National College of Ireland

# Effective approach for Malware Detection using Machine Learning and Deep Learning for IoT-Devices.

MSc Research Project

MSc In Cybersecurity

Devika Rajiv Galinde

Student ID: 20177283

School of Computing

National College of Ireland

Supervisor: Prof. Arghir Nicolae Moldovan

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

**Student Name:** Devika Rajiv Galinde

**Student ID:** 20177283

**Programme:** MSc. in Cyber Security **Year:** 2022-2023

**Module:** MSc Research Project/Internship

**Supervisor:** Prof. Arghir Nicolae Moldovan

**Submission Due Date:** 29/05/2023

**Project Title:** Effective Approach for Malware Detection using Machine Learning and Deep Learning for IoT-Devices.

**Word Count:** 8304 **Page Count** 23

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:** Devika Rajiv Galinde

**Date:** 29/05/2023

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies) | □ |
| **Attach a Moodle submission receipt of the online project submission,** to each project (including multiple copies). | □ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | □ |

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# Effective approach for Malware Detection using Machine Learning and Deep Learning for IoT-Devices.

Devika Rajiv Galinde
Student ID : 20177283

**Abstract**

The Internet of Things (IoT) is an amazing innovative technology created and developed by humans. As, the technology is growing fast, the new devices are coming in the market with excellent features and hence there is high possibility of the applications or the IoT devices getting vulnerable to the cyber treat to the environments. The IoT devices are smart devices which communicates through the internet and therefore, it possessing high chances of getting trapped by the cyber criminals. The related work in this research focuses on the malware detection with multiple Deep Learning(DL) and Machine Learning (ML) models. As, there are various malware variants in the environment, it is the primary focus to prevent it from any attack causing severe loss to the industries. This research aimed in detecting malware with a high accuracy using ML and DL models. For Decision Tree the accuracy is 0.999 ~ 1. The Decision tree seems to be over fitted and so Random Forest algorithm is introduced and the accuracy achieved by RF is 0.998% .

# 1 Introduction

## 1.1 Research Background

Currently, the continuous growth of the Internet of the Things (IoT) has made and shown a remarkable progress in the field of information technology. Therefore, it is being the most crucial engine for the globe's economic growth. In 1998, the British researcher and the cofounder of the MIT (Massachusetts Institute of Technology) had described the idea of the IoT. They described the IoT as set of all the things which are connected to the Internet via the sensors like RFID (Radio Frequency Identification) in order to achieve the intelligent identification along with the management. There are various definition of IoT followed since then in order to highlight the vital scope if the IoT applications. Considering the facts, the IoT can also be considered as a family of the technologies which actually aims to build any type of device or the objects which can connect to the internet and can allow the device to have all the feature access that can be used in an network. There are basic two properties of IoT systems which includes the control and the monitoring. Controlling actually means that remotely controlling any devices/ objects with the help of internet and no specific technology.

Monitoring focusses on the capability of the device to adapt as a behavior that of a sensor along with ability to produce sensitive information about encompassing environment or itself. IoT are also known as the intelligent devices or machines [1].
The IoT smart devices are rapidly growing in the market and industries and mainly focus on interconnecting the computer devices which are integrated in the everyday objects via the internet and they communicate by send and receiving the data. The main advantages are we can actually empower out laptops/computer devices in order to collect information about the surroundings or the environment and this can be done without depending on the humans and thus by also processing the data collected we can also reduce the loss, extravagances and the cost required to do the tasks. The IoT mainly allows the interaction amongst the digital and the physical world. Also, the digital world basically interacts between the physical world through the actuators and the sensors. Thus these sensors are supposed to gather the information/data which should be stored as well as processed. Cloud or remote server or at the network edge the data processing can take place[2].

## 1.2 Problem Definition

At the current situation, IoT devices are used by everyone to fulfil their daily tasks. The IoT devices may include some wearable devices, light bulbs , home appliances, also some of the incorporate IoT technologies like the Google Home, Amazon Echo, Philips Hue, etc. The Figure 1, shows the wide-spread domains where the successful implementations of the IoT has be done and demonstrated [3].

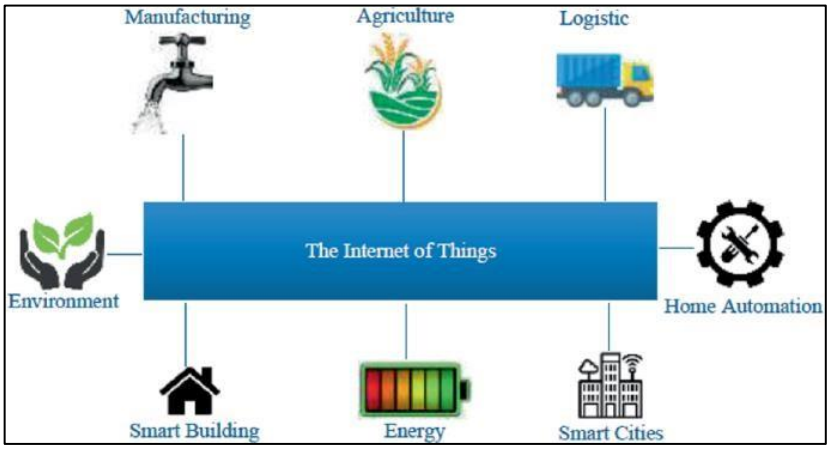


**Figure 1 : Widespread domains of IoT[3]**

**Research Question 1.** What can be the effective approach to detect the malware using Deep Learning and Machine Learning technique for the IoT devices?

With such new and growing technology, however since these are connected and communicate through the internet there are some vulnerabilities that are also being introduced and that can exploited and as well leveraged by the malicious users. It is said that one of the common vehicle of the exploitation is a malware or a malicious software or there can be some security-related issues. In the information system malwares actually are extremely harmful and can actually compromise the CIA triad (Confidentiality, Integrity and also Availability). Malwares are specifically and specially designed software which can harm the users system without their

permissions. The main target of a malware is to gain access of the system or to damage it without any knowledge of the user/owner of the system. Malwares are of different variants in types. Those are Adware, Viruses, Spywares, Ransomware, Worms, Trojans , and many more various types of codes which are malicious in the nature. Some of the most common malware attacks DDoS ( Distributed Denial of Services) attacks which happens on the large scale and the consequence of it is the worst. Mirai Malware is another example of a malware which was found and introduced in August 2016, which turned the networked devices which were running on Linux into the multiple group of bots which led to the network attack on the large-scale. Another example of such type of malware is Hajime , which is extreme prominent in actually causing severe security issues in the IoT devices. It is known to be similar to the Mirai although it spreads through the Telnet ports which are open and unsecured and then it uses the same exact usernames and the password tables which are used by the malware Mirai [4].

