Denial of Service Detection for IoT Networks Using Machine Learning

Husain Abdulla[1][a], Hamed S. Al-Raweshidy[2][b] and Wasan Awad[3][c]

[1]*Department of Computer Science, Brunel University, Uxbridge, U.K.*
[2]*Department of Electronic and Computer Engineering, Uxbridge, U.K.*
[3]*Information Technology College, Ahlia University, Al-Hoora, Bahrain*

Keywords: Intrusion Detection System, IoT, Machine Learning, Security, Anomaly Detection.

Abstract: The Internet of Things (IoT) is considered one of the trending technologies today. IoT affects a variety of industries, including logistics tracking, healthcare, automotive and smart cities. A rising number of cyber-attacks and breaches are rapidly targeting networks equipped with IoT devices. Due to the resource-constrained nature of the IoT devices, one of the Internet security issues impacting IoT devices is the Denial-of-Service (DoS). This encourages the development of new techniques for automatically detecting DoS in IoT networks. In this paper, we test the performance of the following Machine Learning (ML) algorithms in detecting IoT DoS attacks using packet analysis at regular time intervals: Neural Networks (NN), Gaussian Naive Bayes (NB), Decision Trees (DT), and Support Vector Machine (SVM). We were able to achieve 98% accuracy in intrusion detection for IoT devices. We have created a novel way of detecting the attacks using only six attributes, which significantly reduces the time to train the ML Models by 58% on average. This research is based on data collected from actual IoT attacks on IoT networks. This paper shows that using the DT or NN; we can detect attacks on IoT devices. Furthermore, it shows that NB and SVM are poor in detecting IoT attacks. In addition, it proves that middle boxes embedded with ML Models can be utilized to detect attacks in places such as houses, manufactures, and plants.

1 INTRODUCTION

The Internet of Things promises an optimistic technological future where the physical world is integrated with computer-based systems, resulting in economic benefits and improvements in efficiency. The IoT is a network of objects, including devices, home appliances, and vehicles, which may be embedded with electronics, sensors, and software to enable it to connect and exchange data. Although the IoT makes considerable progress, they are vulnerable to cyberattacks due to their resource-constrained nature. Therefore, they rely on external systems, such as intrusion detection systems, to be protected. DoS attacks are common effective attacks to disturb IoT networks.

It is estimated that the number of Internet of Things (IoT) devices will be over 75 billion by 2025 (Fadul, Reising, Loveless & Ofoli, 2021) and they will be collecting data of more than 180 zettabytes. Yet, there are plenty of these IoT devices that are insecure and prone to attacks (Davis, Mason, & Anwar, 2020). A recent security review of IoT devices categorize these attacks into four categories namely: physical, network, software, and encryption attacks (Andrea, Chrysostomou & Hadjichristofi, 2015).

Intrusion Detection System (IDS) is used to prevent the DoS attacks. Apart from the most used method that is based on the port number, which is suited for the rule-based attack detection, machine learning methods are widely used in recent years for DoS and anomaly detection. A recent research for anomaly detection has shown the possibility of machine learning to identify malicious Internet traffic (Bagaa, Taleb, Bernabe & Skarmeta, 2020).

[a] https://orcid.org/0000-0002-3022-1985
[b] https://orcid.org/0000-0002-3702-8192
[c] https://orcid.org/0000-0001-7152-3480

Abdulla, H., Al-Raweshidy, H. and Awad, W.
Denial of Service Detection for IoT Networks Using Machine Learning.
DOI: 10.5220/0011885700003393
In *Proceedings of the 15th International Conference on Agents and Artificial Intelligence (ICAART 2023) - Volume 3*, pages 996-1003
ISBN: 978-989-758-623-1; ISSN: 2184-433X

However, limited research has been done to develop machine learning models with characteristics specifically targeted at IoT device networks and attack traffic. The IoT devices' traffic is different from other devices connected to the Internet (such as laptops and mobile phones) (Mishra, Varadharajan, Tupakula & Pilli, 2019). IoT devices, for example, are often connected to a small number of service endpoints rather than a large number of servers. Furthermore, IoT devices often generate the same network traffic patterns; for example, while logging, regular network pings of small packets at predetermined intervals are used.

