National College of Ireland

# Finding IoT privacy issues through malware Detection using XGBoost machine learning technique

MSc Research Project
MSc. In Cyber Security

Parth Bhardwaj
Student ID: x21169578

School of Computing
National College of Ireland

Supervisor: Dr. Arghir Nicolae Moldovan

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

| | | | |
|---|---|---|---|
| **Student Name:** | Parth Bhardwaj | | |
| **Student ID:** | x21169578 | | |
| **Programme:** | Msc. In Cyber Security | **Year:** | 2022-2023 |
| **Module:** | Msc. Research Project | | |
| **Supervisor:** | Dr. Arghir Nicolae Moldovan | | |
| **Submission Due Date:** | 15/12/2022 | | |
| **Project Title:** | Finding IoT privacy issues through malware Detection using XGBoost machine learning technique | | |
| **Word Count:** | 7428 | **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:** Parth Bhardwaj

**Date:** 15/12/2022

**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): | |

# Finding IoT privacy issues through malware Detection using XGBoost machine learning technique

Parth
Bhardwaj
x21169578

**Abstract**

IoT (Internet of Things) has helped in raising the living conditions of people as it has helped in creating a smart environment where all the devices, some that have large significance in the day-to-day life of people, are connected to a single network. The devices connected to the network can be programmed to work according to the needs of the user. These devices can communicate with each other by exchanging data and this will enable the devices in the network to behave in an intelligent manner thereby increasing the comfort of a person living in such an environment. But these environments are highly vulnerable to cyber-attacks which may result in the entire environment collapsing. One major threat to the IoT environments is malware. Malware must be detected in an IoT network for it to be removed. This problem is addressed in the approach proposed here as a malware detection system in IoT environments is proposed here. The malware will be detected using the machine learning based XGBoost classifier. The classifier will be trained separately by using the data in both the IoT-23 and CICIDS-2017 datasets for performing malware detection. The performance of the trained classifier will be evaluated by computing the accuracy and precision and the trained model will be used for creating a desktop application that is able to detect malware in a network based on the network features provided as input. The results of this approach reveal that the XGBoost classifier is effective in detecting malware from an IoT environment and it will also calculate the accuracy and precision between the datasets which are selected for this XGB model. IoT-23 dataset accuracy is more than the CICIDS2017 and the precision is also more in the IoT23.

## 1 Introduction

In an IoT (Internet of Things) network the devices in the network can share data to a server in the network without the involvement of humans (Chaabouni et al., 2019). IoT has allowed devices to be connected to the internet so that these devices can be controlled from anywhere around the world as data can be sent from and to these devices through the internet. The emergence of computer chips with high processing power along with the advancements in

wireless communication has led to things like cars being part of an IoT network (Shen, Fantacci and Chen, 2020). In line with the number of devices being part of IoT networks increasing it was reported that by 2025, 75 billion devices will be connected to IoT devices worldwide (Fizza et al., 2021). IoT is also being used in different and highly important domains like medicine (YIN et al., 2016), agriculture (Ayaz et al., 2019) etc. The expansion and popularity of IoT has led to concerns being raised about the security of data being transferred through the networks. As many devices are connected to a wireless network and some of these devices being really important to the daily needs of a common person or to the needs of organizations working in domains highly important to society the security in IoT networks is an aspect that needs special attention. So, it is important to ensure that the security of the IoT environment is not compromised.

One of the major threats to internet security over the years has been malware. Hackers can use malware to gain unauthorized access to the computing devices of a person or an organization for obtaining highly sensitive or important data. There are many kinds of malware that work in different ways to help the hacker to gain access to a victim's data. Spyware, Keylogger, Trojan Horse, viruses, reaper, worms, echo Bot etc. are different kinds of malware that are used by hackers (Podder et al., 2020). IoT links have proven to be the weak links in an information technology network resulting in the security of the network being compromised (Downey, 2019). The WannaCry malware attack affected over 600 organizations back in 2017, this included health, financial, banking and government institutions (Saleem, 2019).

The techniques used by hackers are becoming sophisticated and difficult to detect through simple detection techniques, but the use of machine learning has proven to be rather effective in the detection of malware (Podder et al., 2020). So, for detecting malware using machine learning a system is proposed here. The XGBoost model will be trained using the details about IoT network traffic and after training the model will be able to identify the presence of malware on an IoT network based on the details of the IoT network.

After the introduction section the report will consist of a 'Literature Review' section where the literature related to malware detection in using machine learning models in general and in IoT environments are studied for understanding the effectiveness of machine learning models, especially the XGBoost model, in detecting malware. The existing malware detection techniques using machine learning will also give an idea about the effectiveness of machine learning in malware detection.

