel CRC method for error correction in IoT applications, which uses iterative decoding method. The proposed approach uses existing redundancy in CRC nodes, without implying additional processing overheads.
4) Data confidentiality: each IoT device should support data encryption before transmission. The most appropriate mechanisms for encryption include RSA and Blowfish, due to the requirements for low power consumption in cryptographic encryptions. Previously, a method for securing medical data in IoT-based systems was proposed, in which the researchers used a hybrid encryption method, combining AES and RSA. The method showed the ability to conceal the patient’s data [28].
5) Anonymity: another countermeasure to secure the physical IoT layer is to hide sensitive information, such as locations and identity details of network nodes. Unfortunately, some of the robust anonymity approaches such as Zero Knowledge cannot work in an IoT environment because they require significant processing power. An alternative solution is to integrate K-Anonymity approach for low-power IoT devices [29].

5.2.2 IoT network security countermeasures

1) Data privacy countermeasures: using authentication and encryption methods can help to mitigate threats to unauthorized access to sensor nodes. ProfiiIOT, an ML bases IoT device identification method was proposed, which uses network traffic analysis to identify devices in an IoT network [30].
2) Routing security countermeasures: secure routing is the best approach to address threats to routing security. However, most routing protocols have security vulnerabilities, which expose IoT devices to security threats. Network encryption and authentication ensures secure data routing. In [31], the authors overview standard and non-standard protocols for routing in IoT applications, including RPL protocol, the standard protocol, Collection Tree Protocol (CTP) for WSN, Lightweight on-demand ad-hoc distance-vector routing protocol-next generation LOADng. Using standard routing protocols enhances security in IoT.
3) Data integrity: the best countermeasure against data integrity threats is to use cryptographic hash functions.

5.2.3 Application layer security

1) Data security: the most critical countermeasure against threats to data security is to use data authentication and encryption, which prevents unauthorized access.
2) Access control lists (ACLs): one best practice is to establish security policies and permissions for IoT systems, using ACLs to filter user’s access requests and determine whether to allow or deny access. In [32] they used the blockchain concept and ML to ensure IoT security using dynamic access control policy.

3) Firewalls and antivirus software: comprehensive security in the IoT context should involve the use of firewalls and anti-virus programs. In [33] they proposed an intelligent method for improving IP state scanning in IoT networks, using IP randomization, reactive port scanning, and OS fingerprint. The researchers used k algorithm to search for IoT devices with security vulnerabilities. Similar approaches include ZMap [34], an open-source scanner with the capacity to scan IP addresses on a single port within an hour. Similarly, Shodan [35] is a robust search engine for the IoT, which enables users to find publicly accessible devices. Currently, there is no viable antivirus software product with sufficient memory and processor power to work for IoT devices.

6. Conclusion

In this experiment, the RF classifier used with the data set and we get higher accuracy. From the results, we conclude that IoT botnet detection is possible; achieving a higher accuracy rate as an experimental result indicates an accuracy rate of over 99.99% where the true positive rate is 1.000 and the false-negative rate is 0.000. RF accurately classifies the greatest number of instances compared to DT due its nature of aggregating multiple randomized decisions in arriving at a result. In addition, RF provides a higher precision with a lower false-negative rate for data, which emphatically makes it better than the DT algorithm used. The choice of using RF over DT in a system to deter DDoS attacks would be more viable as there would be fewer false positives triggering unnecessary remediation activities and false negatives manifesting as unidentified attack activities. Future work should focus on reducing the time needed for processing data and training the used algorithms to classify various data sets presented for analysis. This recommendation guided by the overhead introduced by classifying a bigger forest of randomized data, i.e. a DDoS attack on a bigger scale. that IoT botnet detection is possible; achieving a higher accuracy rate as an experimental result indicates an accuracy rate of over 99.99% where the true positive rate is 1.000 and the false-negative rate is 0.000. RF accurately classifies the greatest number of instances compared to DT due its nature of aggregating multiple randomized decisions in arriving at a result. In addition, RF provides a higher precision with a lower false-negative rate for data, which emphatically makes it better than the DT algorithm used. The choice of using RF over DT in a system to deter DDoS attacks would be more viable as there would be fewer false positives triggering unnecessary remediation activities and false negatives manifesting as unidentified attack activities. Future work should focus on reducing the time needed for processing data and training the used algorithms to classify various data sets presented for analysis. This recommendation guided by the overhead introduced by classifying a bigger forest of randomized data, i.e. a DDoS attack on a bigger scale.