Given the lack of public datasets of real network IoT attack traffic, there are limited studies on the performance of machine learning algorithms in detecting DoS attacks in IoT network. In this study we utilized a relatively recent open-source data set to perform our research. We found that neural networks performed better than other machine learning algorithms. We expect that the developed neural network model will continue to be effective with traffic of real-world deployments. Our traffic analysis model is designed to run on network middle boxes such as firewalls, network routers and switches to detect anomalous traffic.

The main contributions presented in this paper are:

- Developing four machine learning models using Gaussian Neural Networks, Naive Bayes, Decision Trees, and Support Vector Machine to detect DoS attacks on IoT devices using six attributes only.
- Reducing the time required to training and detect traffic type by 58% on average for the four ML models.

The remainder of this paper is structured as follows; in Section 2, the previous related work to this paper is covered. Section 3 illustrates the methodology used to train, implement, and evaluate the trained ML to detect IoT attacks. In Section 4, four different machine learning models: Neural Networks, Gaussian Naive Bayes, Decision Trees, and Support Vector Machine are trained to detect IoT attacks. It shows that Decision Trees and Neural Networks are better at detecting IoT attacks. In Section 5, the performance of the ML models using two different input methods is compared, one using all dataset attributes and the second using only six attributes. Measuring the performance of two different datasets, it is shown that the performance of the trained ML models is almost similar while time is reduced when using six attributes only by 58% on average for all tested models. Section 6 concludes the paper.

2 RELATED WORK

Several studies have been conducted on analyzing network traffic recently. In addition to the most traditional IDS used which is signature based, machine learning utilization in IDS is widely being studied in recent years (Chaabouni et al. 2019). Restuccia et al. (2018) discussed the role of the Software Defined Networking (SDN), blockchain, and ML in securing IoT networks. Davis et al. (2018) were able to develop a model using autoencoders to detect attacks of botnets on IoT devices. They tested their model on a testbed of 9 devices using 10 different attacks. Their model utilized 115 features. Brun and Yin (2019) analyzed network attacks to develop ML model using Recurrent Neural Network (RNN) to detect the attacks. They collected data from a testbed of three devices and then modified the data using a simulator to simulate the attacks. Shukla (2017) developed IDS based on K-means ML model. They were able to achieve 70 to 90% accuracy in detecting IoT attacks. They tested their model on simulated network of 10 devices. A model using Bidirectional Long Short-Term Memory based Recurrent algorithm to detect botnet activity was developed by McDermott et al. (2018). They have utilized info feature in PCAP traces to feed their model. They were able to achieve 92 % accuracy in detecting the IoT attacks.

A multilevel intrusion detection model framework was developed by Yao et al. (2019). They tested their model on KDDCUP99 dataset. They were able to achieve 96.6% accuracy in detecting the attacks. A hybrid learning approach based on decision trees was used by Amouri et al. (2018) to develop IDS for IoT networks. They were able to achieve accuracy of 100% on a simulated environment of 35 devices. Anthi et al. (2019) developed a three-layer IDS. They used 121 features as input to their model. They have tested their model on a testbed of 8 devices. They were able to achieve 99.97% accuracy in detecting attacks. Yu et al. (2011), developed IDS based on K-Random Forest ML model. They were able to achieve 96% accuracy in detecting IoT attacks. They tested their model on KDDCUP99 dataset. Alhakami, Alharbi et al. (2019) developed a Bayesian based IDS. They used 42 features as input to their model when using KDDCUP99 dataset. They were able to achieve 84.06% accuracy in detecting attacks. SVM-based classifier was used by Jan et al. (2019) to develop IDS for IoT networks. They have used a CICIDS2017 dataset with 40 features. They were able to achieve accuracy of 98%.