There are various security requirements of the IoT devices discussed [5], with respect to its various operational levels which are as follows :

- Information Level : Security should be guaranteed by the requirements like Integrity, Confidentiality, Privacy and Anonymity.
- Access Level : Security mechanisms in order to control the network access with the functionalities like Authentication, Access Control and the Authorization.
- Functional Level : It defines few security requirements like Resilience and Self Organization.

There are various malware detection techniques[4] and approaches as shown below :

- Image Based Malware Recognition.
- Blockchain Technology.
- Mobile Malware.
- Machine Learning.

Thus, this paper focuses the malware detection technique on the Machine Learning(ML). It is basically known as a key in order to secure the vast applications. It is not possible to follow the traditional method to detect the malware with the malware detection software's or the antimalware software's just because of the malware evolution. Thus ML proves and provides the best possible detection techniques for the malwares [4].

This paper is based and focused on malware detection for the IoT devices using the Machine Learning and Deep Learning algorithms using Multiclass classification and Binary Classification. There are various ways and tools available for detecting the malwares although it's not an efficient way .The few anti-malware software tools are Bit defender, McAfee, Virus total, etc. Machine Learning proves to be the most efficient method to detect the malwares. Malwares have different behavior's hence not all the anti-malware tools can detect them. There are various algorithms in ML/DL which can be used to detect the malwares. The models needs to be trained and tested in order to detect the malicious activities.

The aim and contribution along with novelty is as discussed below :

- The research aims to use the Machine Learning and Deep Learning Models such as
- Support Vector Machine(SVM)Naive Bayes (NB) , ANN (Artificial Neural Network), Decision Tree (DT), Random Forest(RF).

- The novelty of this research project was to improve the accuracy of the models by comparing it with the previous papers. Additionally added Binary Classification, which was compared along with the Multiclass Classification.
- The major contribution was adding the binary classification, adding the Random Forest model, improving the accuracies of the ML/DL models. Improving the data-preprocessing which resulted the best results for the models in order to detect the malwares. This is more discussed in the Section 3.

Each having multiclass classification and binary classification. Furthermore, the previous study/research will be discussed more in detail in the Section 2 of this paper. Section 3 comprises of methodology which focuses on the Section 2 of this paper. Section 3 comprises of the methodology, classifications , algorithms, dataset , data pre-processing. Design Specification will be discussed in Section 4. Followed with Implementation in Section 5 , Evaluation in the Section 6. The Section 7, highlights the Conclusion and Future work. The Section 8, includes the references for this entire research paper.

# 2 Related Work

In this particular section of the related work, the previous researches will be more discussed and the challenges they actually faced. The related work will emphasis on the previous machine learning and deep learning techniques which were actually used for the malware detection for the IoT devices focusing on the different datasets used by them. And then selecting one dataset for this implementation. The main motive for this research was to detect the malware for the IoT devices and to improve on the accuracies of the Machine Learning(ML) and Deep Learning (DL) algorithms.

## 2.1 ML/DL Models for Malware Detection.

There are various studies using different malware datasets for the purpose of malware detection for various network devices like android and IoT devices. The datasets can be found publicly are called public datasets and the datasets can also be manually created which are known as private datasets. There are various cyber-attacks which cause tremendous loss. One of the most known attack is DDoS (Distributed Denial of Service) , which affects and has already affected lot many IoT networks which led to tremendous data loss in the recent and past years. So for this, deep learning models are used and also evaluated using the CICIDS2017 dataset in order to detect the DDoS attack which also provided the highest accuracy of 97.16 % to compare with the machine learning algorithms. The paper as well proposed four different deep learning classification models such as Multilayer Perception (MLP) , 1d-CNN , LSTM, CNN (Convolutional Neural Network) + LSTM (Long Short Term Memory) . Of which the hybrid model i.e. CNN + LSTM performed best than other models with accuracy of 97.16% [6].
Another study was proposed by the author [7] , where DL-based novel architecture was proposed which actually classified the malware variants based on a hybrid model. The main study contribution by the author was to propose and create a newest hybrid model architecture which integrated the two-wide ranging network models which were pre-trained in optimized

manner. Thus the architecture also consisted of majorly 4 stages which were the data acquisition , DNN (Deep Neural Network ) architecture and its design along with the evaluation of trained DNN. The proposed research was tested on the different datasets naming Microsoft BIG 2015 , Malevis and Malimg datasets. The experimental results proved that the method selected for the classification and detection of malware was successfully done with resulting in high accuracy.

In other study [8] author proposed DL based malware classification as well as the detection which was based on the DL based cross architecture for IoTs. Detection of IoT malwares and classification of the IoT malware families was implemented by using Bi-GRU-CNN (Bidirectional-Gated Recurrent Unit-Convolutional Neural Network) which was a DL based model and was done using the ELF i.e. Executable and the Linkable Format (as Extensible Linking Format formerly named) input feature as a binary file byte sequences. In the addition, another DL based model combinations , RNN(Recurrent Neural Network) were also considered in order to evaluate the IoT malware detection and also its family classification performances. Those models performances were compared with the authors proposed model. The performance evaluation of the proposed approach had obtained 100% of accuracy for the detection of IoT malware and for the IoT malware family classification was 98%. The further evaluation of the proposed model with the input feature as the only byte sequence exhibited almost the similar performances as a byte sequence and also the CPU types as its input features which also showed that the proposed model was robust and also platform independent in order to classify as well as detect the IoT malwares and its families. Although some of the malware families in the dataset were not classified correctly since , the dataset used was imbalanced. Similarly [9], proposed EDIMA which is Early Detection of IoT Malware Network Activity which was done using the ML techniques which actually converted the sample file having 9,000,000 byte sequence into the byte sequence of 300. This was done using the SVD (Truncated Singular value composition). The RNN along with the LSTM was then applied on the byte sequence of 300 as the feature length in order to detect if the file had a malware or no malware(benign). For, the file byte sequence length less than 900000, the file is then padded with the zero sequences. The result reported by the author showed that the method for detection of the IoT malware achieved 91% accuracy. Whereas , the author as well proposed another model i.e. CNN based shallow model which has parallel convolutional layers and which converted byte sequences as an image for the purpose of feature set. Thus, this model had achieved 99.1% accuracy for the malware detection. Both the models required the preprocessing in order to obtain feature set. Also, the feature of byte sequence which was applied for the LSTM model performance was nominal.