References

[1] G. Falco, P. Fedorov, and C. Caldera, “NeuroMesh : IoT Security Enabled by a Blockchain Powered Botnet Vaccine,” ACM Proceedings: International Conference on Omni-Layer Intelligent Systems, Crete Greece, pp.1-6, 2019.

[2] B. B. Zarpelão et al, "A survey of intrusion detection in Internet of Things," Journal of Network and Computer Applications, Vol. 84, pp. 25-37, 2017.

[3] Y. Meidan et al ,"N-BaIoT-Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders ",IEEE Pervasive Computing ,Vol. 17 , No. 3, pp. 12-22, 2018.

[4] Rose, K., Eldridge, S. and Chapin, L., "The internet of things: An overview,". *The Internet Society (ISOC)*, Vol. 1, No. 1, pp.1-50, 2015.

[5] Y. Fu et al, "An Automata Based Intrusion Detection Method for Internet of Things," Mobile Information Systems, Vol. 2017, pp. 1-13, 2017.

[6] J. Granjal, J. M. Silva and N. Lourenço" ,Intrusion detection and prevention in CoAP wireless sensor networks using anomaly detection ",Sensors (Switzerland) ,Vol. 18, pp. 24-45, 2018.

[7] M. Nobakht, V. Sivaraman and R. Boreli" ,A host-based intrusion detection and mitigation framework for smart home IoT using OpenFlow", 11th International Conference on Availability, Reliability and Security (ARES), Salzburg, Austria, 2016.

[8] E. Giancarlo, V. Villano, “Classification of logs using Machine Learning Technique,” Master's thesis, NTNU, 2018.

[9] G. Biau" ,Analysis of a random forests model ",Journal of Machine Learning Research ,Vol. 13, pp. 1063-1095, 2012.

[10] P. Sethi and S. R. Sarangi, "Internet of Things: Architectures, Protocols, and Applications," Journal of Electrical and Computer Engineering, Vol. 2017, pp. 1-25, 2017.

[11] N. Almrezeq, L. Almadhoor, T. Alrasheed, A. Abd El-Aziz and S. Nashwan, "Design a secure IoT Architecture using Smart Wireless Networks" International Journal of Communication Networks and Information Security (IJCNIS), Vol. 12 No. 3, pp. 401-408, 2020.

[12] X. Liu et al, "A security framework for the internet of things in the future internet architecture," Future Internet, Vol. 9, pp. 27, 2017.

[13] J. Lin et al, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," IEEE Internet of Things Journal, Vol. 4, No. 5, pp. 1125-1142, 2017.

[14] E. Leloglu, “A Review of Security Concerns in Internet of Things,” Journal of Computer and Communications, Vol. 1, pp. 121–136, 2017.

[15] A. Lohachab and B. Karambir, "Critical Analysis of DDoS—An Emerging Security Threat over IoT Networks," Journal of Communications and Information Networks, Vol. 3, No. 3, pp. 57-78, 2018.

[16] S. Hameed et al" ,SDN based collaborative scheme for mitigation of DDoS attacks ",Future Internet ,Vol. 10 ,pp. 23, 2018.

[17] S. García, A. Zunino and M. Campo, "Survey on network-based botnet detection methods," Security and Communication Networks, Vol. 7, No. 5, pp. 878-903, 2014.