For convergence of classifier parameters, Senthil et al. (2021) developed a fast-learning network with particle swarm optimization. Despite the positive results, the system's complexity is too high to be applied to sensor nodes due to their limited processing and energy storage capabilities. In their suggested intrusion detection system, Moukhafi et al. (2018) coupled a hybrid genetic algorithm and support vector machine with particle swarm optimization for feature subset selection. This system was nearly 100 percent accurate in distinguishing DoS attacks from other sorts of attacks; however, it was unable to distinguish typical class signals from other types of signals. Vijayanand et al. (2018) proposed a hybrid feature-selection method based on mutual information and genetic algorithm for support vector machine-based classifier in order to increase classification accuracy. They also shown in their experimental results that support vector machine - based classifier is capable of outperforming an artificial neural network (ANN). When the classifier was trained with 400 samples, they attained accuracy of 96 percent. Both the genetic algorithm and mutual information could require as few as three informative features. The findings revealed that obtaining similar outcomes using both a genetic algorithm and mutual information could require as few as three informative features. This strategy, however, does not seem to be a feasible option, given the power and computation-cost limits of IoT devices.

In (Restuccia et al., 2018), (Meidan et al., 2018) and (Brun and Yin, 2019) researchers did not include the accuracy achieved by their research. In (Shukla, 2017), (McDermott et al., 2018) and (Yao et al., 2019) researchers achieved low accuracy comparing to the other reviewed studies. In (Yao et al., 2019), researchers were able to achieve 100% accuracy however it is based on network fluctuations; hence, it requires devices to be irresponsive to detect the attacks which is not the case with every IoT device. In (Anthi et al., 2019), researchers were able to achieve 99% accuracy however the number of features used is high (121) comparing to the other studies. None of the studies reviewed verified their models on multiple data sets.

The models reviewed are based on a high number of features; however, the model used in this paper is based on six attributes only. Other work does not compare the detection of IoT attacks on different datasets however the model developed in this paper is tested on two different datasets. To the authors knowledge, this is the first paper that compares performance of several ML algorithm in DoS detection on different datasets with focusing on optimization of the input parameters to reduce the time and resource required to train the ML models.

3 METHODOLOGY

3.1 System Overview

Various assumptions regarding consumer IoT networks are made in the threat model (Figure 1). We assume the network includes a middle box device such as a home gateway router, that links the IoT network to other networks and analyzes traffic between IoT devices on the local area network and the Internet. This device will analyze, store, alter, and block any network communication that passes through it. This middle box handles all communication between LAN Wi-Fi devices and Internet-connected devices.

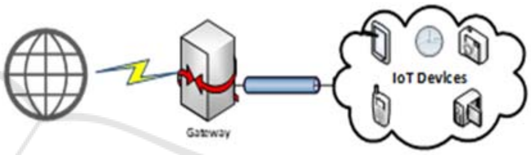

Figure 1: Middle box Approach for Capturing IoT traffic.

Our aim is to protect IoT devices from DoS attack traffic; hence they are connected to the middle box which enables them to send and receive network traffic, including attack traffic. In addition, each device may counter DoS attacks, and the duration of consecutive attacks may vary. Traffic is analyzed in time series of 1 second which is shorter than typical DoS attacks to avoid (Kabir, Hu, Wang & Zhuo, 2018).

The programing logic of the trained model is mentioned in Algorithm 1. Algorithm 1 takes the data captured by the middle box as well as the instructed ML model such as NN, SVM, NB or DT. After that, Algorithm 1 starts to train the model using the captured data. Once trained model is available, Algorithm 1 starts to analyze the traffic, if anomaly/attack is detected then traffic is blocked if not then traffic is allowed.