According to [10], the machine learning and also the deep learning models were implemented in order to anomaly detect the malware for the IoT devices by using the IoT-23 dataset and they showed the multiclass classification. The ML/DL models used were Naïve Bayes, Support Vector Machine (SVM) , Decision Tree (DT) and ANN (Artificial Neural Network). Furthermore technically correcting for [10] is that it is not CNN (Convolutional Neural Network) but it's an ANN. There was another study for the malware detection using the machine learning [11], the main motive was to focus on the IoT networks and its security aspects. Multiple algorithms were used for the multiclass classification such as Random Forest (RF), Multi-Layer Perceptron (MLP), Naive Bayes (NB), AdaBoost (ADA), Support Vector Machine (SVM) and also variants of ANN class of the algorithms. The data was pre-processed

which consisted of data selection, the data was formatted and then the statistical correction and data splitting was done. After these steps the processed data was then fed to the ML/DL models. The data splitting was done randomly into the 80:20 ratio , where the 80% was the testing data and 20% was the training data. The algorithms were then compared on the accuracy, recall score , support score and the f-1 score. The labels were not used in the original format. Some limitations of [12] were of the technical nature. The dataset was split in the smaller parts and also the data was encoded which had less categories as compared to the original categories. Less categories were considered because of the computational problems. Thus, recommends to use all the categories of the original dataset and to have high configuration device.

The generative way in order to detect the cyber-attacks was done by [12], in which they performed methods which are generative way of deep learning such as the AAE (Adversarial Autoencoders) along with the BiGAN (Bidirectional Generative Adversarial Networks) in order to detect the intruders which were actually based on the analysis of the network data's. They used the IoT-23 full dataset version based on the IoT devices like the Philips Hue , Somfy door lock and the Amazon Echo. These devices was then used in order to train the generative DL models which detected various of malware attacks like the DDoS, including the botnets attacks Torii, Mirai and Okiru. They used 1.8 million of the network flows in order to train the different models. Both the AAE and the BiGAN models were able to achieve 0.99 of the F1score. Thus, the proposed paper[13], showed the limited set of IoT devices and the attacks , thus it is possible to use the generative deep learning methods like the BiGAN and the AAE in order to classify the attacks with relatively high accuracy. The results showed GAN based attacks are extremely good at detecting and classifying the attacks.

Another attack detection of the DDoS was implemented in the IoT devices based on the SVM [13].The algorithms such as the SVM , DT , and the RF were actually trained and also tested in order to classify the attacked packets of the DDoS. PCA (Principal Component Analysis) technique was used in order to improve the performances of the algorithms used. The experiment showed that the DT and the RF algorithms classified the DDoS packets extreme accurately as compared to the SVM. The dataset was selected based on IoT device and then it was pre-processed and then the same pre-processed data was set to the next of the PCA module. Later, the three ML algorithms were used to analyze the IoT -23 dataset and then it was trained and tested and further it was evaluated and compared with the Accuracy , Precision, Recall and F-1 score. The research also concluded that PCA as a feature selection can increase the ML algorithm performances.

The another study showed detection of the cyber-attacks and its traces for the IoT data [14]. The research proposed leveraging DL algorithms like LSTM , DAE , and MLP in order to detect the anomalies in the networks. The author also implemented dimensionality reduction approaches in the form of the DAE to decrease the number of the features which were used in the vector in order to train the classifiers. The LSTM cells were utilized or used I order to enhance the classification effectively. The approach was then tested on the heterogeneous and large dataset i.e. IoT-23 which consists the network traffic data from the IoT devices.

A modern analysis was done for the adding ML which was based on the methods of IoT Cyber security. In which the study examined the effectiveness of algorithms for intrusion and malware detection. The algorithms studied were SVM, KNN(K-Nearest Neighbor) and RF. The algorithms were trained and tested further using the dataset named Aposemat IoT-23, published in 2020. Some limitations in the study included testing for more SVM kernels like the

polynomial kernels in order to optimize the SVM results. Also the future kernel testing , can improve the accuracy for SVM in order to detect the malwares [15].

There was a study for the universal feature set for the IoT Botnet attack and its detection. This study tested and trained the algorithms for three datasets. The main motive of the author [16] , to use the universal feature set was that , it would help the ML algorithms to discriminate the actual botnet attacks from normal traffic. This was irrespective of underlying datasets. The performances of the ML algorithms were then compared with three different Botnet datasets.

In order to detect the attacks of a botnet , the paper proposed universal feature set which was based on the LR(Logistic Regression) and also frequency counting technique. The process involved few steps in order to detect a botnet attack. The steps were data acquisition in which the ML algorithms are trained , data processing was done for the datasets, Dataset Splition, feature extraction, dataset labelling , feature selection , models were trained , and then the botnet attack detection was done. For all the three datasets CICIDS2017, CTU-13, IoT-23 , the top most features were selected. The botnet attack detection was successful when the data was trained and tested. All the bonnet attacks were not detected , hence the future work aimed to detect the generalize ML algorithms.

There are many studies based on the anomaly malware detection similarly, the author [17] , proposed anomaly detection which was based on the flow and the flag features for the IoT networks using the Feed-Forward neural network. In which various types of the network flow features are analyzed. The research emphasis on the design as well as development detection system which detects the anomalous activities for the IoT networks using the Feed-Forward neural network which is based on the flow and flag features

The further research of this paper aimed to work on improving the accuracy for the ML/DL models for the efficient malware detection for SVM, NB, ANN, DT. The other proposal for the paper includes the Binary Classification and Multiclassification and with designed Random Forest Algorithm. The below Table 1, shows the summary table for the related works for the malware detection.