[18] D. Kajaree and R. . Behera, “A study of DDoS attacks detection using supervised machine learning and a comparative cross-validation,” *Int. J. Innov. Res. Comput. Commun. Eng.*, Vol. 5, No. 2, pp. 1302–1309, 2017.

[19] T. Luo and S. G. Nagarajan, "Distributed anomaly detection using autoencoder neural networks in WSN for IoT," IEEE International Conference on Communications (ICC 2018), Kansas City, MO, USA, 2018.

[20] C. Zhou & R. Paffenroth, “Anomaly Detection with Robust Deep Autoencoders,” In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge

Discovery and Data Mining, Halifax NS Canada, pp. 665–674, 2017.

[21] Q. Niyaz, W. Sun and A. Y. Javaid, "A Deep Learning Based DDoS Detection System in Software-Defined Networking (SDN)," ICST Transactions on Security and Safety, Vol. 4, No. 12, pp. 153515-12, 2017.

[22] E. Frank et al, "weka," in Anonymous Boston, MA: Springer US, pp. 1305-1314, 2005.

[23] I. Andrea, C. Chrysostomou and G. Hadjichristofi, "Internet of things: Security vulnerabilities and challenges,", IEEE Symposium on Computers and Communication (ISCC), Larnaca, Cyprus, 2015.

[24] B. Bezawada, M. Bachani, J. Peterson, H. Shirazi, I. Ray, and I. Ray, “IoTSense : Behavioral Fingerprinting of IoT Devices,” arXiv preprint arXiv:1804.03852, 2018.

[25] Y. Meidan, M. Bohadana, N. O. Tippenhauer, and J. D. Guarnizo, “Detection of Unauthorized IoT Devices Using Machine Learning Techniques,” arXiv preprint arXiv:1709.04647, 2017.

[26] M. Miettinen et al, "IoT SENTINEL: Automated device-type identification for security enforcement in IoT,", IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 2017.

[27] E. Tsimbalo, S. Member, X. Fafoutis, and R. J. Piechocki, “CRC error correction in IoT applications,” IEEE Transactions on Industrial Informatics, Vol. 13, pp. 361-369, 2017.

[28] M. Elhoseny et al, "Secure Medical Data Transmission Model for IoT-Based Healthcare Systems," IEEE Access, Vol. 6, pp. 20596-20608, 2018.

[29] F. Liu and T. Li, "A Clustering K-Anonymity Privacy-Preserving Method for Wearable IoT Devices," Security and Communication Networks, Vol. 2018, pp. 1-8, 2018.

[30] Meidan, Yair, Michael Bohadana, Asaf Shabtai, Juan David Guarnizo, Martín Ochoa, Nils Ole Tippenhauer, and Yuval Elovici. "ProfilIoT: a machine learning approach for IoT device identification based on network traffic analysis." In Proceedings of the symposium on applied computing, Marrakech Morocco, pp. 506-509, 2017.

[31] T. H. Bharat, “Network routing protocols in IoT,” International Journal of Advances in Electronics and Computer Science, Vol1, No. 4, pp. 29–33, 2017.

[32] A. Outchakoucht and J. P. Leroy, “Dynamic Access Control Policy based on Blockchain and Machine Learning for the Internet of Things,” Int. J. Adv. Comput. Sci. Appl, Vol. 8, No. 7, pp. 417-424, 2017.

[34] H. Kim, T. Kim and D. Jang, "An Intelligent Improvement of Internet-Wide Scan Engine for Fast Discovery of Vulnerable IoT Devices," Symmetry, Vol. 10, No. 5, pp. 151-167, 2018.

[35] H. Al-alami, A. Hadi, and H. Al-bahadili, “Vulnerability Scanning of IoT Devices in Jordan Using Shodan,” In 2017 2nd International Conference on the Applications of Information Technology in Developing Renewable Energy Processes & Systems (IT-DREPS), Amman, Jordan, pp. 1-6, 2017.