Python is the language chosen to implement the model. Google Colab is the execution environment chosen to implement, train and test the models. At the time of the experiment, the Google Colab allowed the use of 25.6 GB of RAM, Disk space of 225.89 GB and offered Intel(R) Xeon(R) CPU @ 2.20GHz.

Algorithm 1: Machine learning based IDS programming.

```
INPUTS: Datasets, machine learning models
OUTPUT: machine learning based IDS
PROCEDURE:
while True do
    Read traffic going through the middle box
    Apply machine learning model
    Train the machine learning model
    if Trained machine learning model is available then
        Test the traffic
        if attack is detected then
            Block traffic
        else
            Allow traffic
        end if
    else
        Wait for creating a trained machine learning model
    end if
end while
```

3.2 Data Sample

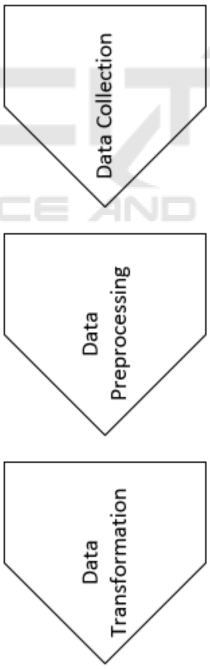

Figure 2: Data preparation process.

Figure 2 shows the flow followed to prepare the data sample for training and testing the models. A subset of an open dataset was used in this test (Hamza, Gharakheili, Benson, and Sivaraman, 2019). This data was collected from an instrumented living lab with 10 IoT devices emulating a smart environment. The sample data used include several types of IoT devices, including motion sensors, cameras, plugs, lights, and appliances. This data is in the form of Packet Capture (PCAP) traces. The data contains the following type of attacks Address Resolution Protocol (ARP) Spoofing, TCP Sync, Ping of Death, UDP Device, TCP Sync Reflection, SMURF, Simple Network Management Protocol (SNMP), Simple Service Discovery Protocol (SSDP).

Table 2 shows that UDP and TCP protocols represent more than 80% of the data. Table 3 shows that the highest used services are TCP, NTP and UDP. The number of attack cases represent 1.7% of dataset as shown in Table IV.

Table 2: Percentage of each protocol type in sample dataset.

Protocol Type	Percent
ICMP	2.8
IGMP	0.1
TCP	32.3
UDP	48.8
NULL	16.0

Table 3: Percentage of each service in sample dataset.

service	Percent
TCP	21.9
NTP	10.8
UDP	9.4
GQUIC	5.4
ARP	4.8
ICMP	2.8
TLSv1.2	2.8
SSHv2	2.7
TLSv1	2.6
DNS	2.4
STUN	1.0
HTTP	0.6
HTTP/XML	0.6
ICMPv6	0.3
MDNS	0.2
IGMPv2	0.1
MQTT	0.1
Others	31.4

Table 4: Percentage of attack cases in sample dataset.

Attack	Percent
0	98.3
1	1.7

3.3 Selected Features

The data set was preprocessed to be entered into the different machine learning algorithms as input. The following attributes were extracted from the dataset:

- **Attack:** The data in benign state were tagged with 0 and during attack with 1.
- **Protocol_type:** protocol type of the connection i.e. TCP, UDP, and ICMP
- **Service:** http, ftp, smtp, telnet, etc.
- **Length:** total bytes sent or received in one connection.
- **Count:** sum of connections to the same destination IP address occurred in the past 2 seconds.
- **Srv_count:** sum of connections to the same destination port number occurred in the past 2 seconds.

Table 5 shows sample values for each of the selected parameters.

Table 5: Sample values for the selected features.