**Table 1 : Summary Table for Different Dataset used for Malware Detection**

| Reference | Dataset | ML/DL Models. | Best performing Algorithm | Best Accuracy. | F-1 Score |
|---|---|---|---|---|---|
| [6] | CICIDS2017. | MLP,1d-CNN,LSTM, CNN, CNN+LSTM | CNN+ LSTM | 97.16 % | - |
| [7]a | Microsoft BIG 2015 (Classification) | - | - | 94.88% | - |
| [7]b | Malevis(Classification) | - | - | 96.5% | - |
| [7]c | Malimg(Classification) | - | - | 97.98 % | - |
| [8] | Twisc Research centre[18] | For detection: BIGRU-CNN. For Classification :RNN, LSTM, GRU, Bi-RNN, Bi-LSTM, Bi-GRU, Bi-GRUCNN. | BI-GRUCNN | 100% | - |

| | | | | | |
|---|---|---|---|---|---|
| [9] | Worked at the ISP network at the user access gateway to detect the malware activities. | RNN with LSTM and CNN | RNN with LSTM | 91% | - |
| [10] | IoT-23 | NB, SVM, DT, and ANN. | DT | 73% | - |
| [11] | IoT-23 | RF, NB, MLP,ADA, SVM and also variants of ANN class of the algorithms | RF | 99.5% | - |
| [12] | IoT-23 | AAE and BiGAN- | AAE and BiGAN | - | 0.99 |
| [13] | IoT- 23 | DT , RF, SVM | DT | 99.5% | - |
| [14] | IoT-23 | RF, SVM , DAE, LSTM. | RF | 99 % | - |
| [15] a | Aposemat IoT-23(for intrusion detection) | RF,SVM, KNN | RF | 92.96 % | - |
| [15] b | Aposemat IoT-23(for malware detection) | RF,SVM,KNN | RF | 92.27 % | - |
| [16]a | CICIDS2017 (for DDoS detection ) | RF,LR,KNN,NB | RF and LR | 100% | - |
| [16]b | CTU-13 (for botnet detection) | RF,LR,KNN,NB | RF and LR | 100 % | - |
| [16]c | IoT-23 attack (Mirai detection) | RF,LR,KNN,NB | RF | 100 % | - |
| [17]a | BoT-IoT | - | - | 98.37% | - |
| [17]b | MQTT | - | - | 99.93% | - |
| [17]c | MQTTest | - | - | 99.96 % | - |

# 3 Research Methodology

This research paper proposes an effective way to detect malwares for the IoT devices using the Machine Learning(ML) and Deep Learning(DL) models. The dataset selected for study was the IoT- 23 Dataset. The main reason to work on this dataset , it being the new dataset to work and research. The research proposes the ML and DL models with Binary classification and Multiclass Classifications. Below Figure 2, shows the block diagram for the malware detection using the ML/DL models.

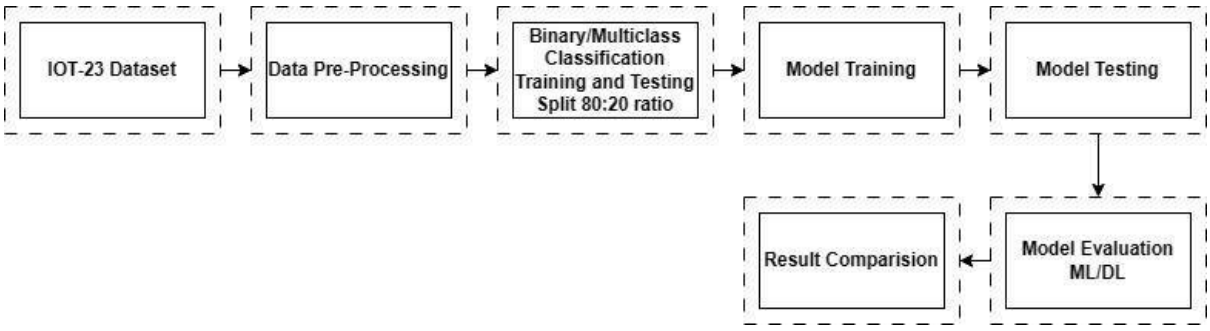


**Figure 2 : Block diagram for the research methodology.**

The main aim of the project was to detect the malwares for the IoT devices using few ML and DL methods. Focused on improving the accuracy for multiclass classification [10] , for all the models used NB, DT, SVM and ANN (Technically corrected, the author mentioned it as CNN, although after research understood that's its ANN). Also additionally added RF algorithm and the binary classification.

The IoT-23 dataset lighter version was downloaded and then the data was preprocessed with 14 features which generated the .csv file named " iot23_combined_14_features" which was then transformed into the training and testing for the ratio of 80:20 for the binary as well as the multiclass classification. There are 5 models trained and tested for binary classification and multiclass classification they are RF, DT , ANN , SVM and NB. The ML and DL models were evaluated and then the results were compared for the best accuracies.

***Binary Classification :***

For binary classification, the labels were grouped into two categories: "Benign" and "Malware." All the attack labels were combined into the "Malware" category. The same five machine learning algorithms, Decision Tree, Random Forest, Support Vector Machine (SVM), Artificial Neural Network (ANN), and Naïve Bayes, were evaluated for their effectiveness in detecting malware in a binary classification setting. The dataset was split into training and testing sets in an 80:20 ratio. The labels shown in table below were replaced as Malware. **Table 2 : IoT Dataset Original Labels**

| **Dataset** | **Labels** |
|---|---|
| IoT- 23 | PartOfAHorizontalPortScan", "Okiru", "DDoS", "C&C", "Attack", "C&C-HeartBeat", "C&C-FileDownload", "C&C-Torii", "C&C-HeartBeatFileDownload", "FileDownload", and "C&C-Mirai" |

- **Decision Tree:** A Decision Tree classifier was developed and evaluated for binary classification. The model's performance was measured using accuracy, precision, recall, and F1-score.
- **Random Forest**: A Random Forest classifier was developed and evaluated for binary classification. This classifier's performance was also measured using accuracy, precision, recall, and F1-score.

- **Support Vector Machine (SVM):** An SVM classifier was developed and evaluated for binary classification. The model's performance was measured using accuracy, precision, recall, and F1-score.
- **Artificial Neural Network (ANN):** The ANN model was adapted for binary classification by changing the output layer to have two nodes instead of twelve, with a SoftMax activation function. The model's performance was measured using accuracy, precision, recall, and F1-score.
- **Naïve Bayes:** A Naïve Bayes classifier was developed and evaluated for binary classification. The model's performance was measured using accuracy, precision, recall, and F1-score.

***Multiclass Classification Experiments:***

For multiclass classification, the labels were kept in their original categories. The same five machine learning algorithms, Decision Tree, Random Forest, Support Vector Machine (SVM), Artificial Neural Network (ANN), and Naïve Bayes, were evaluated for their effectiveness in detecting malware in a multiclass classification setting. The dataset was split into training and testing sets in an 80:20 ratio.