[36] A. PINTO and R. COSTA, "Hash-chain-based authentication for IoT," ADCAIJ : Advances in Distributed Computing and Artificial Intelligence Journal, Vol. 5, No. 4, pp. 43-57, 2016.

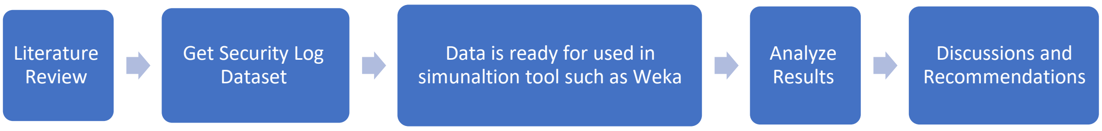
Literature Review
Get Security Log Dataset
Data is ready for used in simunaltion tool such as Weka
Analyze Results
Discussions and Recommendations

Figure 1: Flow Diagram

Table 1: Comparison of ML detection techniques for IoT-based DDoS attacks [18]

ML Classifiers	**Training & Testing Time**	**Precision**	**Advantage**	**Disadvantage**
Naïve Bayes	0.450	0.767031 (with selection feature)	• Shorter training time • Selection features can enhance precision significantly	Low precision
Decision Tree Classifier	22.689; 0.09545	0.898595 (with selection feature)	• Better defense against DDoS attacks • Can fit big training datasets better than small datasets	Longer training and testing times
Linear Discriminate Analysis	0.450363	0.771159	• Gives superior precision levels and shorter training time	N/A

Table 2: Proposed IoT Botnet detection methods

Ref #	**Advantages of the proposed method(s)**	**Weakness of the proposed method(s)**	**Proposed Method**
[20]	• Does not require clean training data • Testing set is not affected by outliers hence has consistent results • High recall	• Low precision • Time consuming in high rank matrix dimensions • Potential for high false negative rate for data with numerous sparse components	Robust Deep Auto-encoder
[21]	• Network-based which enhances the detection rate • Automatically extracts features from packet headers • High detection accuracy (99.82%) • Low false positive rate	• Training is time consuming • Data is pre-processing resulting in the loss of some features	Deep learning stacked auto-encoder

Table 3: Dataset Attributes [9]

	Attribute	**Attribute Type**	**Description**
1	**pkSeqID**	Numeric	Row identifier
2	**Proto**	Nominal	Textual representation of transaction protocols presents in network flow
3	**Saddr**	Nominal	Source IP address
4	**Daddr**	Nominal	Destination IP address
5	**Seq**	Numeric	Argus sequence number
6	**Mean**	Numeric	Average duration of aggregated records
7	**Stddev**	Numeric	Standard deviation of aggregated records
8	**Min**	Numeric	Minimum duration of aggregated records
9	**Max**	Numeric	maximum duration of aggregated records
10	**Srate**	Numeric	Source-to-destination packets per second

11	**Drate**	Numeric	Destination-to-source packets per second
12	**N_IN_Conn_P_SrcIP**	Numeric	Number of inbound connections per source IP
13	**N_IN_Conn_P_DstIP**	Numeric	Number of inbound connections per destination IP
14	**Attack**	Numeric	Class label: 0 for Normal traffic, 1 for Attack Traffic.
15	**Category**	Numeric	Traffic category
16	**Subcategory**	Nominal	Traffic subcategory
17	**state_number**	Nominal	Numerical representation of feature state

Table 4: Stratified cross-validation Summary for Random Forest & Decision Tree

Random Forest	Number of Instances	Accuracy
Correctly Classified Instances	733146	99.9238 %
Incorrectly Classified Instances	559	0.0762 %
Kappa statistic	1	
Decision Tree	**Number of Instances**	**Accuracy**
Correctly Classified Instances	732958	99.8982 %
Incorrectly Classified Instances	747	0.1018%
Kappa statistic	0.9999	
Total Number of Instances	733705	