Parameter	Sample Value
attack	0
protocol_type	ICMP
service	MQTT
length	466
count	13
srv_count	599

3.4 Performance Evaluation

The following metrics were calculated for each model:

- Accuracy $=\frac{TP+TN}{TP+FP+FN+TN}$
- Precision $=\frac{TP}{TP+FP}$
- Recall $=\frac{TP}{TP+FN}$
- F1 Score $=\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

where

- TP = the number of True Positives
- TN = the number of True Negatives
- FP = the number of False Positives
- FN = the number of False Negatives

Accuracy, precision, recall, and F1 are used to evaluate the four ML algorithms chosen for this study.

4 EXPERIMENTATION

We tested four machine learning algorithms to classify normal and DoS attack traffic.

4.1 Gaussian Naive Bayes Model

Gaussian Naive Bayes Model was implemented using Equation (1). The equation assumes that the six chosen variables $(a_1, a_2, \ldots, a_n|c)$ are independent. The class to be predicted "c" is category of the traffic which is "Attack" or "Benign".

$$P(E|c) = P(a_1, a_2, \ldots, a_n|c) = \prod_{i=1}^{n} P(a_i|c) \qquad (1)$$

The results are shown in Table 6. It achieved 0.899 accuracy in detection of attacks. The main reason for having low accuracy value is this this model's nature, which assumes that attributes are independent; however, insecurity is related to each other. For example, ping of death will result in both high ICMP send packets and high ICMP response packets; however, this model cannot relate send and response packets to each other.

Table 6: Gaussian Naive Bayes Model Classification Results.

Metric	Results
Accuracy	0.89914
F1-Score	0.89808
Precision	0.91601
Recall	0.89914

4.2 Decision Tree Model

We have used C4.5 algorithm in implementing the decision tree model which is represented by Equation (2):

$$Info(S) = -\sum_{i=1}^{k}(\left(\frac{freq(C_i,S)}{|S|}\right).log_2\left(\frac{freq(C_i,S)}{|S|}\right)) \quad (2)$$

When implementing Decision Tree model, it achieved 0.98 in both accuracy and precision in detection of attacks as shown in Table 7. The ability of the decision tree model to breakdown the data into manageable parts is the reason for this model to achieve higher results in this classification problem.

Table 7: Decision Tree Classification Results.

Metric	Results
Accuracy	0.98244
F1-Score	0.98243
Precision	0.98303
Recall	0.98244

4.3 Support Vector Machine Model

The SVM model was implemented the “Radial Basis Function” RBF kernel which is represented by the Equation (3):

$$K(x, x') = exp(-\gamma\|x - x'\|^2) \quad (3)$$

Lower accuracy and precision were achieved when implementing SVM model as shown in Table 8. The main reason for having low precision value is having a nonlinear separable problem so attack and benign state cannot be decided in a linear approach.

Table 8: SVM Classification Results.

Metric	Results
Accuracy	0.89963
F1-Score	0.89860
Precision	0.91638
Recall	0.89963

4.4 Neural Network Model

The topology of our ANN consisted of three hidden layers of size 8 nodes, 4 nodes and 2 nodes. The output layer is of a single node. Each node is using the following Equation (4) to calculate the weight:

$$output = f(\sum_i w_i x_i) \quad (4)$$

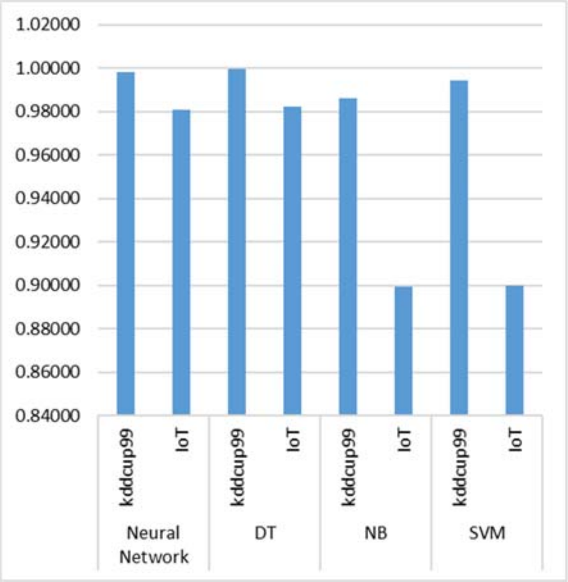

Figure 3: Comparing the accuracy result of the IoT dataset with KDDCUP99 dataset.