## 3.1 Dataset : IoT-23

The dataset was studied properly in order to further implement and get the malware detection. IoT-23 is actually a recent dataset which was published in the January 2020 , which has the captures which ranges from 2018 to 2019.This dataset is actually of the network traffic from the IoT devices. It consists of the 20 malware captures which are executed in the IoT devices and 3 malware captures which are the honeypots as benign for the traffic of IoT devices. The network traffic of the Internet of Things (IoT) was obtained within the Stratosphere Laboratory, which is affiliated with the AIC group, FEL, CTU University, located in the Czech Republic.
There are 2 IoT-23 datasets provided , one is the full IoT-23 dataset which is of 21GB and other is the lighter version which is of 8.7 GB which consists the labelled flows which are without the pcap files. The dataset used for this is the lighter version which is of 8.8 GB. After downloading the lighter version dataset, each of the captures have a folder which contains the README.md and conn.log labelled file. The README.md file contains pertinent data regarding capture and the malware, including the potential names of the malware , md5, sha1, and sha256 hash values of the malware binary. Additionally, the document provides the duration of the capture in the seconds, also hyperlink to the VirusTotal malware file, and a brief description of the contents within the folder. The conn.log labelled file is actually the Zeek conn.log file which is actually obtained by executing the Zeek network analyzer which uses the original pcap files [19]. The below table 3, shows the attack labels of the IoT-23 dataset.

**Table 3 : Attack Labels of IoT-23 dataset.**

| Attack Labels | Description |
|---|---|
| Attack | This label indicates there are different types of attacks from the malware infected device to the another host. |

| | |
|---|---|
| Benign | This label indicates no malware or suspicious or malicious activities in the connections. |
| C&C | The infected/malicious device is connected to the CC server. |
| DDoS | This label shows that Distributed Denial of Service attack is getting executed in the malicious/infected device. |
| FileDownload | This label shows that the file gets download to the malicious/ infected device. |
| HeartBeat | This label shows/ indicates the packets are sent on the connections and those are used to keep the track on the malicious/infected host by the server named C&C. |
| Mirai | This label defines that there are characteristics of the Mirai Botnet present in the connections. |
| Okiru | This label actually indicates to have the characteristics of the Okiru Botnet in the connections. |
| PortOfAHorizontalPortScan | This label indicates to do a task of horizontal scanning in order to gather all the information and then further perform attack in the connections. |
| Torii | This label indicated to have the characteristics of the Torii Botnet in the connections. |

Also, the table 4, below shows the variables of the Zeek files.

**Table 4 : Variables and definition of the Zeek files.**

| | |
|---|---|
| ts | This is the time of the first packet |
| uid | A unique identifier of the connection |
| id | The connection's 4-tuple of endpoint addresses/ports |
| proto | The transport layer protocol of the connection |
| service | An identification of an application protocol |
| duration | How long the connection lasted |
| orig_bytes | The number of payload bytes the originator sent |
| resp_bytes | The number of payload bytes the responder sent |
| conn_state | Possible connection state values |
| local_orig | If the connection is originated locally, this will be T |
| local_resp | If the connection is responded locally, this will be T |
| missed_bytes | Indicates the number of bytes missed in content gaps |
| history | Records the state history of connections as a string |
| orig_pkts | Number of packets that the originator sent |
| orig_ip_bytes | Number of IP level bytes that the originator sent |
| resp_pkts | Number of packets that the responder sent |
| resp_ip_bytes | Number of IP level bytes that the responder sent |
| tunnel_parents | uid values for any encapsulating parent connections |
| orig_l2_addr | Link-layer address of the originator |

## 3.2 Data Pre-Processing :

The network traffic of the Internet of Things (IoT) was obtained within the Stratosphere Laboratory, which is affiliated with the AIC group, FEL, CTU University, located in the Czech Republic. The dataset contains network traffic data collected from a range of IoT devices and includes both benign and malicious traffic. The pre-processing steps used in this study were adapted from the work of [10] and [20]

The data-pre-processing for [10] , was studied properly in which all the 20 malware captures were pre-processed along with the 21 features (columns). The accuracy achieved was not good as compared to different papers studied in related works. The data-pre-processing for [20], 4 malware captures were considered for data- pre-processing along with 14 features (21- 14 columns drop down). So concatenated the ideas of both and worked for data pre-processing for

all 20 malware captures with 14 features. The table 5, below shows the 14 features selected for the data pre-processing for the dataset.

**Table 5 : Data pre-processing for 14 features for IoT-23 dataset**

| Dataset | 14 selected features |
|---|---|
| IoT 23 | 'id.orig_h','id.orig_p','id.resp_h','id.resp_p','proto','duration','orig_bytes', 'resp_bytes','missed_bytes','orig_pkts','orig_ip_bytes','resp_pkts', 'resp_ip_bytes', 'label' |

Initially, the raw dataset was split into individual CSV files based on the capture duration of each traffic trace. A total of 20 CSV files were obtained and concatenated into a single dataframe using the pandas library in Python [4]. The following columns were dropped from the dataframe: "ts", "uid", "service", "conn_state", "local_resp", "local_orig", and "history". These columns contain redundant or irrelevant information for the analysis. The remaining columns were retained for further analysis.

To facilitate label encoding, the "label" column was cleaned and standardized. Specifically, labels containing the same malicious category but with different descriptions or formatting were mapped to a common label. The following labels were consolidated as shown below in table. The remaining labels were assigned the label "Benign". The table 6, below shows the original labels of the dataset.

**Table 6 : Original labels of IoT-23 Dataset**

| Dataset | Original Labels of Dataset |
|---|---|
| IoT-23 | "PartOfAHorizontalPortScan", "Okiru", "DDoS", "C&C", "Attack", "C&C-HeartBeat", "C&C-FileDownload", "C&C-Torii", "C&CHeartBeat-FileDownload", "FileDownload", and "C&C-Mirai" |

Additionally, columns containing missing values were replaced with zeros. The data types of specific columns were also converted to improve the efficiency of the analysis. Columns containing integer values, such as "id.orig_p", "id.resp_p", "orig_bytes", "resp_bytes", "missed_bytes", "orig_pkts", "orig_ip_bytes", "resp_pkts", and "resp_ip_bytes", were converted to 32-bit integer types. The "duration" column was converted to a 32-bit float type. Finally, categorical variables were label-encoded using the scikit-learn library in Python.] The columns "id.orig_h", "id.resp_h", and "proto" were label-encoded to obtain numerical representations of categorical variables.