Table 5: Accuracy detection RF by classification

Class	TP Rate	FP Rate	Precision	Recall	F-Measure
UDP	1.000	0.000	1.000	1.000	1.000
TCP	1.000	0.000	1.000	1.000	1.000
Service Scan	1.000	0.000	1.000	1.000	1.000
HTTP	0.994	0.000	1.000	0.994	0.997
OS Fingerprint	1.000	0.000	1.000	1.000	1.000
Normal	1.000	0.000	1.000	1.000	1.000
Keylogging	1.000	0.000	1.000	1.000	1.000
Weighted Avg.	1.000	0.000	1.000	1.000	1.000

Table 6: Confusion Matrix for RF Classification for attack

Classified as	A	B	C	D	E	f	G
a = UDP	396578	2	0	0	0	0	0
b = TCP	2	318335	0	0	0	0	0
c = Service Scan	0	0	14541	0	1	0	0
d = HTTP	0	3	0	501	0	0	0
e =OS Fingerprint	0	0	0	0	3621	0	0
f = Normal	0	0	0	0	0	107	0
g = Keylogging	0	0	0	0	0	0	41

Table 7: Accuracy detection DT by classification

Class	TP Rate	FP Rate	Precision	Recall	F-Measure
UDP	1.000	0.000	1.000	1.000	1.000
TCP	1.000	0.000	1.000	1.000	1.000
Service Scan	1.000	0.000	1.000	1.000	1.000
HTTP	0.982	0.000	0.992	0.982	0.987
OS Fingerprint	0.999	0.000	0.999	0.999	0.999
Normal	1.000	0.000	1.000	1.000	1.000
Keylogging	0.929	0.000	1.000	0.929	0.963
Weighted Avg.	1.000	0.000	1.000	1.000	1.000

Table 8: Confusion Matrix for DT Classification for attack

Classified as	A	B	c	D	E	f	G
a = UDP	396577	1	0	1	1	0	0
b = TCP	5	318329	0	3	0	0	0
c = Service Scan	0	0	14540	0	2	0	0
d = HTTP	0	7	0	495	2	0	0
e = OS Fingerprint	2	0	3	0	3616	0	0
f = Normal	0	0	0	0	0	107	0
g = Keylogging	0	0	1	0	0	0	13

Table 9. Classification of IoT Attacks

Physical Attacks	Network Attacks	Software Attacks	Encryption Attacks
Node Tempering	Routing Attacks	Virus & worms attacks	Side Channel Attacks
RF Interference	RFID Spoofing Attacks	Spyware & Adware Attacks	Cryptanalysis Attacks
Node Jamming	RFID Cloning Attacks	Trojan Horses	MITM attacks
Malicious Node Injection	MITM	DoS	
Physical Damage	DoS	Malicious scripts	
Social Engineering	Sybil Attacks		
Code Injection			

Table 10. Comparison of Security countermeasures for different IoT layers

IOT (Layer) Attack	Countermeasures for specific IoT layers	ML-based security solutions	Sources
Physical Layla	Secure booting for IoT Devices	IOTSense	[24]
	Device Authentication based on Low Power Methods	IOT Sentinel CRC. Checksum.	[25] [26] [27]
	Data Confidentiality	RSA Blowfish	[28]
	Data Anonymity	K-Anonymity	[29]
Network Layer	Secure Communication in IoT Devices	ProiIOT	[30]
	IoT Network Routing	Hashing routing, Standard IoT Network Routing Protocols, RPL, LOADng	[31]
	Secure user data in IoT Devices	Hash-Chain-based Authentication	[36]
Application Layer	Data Security		
	ACLs	ACLs Dynamic Access Control Policy (block chain)	[32]
	Firewalls	IP state scanning, XMap, Shodan	[33] [34] [35]