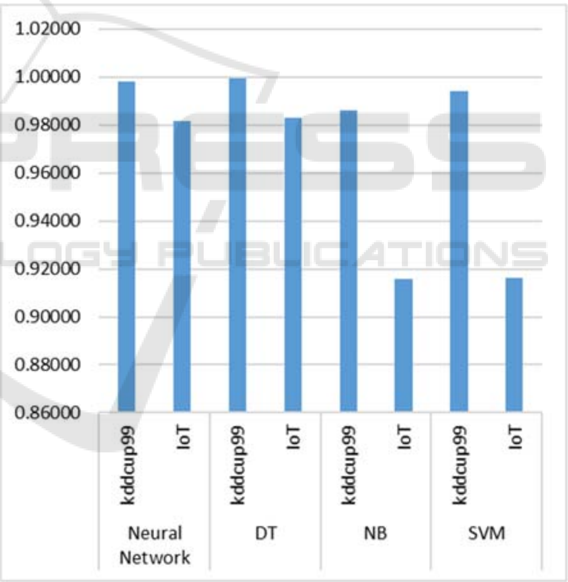

Figure 4: Comparing the precision score result of the IoT dataset with KDDCUP99 dataset.

Here is the Equation (5) for the activation function ReLU:

$$ReLU(x) = \max(0, x) \quad (5)$$

Results are shown in Table 9. It achieved an accuracy of 0.98 in the detection of attacks with a precision of 0.98. The main reason that the neural network achieved high accuracy results is its ability to find a hidden relationship between input and output in a non-linear approach. Also, because the classification is limited to two outputs, it performed very well.

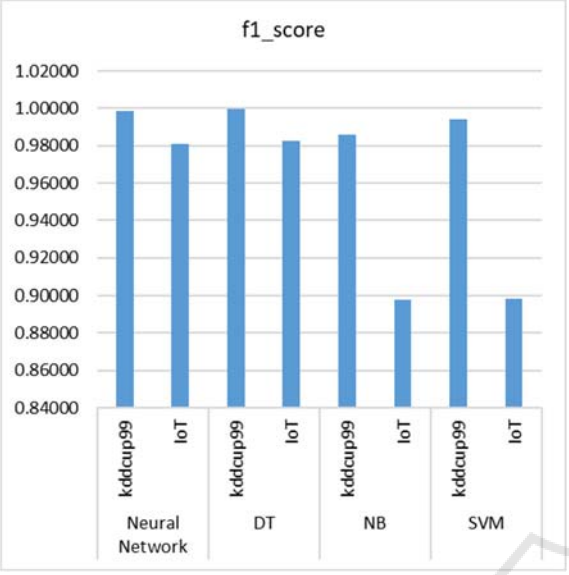

Figure 5: Comparing the f1_score result of the IoT dataset with KDDCUP99 dataset.

Table 9: ANN Classification Results.

Metric	Results
Accuracy	0.98103
F1-Score	0.98102
Precision	0.98172
Recall	0.98103

5 COMPARING WITH KDDCUP99

When comparing intrusion detection in both kddcup99 and IoT sample data set, neural network and DT maintained similar performance as shown in Figure 3, Figure 4 and Figure 5. NB and SVM failed to maintain the similar performance in intrusion detection in both kddcup99 and IoT sample data set.

Figure 6 shows that the time for identifying the test sample traffic type is reduced by more 70% for ANN, 56% for SVM, 75% for DT and 30% for NB. This result in average of 58%-time reduction for the four models.

Based on the above findings, we can see that there is decent reduction in execution time while maintain similar performance for Neural Network and DT models.