**Table 7 : Labels and Counts for the iot23_combined_14_features**

| Labels | Counts |
|---|---|
| PartOfAHorizontalPortScan | 825939 |
| Okiru | 262691 |
| Benign | 197809 |
| DDoS | 138777 |
| C&C | 15100 |
| Attack | 39153 |
| C&C- HeartBeat | 349 |
| C&C- FileDownload | 43 |

| | |
|---|---|
| C&C - Torii | 30 |
| FileDownload | 13 |
| C&C-HeartBeat-FileDownload | 8 |
| C&C - Mirai | 1 |

The iot23_combined_14_features.csv file contains a total number of 1048574 records having size of the file 89001 KB. The Table 7, shows the labels and counts for the .csv file. There are total of 10 labels as shown above.

# 4 Design Specification

The dataset selection was the foremost criteria for the research. Based on the related/previous work. It was then decided to further work on IoT-23 dataset. The Data-pre-processing for 14 features for all 20 logs, Binary Classification and the Random Forest Algorithm was designed and implemented. This is discussed more in details in the Section 3 , 3.2 & 5.

# 5 Implementation

- The experiments in this study were implemented using python programming language and various libraries. The data pre-processing and data transformation steps were implemented using pandas python library. The 20 malware captures were read separately and were concatenated into one single dataframe and the data frame was exported to csv for further use for all the ML/Dl models. The dataset was splitted in a 80:20 ratio.
- For the model which required normalization such as ANN and NB we used the Min-Max scaler from scikit-learn python library.
- For Binary class and Multiclass classification the same models were used 1 deep learning and 4 machine learning models. (ANN ,SVM, Decision Tree, Random Forest, Naive Bayes) Machine learning models were built using the scikit-learn library. And the hyper parameters for random forest were tuned using RandomizedSearchCV.
- For Binary classification all the malware attacks were labelled as malware and benign as benign. For the implementation of the ANN model, we utilized the Keras library in Python to create a neural network that consists of multiple hidden layers.
- The neural network's architecture includes an input layer with 13 nodes, which corresponds (13 input features+1 output=14) and available in the pre-processed dataset.
- The model also includes four hidden layers, each with a ReLU activation function and a varying number of nodes: 2000, 1500, 800, and 400.To prevent overfitting, we incorporated four dropout layers, each with a dropout rate of 0.2 after every hidden layer.
- The ANN model's output layer consists of two nodes, corresponding to the two classes ("Benign" and "Malware") in the dataset, and a SoftMax activation function. We compiled the model using the categorical cross-entropy loss function, the Adam optimizer, and the accuracy metric. The model was trained on the pre-processed training data for 10 epochs.

- The performance of each model was evaluated using various metrics, including accuracy, precision, recall, and F1-score.
- All experiments were conducted on a computer with an Ryzen 7 processor and 16GB of RAM. The programming environment used was Anaconda, which provides a convenient platform for data science and machine learning tasks.

# 6 Evaluation

## 6.1 Hardware and the Experimental Setup :

The experiment was conducted on the personal system where the experimental setup was made to further conduct the implementation.

- Hardware Specification: Processor of AMD Ryzen 7 @5700U with Radeon Graphics @1.80 GHz , 16GB of RAM.
- Windows Specification : Windows 10, 22H2v.
- Experimental Setup : Windows 10 , Anaconda 3.1, Jupyter Notebook v6.4.12, with Python 3.9.13 and the TensorFlow of the 2.4 environments.

## 6.2 Evaluation of the Metrics :

In order to evaluate the ML/DL model results, few metrics are used which are discussed as below :

1. Time : It is the most important aspect for the evaluation of the metrics. It is basically the executable the time taken by any of the algorithms in machine learning and the deep learning models

2. Precision : It is basically a metric which evaluates all the ML/DL models. It further calculates the fraction of the exact correctly identified positives. It is calculated as shown below :

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

3. Recall : It is basically a metric which is a measure of positives for actual numbers which are corrected identified. It is calculated as shown below :

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

4. True Positive : It is the outcome, where the models actually predicts correctly as the positive class.

5. False Positive : It is the outcome , where the model actually predicts incorrectly as the positive class.

6. Support Score : It is basically a measuring metrics which scikit-learn of the python library. It indicates the no. of the occurrences of each and every label when the condition is true.

7. F1 Score : Considering both the false positives and the false negatives , the f1 score ids one of the metrics which calculates harmonic mean of the recall and the precision. It considered as a better measure. It is calculated as shown as :

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

## 6.3 Test results for the ML and the DL models :

The rest results achieved for various algorithms are as shown below :

### 6.3.1.1 Binary Classification :

- Support Vector Machine (SVM) :

The training time taken by SVM is 16892.89 seconds and the testing time is 1840.13 seconds. Thus the classification report below shows the accuracy as 0.902 which is 90%. The table 8 and table 9 , shows the Binary classification report and accuracy for SVM.

**Table 8 : Binary Classification Report for SVM**

| Binary Classification | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Benign | 1.000 | 0.075 | 0.140 | 11473 |
| Malware | 0.866 | 1.000 | 0.928 | 68527 |

**Table 9 : Binary Classification Accuracy for SVM**

| Proposed Model Classification | Accuracy |
|---|---|
| Binary Classification | 0.867 |

- Naïve Bayes (NB) :

The training time taken by NB is 17.99 seconds and the testing time is 18.106 seconds. Thus the classification report below shows the accuracy as 0.903 which is 90.03%. The table 10 and table 11, shows the Binary classification report and accuracy for NB.

**Table 10 : Binary Classification Report for NB.**

| Binary Classification | Precision | Recall | F1-score | Support |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| Benign | 0.999 | 0.293 | 0.453 | 39555 |
| Malware | 0.899 | 1.000 | 0.947 | 249380 |

**Table 11 : Binary Classification Accuracy for NB.**

| Proposed Model Classification | Accuracy |
|---|---|
| Binary Classification | 0.903 |

- Artificial Neural Network (ANN) :

The training time taken by ANN is 6925.82 seconds and the testing time is 177.88 seconds. Thus the classification report below shows the accuracy as 0.919 which is 91.90 %. The table 12 and table 13 , shows the Binary classification report and accuracy for ANN.