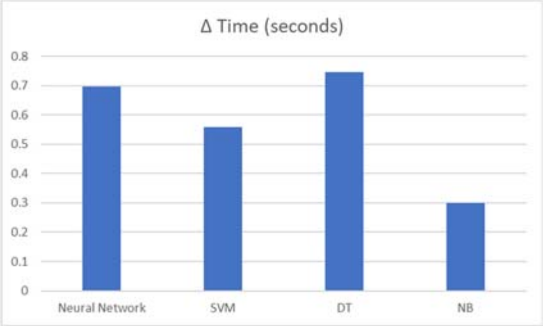

Figure 6: The difference in time for execution between using all variables and the six chosen variables.

6 CONCLUSION

The DoS attack is one of the security problems that affect IoT devices. This paper explored the use of different machine learning algorithms, including Gaussian Naive Bayes. SVM, Decision Trees and ANN algorithms in detecting DoS attacks in IoT networks. It also found that ANN and Decision Trees performs the best in detecting DoS attacks. Limited research was done on this area and most of the reviewed studies were not using a sample of IoT networks. Other studies did not include details all performance metrics such as accuracy precision and recall percentages.

We have used several machine learning models to detect the DoS attacks in IoT networks. Gaussian Naive Bayes and SVM machine learning models achieved low precision compared to ANN and Decision Trees due to various reasons related to the nature of the models. ANN and Decision Trees achieved more than 0.98 accuracy and precision. We were able to reach 98 percent accuracy in intrusion detection for IoT attacks. We developed a revolutionary method of identifying attacks based on only six attributes, which cuts the time it takes to train the selected ML Models by 58% on average.

This research shows that machine learning techniques such as ANN and Decision Trees, when taught with low-dimensional characteristics, can distinguish between normal IoT device traffic and DoS attack traffic. This finding encourages more research into detecting DoS in real-world IoT networks.

REFERENCES

Alhakami, W., ALharbi, A., Bourouis, S., Alroobaea, R., & Bouguila, N. (2019). Network anomaly intrusion detection using a nonparametric bayesian approach and feature selection. *IEEE Access, 7*, 52181-52190. doi:10.1109/ACCESS.2019.2912115

Amouri, A., Alaparthy, V. T., & Morgera, S. D. (Apr 2018). Cross layer-based intrusion detection based on network behavior for IoT. Paper presented at the 1-4. doi:10.1109/WAMICON.2018.8363921 Retrieved from https://ieeexplore.ieee.org/document/8363921

Andrea, I., Chrysostomou, C., & Hadjichristofi, G. (Jul 2015). Internet of things: Security vulnerabilities and challenges. Paper presented at the 180-187. doi:10.1109/ISCC.2015.7405513 Retrieved from https://ieeexplore.ieee.org/document/7405513

Anthi, E., Williams, L., Slowinska, M., Theodorakopoulos, G., & Burnap, P. (2019). A supervised intrusion detection system for smart home IoT devices. *IEEE Internet of Things Journal, 6*(5), 9042-9053. doi:10.1109/JIOT.2019.2926365

Bagaa, M., Taleb, T., Bernabe, J. B., & Skarmeta, A. (2020). A machine learning security framework for iot systems. *IEEE Access, 8*, 114066-114077. doi:10.1109/ACCESS.2020.2996214

Brun, O., & Yin, Y. (Jun 2019). Random neural networks and deep learning for attack detection at the edge. Paper presented at the 11-14. doi:10.1109/ICFC.2019.00009 Retrieved fromhttps://ieeexplore.ieee.org/document/8822151

Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., & Faruki, P. (2019). Network intrusion detection for IoT security based on learning techniques. *IEEE Communications Surveys & Tutorials, 21*(3), 2671-2701. doi:10.1109/COMST.2019.2896380

Davis, B. D., Mason, J. C., & Anwar, M. (2020). Vulnerability studies and security postures of IoT devices: A smart home case study. *IEEE Internet of Things Journal, 7*(10), 10102-10110. doi:10.1109/JIOT.2020.2983983

Fadul, M., Reising, D., Loveless, T. D., & Ofoli, A. (2021). Nelder-mead simplex channel estimation for the RF-DNA fingerprinting of OFDM transmitters under rayleigh fading conditions. *IEEE Transactions on Information Forensics and Security, 16*, 2381-2396. doi:10.1109/TIFS.2021.3054524

Hamza, A., Gharakheili, H., Benson, T., & Sivaraman, V. (Apr 3, 2019). Detecting volumetric attacks on loT devices via SDN-based monitoring of MUD activity. Paper presented at the 36-48. doi:10.1145/3314148.3314352 Retrieved from http://dl.acm.org/citation.cfm?id=3314352

Jan, S. U., Ahmed, S., Shakhov, V., & Koo, I. (2019). Toward a lightweight intrusion detection system for the internet of things. *IEEE Access, 7*, 42450-42471. doi:10.1109/ACCESS.2019.2907965

Kabir, E., Hu, J., Wang, H., & Zhuo, G. (2018). A novel statistical technique for intrusion detection systems. *Future Generation Computer Systems, 79*, 303-318. doi:10.1016/j.future.2017.01.029

McDermott, C. D., Majdani, F., & Petrovski, A. V. (Jul 2018). Botnet detection in the internet of things using deep learning approaches. Paper presented at the 1-8. doi:10.1109/IJCNN.2018.8489489 Retrieved from https://ieeexplore.ieee.org/document/8489489

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-BaIoT-network-based detection of IoT botnet attacks using deep autoencoders. *IEEE Pervasive Computing, 17*(3), 12-22. doi:10.1109/MPRV.2018.03367731

Mishra, P., Varadharajan, V., Tupakula, U., & Pilli, E. S. (2019). A detailed investigation and analysis of using machine learning techniques for intrusion detection. *IEEE Communications Surveys and Tutorials, 21*(1), 686-728. doi:10.1109/COMST.2018.2847722

Moukhafi, M., El Yassini, K., & Bri, S. (2018). A novel hybrid GA and SVM with PSO feature selection for intrusion detection system. *International Journal of Advances in Scientific Research and Engineering, 4*(5), 129-134. doi:10.31695/IJASRE.2018.32724

Restuccia, F., D'Oro, S., & Melodia, T. (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. doi:10.1109/JIOT.2018.2846040

Senthil, G. A., Raaza, A., & Kumar, N. (2021). Internet of things multi hop energy efficient cluster-based routing using particle swarm optimization. *Wireless Networks, 27*(8), 5207-5215. doi:10.1007/s11276-021-02801-0

Shui Yu, Wanlei Zhou, Doss, R., & Weijia Jia. (2011). Traceback of DDoS attacks using entropy variations. *IEEE Transactions on Parallel and Distributed Systems, 22*(3), 412-425. doi:10.1109/TPDS.2010.97

Shukla, P. (Sep 2017). ML-IDS: A machine learning approach to detect wormhole attacks in internet of things. Paper presented at the 234-240. doi:10.1109/IntelliSys.2017.8324298 Retrieved from https://ieeexplore.ieee.org/document/8324298

Vijayanand, R., Devaraj, D., & Kannapiran, B. (2018). A novel intrusion detection system for wireless mesh network with hybrid feature selection technique based on GA and MI. *Journal of Intelligent & Fuzzy Systems, 34*(3), 1243-1250. doi:10.3233/JIFS-169421

Yao, H., Fu, D., Zhang, P., Li, M., & Liu, Y. (2019). MSML: A novel multilevel semi-supervised machine learning framework for intrusion detection system. *IEEE Internet of Things Journal, 6*(2), 1949-1959. doi:10.1109/JIOT.2018.2873125.