**Table 12 : Binary Classification Report for ANN**

| Binary Classification | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Benign | 0.921 | 0.443 | 0.598 | 39555 |
| Malware | 0.918 | 0.994 | 0.955 | 249380 |

**Table 13 : Binary Classification Accuracy for ANN**

| Proposed Model Classification | Accuracy |
|---|---|
| Binary Classification | 0.919 |

- Decision Tree (DT) :

The training time taken by DT is 35.22 seconds and the testing time taken is 0.123 with a score of 0.9999. Thus , the classification report shows the accuracy as 1.000 which is 100%. This can also be assumed that DT achieved this by over fitting. So , Random Forest algorithm was introduced. The table 14 and table 15, shows the Binary classification report and accuracy for DT.

**Table 14 : Binary Classification Report for DT.**

| Binary Classification | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Benign | 1.000 | 1.000 | 1.000 | 39555 |
| Malware | 1.000 | 1.000 | 1.000 | 249380 |

**Table 15 : Binary Classification Accuracy for DT.**

| Proposed Model Classification | Accuracy |
|---|---|
| Binary Classification | 1.000 |

- Random Forest (RF) :

The training time taken by RF is 412.26 seconds and the testing time is 5.34 seconds with a score of 0.995 . Thus the classification report below shows the accuracy as 0.995 which is 99%. The table 16 and table 17, shows the Binary classification report and accuracy for RF.

**Table 16 : Binary Classification Report for RF.**

| Binary Classification | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Benign | 1.000 | 0.962 | 0.981 | 39555 |
| Malware | 0.994 | 1.000 | 0.997 | 249380 |

**Table 17 : Binary Classification Accuracy for RF.**

| Proposed Model Classification | Accuracy |
|---|---|
| Binary Classification | 0.995 |

### 6.3.1.2 Multiclass Classification :

- Support Vector Machine :

The training time taken by SVM is 12974.51 seconds and the testing time is 1136.71 seconds. Thus the classification report below shows the accuracy as 0.805 which is 80.50 %. The table 18, shows the Multiclass classification results and accuracy for SVM.

**Table 18 : Multiclass Classification Results for SVM**

| Metrics | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Accuracy | | | 0.805 | 80000 |
| Macro avg | 0.548 | 0.388 | 0.429 | 80000 |
| Weighted avg | 0.795 | 0.805 | 0.776 | 80000 |

- Naïve Bayes :

The training time taken by NB is 5.60 seconds and the testing time is 7.28 seconds. Thus the classification report below shows the accuracy as 0.797 which is 79.70 %. The table 19, shows the Multiclass classification results and accuracy for NB.

**Table 19 : Multiclass Classification Results for NB**

| Metrics | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Accuracy | | | 0.797 | 288935 |
| Macro avg | 0.672 | 0.597 | 0.510 | 288935 |
| Weighted avg | 0.812 | 0.797 | 0.744 | 288935 |

- Artificial Neural Network :

The training time taken by ANN is 7084.11 seconds and the testing time is 86.98 seconds. Thus the classification report below shows the accuracy as 0.917 which is 91.70 %. The table 20, shows the Multiclass classification results and accuracy for ANN. **Table 20 : Multiclass Classification Results for ANN**

| Metrics | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Accuracy | | | 0.917 | 288935 |
| Macro avg | 0.747 | 0.638 | 0.635 | 288935 |
| Weighted avg | 0.924 | 0.917 | 0.905 | 288935 |

- Decision Tree :

The training time taken by DT is 25.19 seconds and the testing time is 0.196 seconds with a score value of 0.999 ~ 1. Thus the classification report below shows the accuracy as 1.000 which is 100 %. The table 21, shows the Multiclass classification results and accuracy for DT.

**Table 21 : Multiclass Classification Results for DT**

| Metrics | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Accuracy | | | 1.000 | 288935 |
| Macro avg | 0.998 | 0.969 | 0.981 | 288935 |
| Weighted avg | 1.000 | 1.000 | 1.000 | 288935 |

- Random Forest :

The training time taken by RF is 148.64 seconds and the testing time is 3.66 seconds. Thus the classification report below shows the accuracy as 0.986 which is 98.60 %. The table 22, shows the Multiclass classification results and accuracy for RF.

**Table 22 : Multiclass Classification Results for RF**

| Metrics | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Accuracy | | | 0.986 | 288935 |
| Macro avg | 0.865 | 0.720 | 0.745 | 288935 |
| Weighted avg | 0.986 | 0.986 | 0.985 | 288935 |

## 6.4 Result Comparison

Comparing the results for the previous work done for [10] and proposed model.

### 6.4.1 Experimental Results for Multiclass Classification.

| Method | Testing Accuracy | Time Cost (Testing) |
|---|---|---|
| Naïve Bayes | 0.797 | 7.28 seconds |
| Support Vector Machine | 0.805 | 1133.606 seconds |

| | | |
|---|---|---|
| Artificial Neural Network | 0.917 | 86.98 seconds |
| Decision Tree | 0.999 ~1 | 0.196 seconds |
| Random Forest | 0.986 | 3.66 seconds |

### 6.4.2 Comparison with Paper [8]

Result Comparison with paper [10] ,for multiclass classification. Additionally added Random Forest algorithm to check the accuracy for it. Although the paper [10] , have only 4 models excluding the Random Forest. Its ANN and not CNN Technically corrected model.

| Method | Testing Accuracy | Time Cost (Testing ) |
|---|---|---|
| Naïve Bayes | 0.30 | 5.705 seconds |
| Support Vector Machine | 0.69 | 12356.68 seconds |
| Artificial Neural Network | 0.69 | 3894.838 seconds |
| Decision Tree | 0.73 | 3.271 seconds |
| Random Forest (not in paper[8] | 0.73 | 51.00 seconds. |

## 6.5 Discussion

The binary classification and multiclass classification was done for the IoT-23 dataset. In which the accuracy was improved as compared to the accuracy for [10]. The binary and multiclass classification gave excellent results for the malware detection. It is really important to have high configuration system for the algorithms to run properly.

# 7 Conclusion and Future Work

In this study, various machine learning algorithms, including Decision Tree, Random Forest, Support Vector Machine (SVM), Artificial Neural Network (ANN), and Naïve Bayes, were evaluated for their effectiveness in detecting malware in network traffic for both binary and multiclass classification settings. The Random Forest classifier demonstrated the best performance in terms of accuracy, precision, recall, and F1-score for both binary and multiclass classification tasks. The other models also exhibited satisfactory performance, indicating their potential in classifying network traffic. Although the models performed well in the given dataset, there are opportunities for improvement and further research. The following future work can be considered to enhance the performance of malware detection models and make them more robust:

**Feature Engineering:** Exploring additional features or selecting the most relevant features for classification can potentially improve the models' performance. Feature selection techniques, such as Recursive Feature Elimination (RFE) or Principal Component Analysis (PCA), can be tested to determine the most significant features.

**Ensemble Learning:** Combining the predictions from multiple models can improve classification accuracy and robustness. Ensemble learning techniques, such as bagging, boosting, or stacking, can be explored to enhance the overall performance of malware detection.
**Deep Learning Techniques:** Deep learning models, such as Recurrent Neural Networks (RNN), can be explored for detecting malware in network traffic. These models can potentially capture complex patterns in the data and improve classification performance.
**Transfer Learning:** Pre-trained models can be fine-tuned and adapted to the specific domain of network traffic analysis. This approach can potentially leverage the knowledge from other domains to improve malware detection in IoT networks.

In conclusion, the machine learning algorithms evaluated in this study show promising results in detecting malware in network traffic. However, the effectiveness of these models can be further improved by exploring advanced techniques, ensemble learning, and deep learning approaches. By doing so, more robust and reliable malware detection models can be developed to enhance the security of IoT networks.

# References :


[1] "Secure IoT Devices for the Maintenance of Machine Tools | Elsevier Enhanced Reader." https://reader.elsevier.com/reader/sd/pii/S2212827116309878?token=8FD9C46FA9CEB92094E2B004EB0F93FBF363A4A65BA807E50974E0E57F13E07458B5B05826366EDBB71B28C83F4BAD15&originRegion=eu-west-1&originCreation=20230426145250 (accessed Apr. 26, 2023).

[2] H. Garg and M. Dave, "Securing IoT Devices and SecurelyConnecting the Dots Using REST API and Middleware," in *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, Apr. 2019, pp. 1–6. doi: 10.1109/IoTSIU.2019.8777334.

[3] R. O. Ogundokun, J. B. Awotunde, S. Misra, O. C. Abikoye, and O. Folarin, "Application of Machine Learning for Ransomware Detection in IoT Devices," in *Artificial Intelligence for Cyber Security: Methods, Issues and Possible Horizons or Opportunities*, S. Misra and A. Kumar Tyagi, Eds., in Studies in Computational Intelligence. Cham: Springer International Publishing, 2021, pp. 393–420. doi: 10.1007/978-3-030-72236-4_16.

[4] V. Clincy and H. Shahriar, "IoT Malware Analysis," in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, Jul. 2019, pp. 920–921. doi: 10.1109/COMPSAC.2019.00141.

[5] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8182–8201, Oct. 2019, doi: 10.1109/JIOT.2019.2935189.

[6] M. Roopak, G. Yun Tian, and J. Chambers, "Deep Learning Models for Cyber Security in IoT Networks," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan. 2019, pp. 0452–0457. doi: 10.1109/CCWC.2019.8666588.

[7] Ö. Aslan and A. A. Yilmaz, "A New Malware Classification Framework Based on Deep Learning Algorithms," *IEEE Access*, vol. 9, pp. 87936–87951, 2021, doi: 10.1109/ACCESS.2021.3089586.

[8] R. Chaganti, V. Ravi, and T. D. Pham, "Deep learning based cross architecture internet of things malware detection and classification," *Comput. Secur.*, vol. 120, p. 102779, Sep. 2022, doi: 10.1016/j.cose.2022.102779.

[9] A. Kumar and T. J. Lim, "EDIMA: Early Detection of IoT Malware Network Activity Using Machine Learning Techniques," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, Apr. 2019, pp. 289–294. doi: 10.1109/WF-IoT.2019.8767194.

[10] Y. Liang, "Anomaly Detection IoT23." Feb. 21, 2023. Accessed: Feb. 27, 2023. [Online]. Available: https://github.com/yliang725/Anomaly-Detection-IoT23

[11] N.-A. Stoian, "Machine Learning for Anomaly Detection in IoT networks: Malware analysis on the IoT-23 Data set".

[12] N. Abdalgawad, A. Sajun, Y. Kaddoura, I. A. Zualkernan, and F. Aloul, "Generative Deep Learning to Detect Cyberattacks for the IoT-23 Dataset," *IEEE Access*, vol. 10, pp. 6430–6441, 2022, doi: 10.1109/ACCESS.2021.3140015.

[13] D. Nanthiya, P. Keerthika, S. B. Gopal, S. B. Kayalvizhi, T. Raja, and R. S. Priya, "SVM Based DDoS Attack Detection in IoT Using Iot-23 Botnet Dataset," in *2021 Innovations in Power and Advanced Computing Technologies (i-PACT)*, Nov. 2021, pp. 1–7. doi: 10.1109/i-PACT52855.2021.9696569.

[14] V. Dutta, M. Choraś, M. Pawlicki, and R. Kozik, "Detection of Cyberattacks Traces in IoT Data," *JUCS - J. Univers. Comput. Sci.*, vol. 26, no. 11, pp. 1422–1434, Nov. 2020, doi: 10.3897/jucs.2020.075.

[15] S. Strecker, R. Dave, N. Siddiqui, and N. Seliya, "A Modern Analysis of Aging Machine Learning Based IoT Cybersecurity Methods." arXiv, Oct. 14, 2021. doi: 10.48550/arXiv.2110.07832.

[16] F. Hussain, S. G. Abbas, U. U. Fayyaz, G. A. Shah, A. Toqeer, and A. Ali, "Towards a Universal Features Set for IoT Botnet Attacks Detection," in *2020 IEEE 23rd International Multitopic Conference (INMIC)*, Nov. 2020, pp. 1–6. doi: 10.1109/INMIC50486.2020.9318106.

[17] I. Ullah and Q. H. Mahmoud, "An Anomaly Detection Model for IoT Networks based on Flow and Flag Features using a Feed-Forward Neural Network," in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, Jan. 2022, pp. 363–368. doi: 10.1109/CCNC49033.2022.9700597.

[18] "TWISC Research Centers | TWISC - Taiwan Information Security Center." https://www.twisc.org/research-centers/ (accessed May 01, 2023).

[19] "IoT-23 Dataset: A labeled dataset of Malware and Benign IoT Traffic.," *Stratosphere IPS*. https://www.stratosphereips.org/datasets-iot23 (accessed Apr. 24, 2023).

[20] "Delete utils.py · Devdatta03/IOT_Traffic_Classifier@2cfb12a," *GitHub*. https://github.com/Devdatta03/IOT_Traffic_Classifier/commit/2cfb12a7620cd583964deac17333f3f0785c924f (accessed Apr. 28, 2023).