## 1.2 Novelty

- This project proposes the use of the highly effective, as it is an ensemble model, XgBoost model for malware detection in IoT networks. No existing systems used the XgBoost algorithm for the detection of malware from IoT networks.
- The IoT-23 is one of the latest datasets that contains data which can be used for malware detection and this dataset has been used only in a limited number of existing systems. In this

approach the effectiveness of using this dataset for malware detection from IoT networks was evaluated.

### 1.3 Contributions

- An IoT based malware detection system was successfully developed as the system proposed in the project was able to effectively detect malware from IoT networks.
- The XgBoost model was trained for malware detection using the IoT-23 and CICIDS-2017 datasets, separately. From the accuracy and precision achieved by the XgBoost model it was observed that the XgBoost model was able to detect malware more effectively when trained using the IoT-23 dataset. The results of this approach suggest that the IoT-23 dataset is more effective in malware detection from IoT networks compared to an older dataset.

## 2 Related Work

### 2.1 Malware detection using machine learning.

Machine learning algorithms are found to be highly effective in detecting malware (Podder et al., 2020). Machine learning techniques were used for detecting anomaly-based network intrusions in (García-Teodoro et al., 2009). This approach showcased the effectiveness of machine learning techniques in detecting intrusions. The presence of botnet in network traffic was identified using a supervised machine learning model in (Stevanovic and Pedersen, 2014). This approach shows that the machine learning algorithm can effectively identify abnormal network traffic based on the details of the traffic flow in the network. All these approaches showcase the effectiveness of machine learning models in detecting abnormal network traffic based on the details of the traffic.

Ensemble machine learning models were used for improving the accuracy of detecting botnet from IoT networks (Santha devi D ,2020). Stochastic Gradient Descent Classification (SGDC) and Ada-Boost were used in this approach and the machine learning models were trained and tested using the CTU-13 dataset. The accuracy in detecting Internet Relay Chat attack was improved by the SGDC model and it achieved an accuracy of 98.31%. The Ada-Boost performed better than the SGDC in detecting Spam attacks and DDoS as it achieved an accuracy of 98.73%. It was observed in this approach that a dataset containing samples of network traffic affected by malware can be used for training machine learning models and the machine learning models trained by the details of the network traffic is able to effectively detect malware in networks as both the machine learning models were able to achieve a good accuracy in detecting malware. A limitation of this approach is that the data set contained details about the detection of only DDoS and no other kinds of malware, so the machine learning models are not able to detect any other kinds of malware in this approach.

The traffic in an IoT environment was analyzed for botnet using Random Forest (RF), Support-Vector Machine (SVM) and Logistic Regression (LR) in (Bagui, Wang and Bagui, 2021). The classification was performed for detecting botnet in nine devices. The dataset from

the UCI's machine learning repository was used in this approach and all the machine learning classifiers were highly effective in detecting botnets as all three machine learning classifiers achieved accuracies greater than 99%. The RF algorithm achieved the best accuracy in this approach. This approach also showed the effectiveness of the machine learning algorithms in learning the ability to detect malware based on the network traffic in an IoT environment. The main limitation of this approach is that it does not perform feature selection before training the machine learning models.

Anomalies in IoT networks were identified using machine learning algorithms in (Thamaraiselvi and Anitha Selva Mary, 2020). The Naive Bayes (NB), Decision tree (DT), SVM and RF were used in this approach. The data from the IoT-23 dataset was used in this approach and it was observed that this dataset is highly effective in training machine learning models for detecting anomalies in IoT networks as the RF algorithm can effectively detect anomalies with accuracy of 99.5%. The accuracy of the machine learning classifier is observed to be a good metric for evaluating the performance of a machine learning classifier.

Supervised machine learning algorithms like K-nearest neighbor (KNN), SVM, LR, DT , RF and Multi-layer perceptron (MLP) were used for detecting anomalies from IoT network traffic in (Tyagi and Kumar, 2021). The machine learning classifiers were able to classify the attacks like Reconnaissance, DoS, Information Theft and DDoS. A dataset collected from the UNSW-Canberra cyber data repository was used here and metrics like accuracy and precision were determined for evaluating the performances of the different machine learning models. Both the DT and RF classifiers achieve an accuracy of 99.9% in detecting anomalies. But the evaluation of other parameters has shown that RF has the best performance in detection. In this approach it is shown that metrics like accuracy and precision can be used for performance evaluation of the machine learning classifiers. It has been shown in this approach that feature selection will result in the training time of the machine learning models decreasing without any decrease in the performance of the machine learning models.

The performances of several machine learning algorithms in detecting anomalies from IoT networks are compared in (Hasan et al., 2019). SVM, RF, DT , Artificial Neural Network (ANN) and Logistic Regression (LR) are used in this approach and it has been shown that metrics like precision and accuracy can be used for comparing the performances of the machine learning models. The accuracy achieved by the RF, DT and ANN is identical with a value of 99.4% but this approach used metrics other than accuracy to determine that RF showcases the best performance. The data from the IoT networks was used for training the machine learning models effectively as the NSL-KDD, Distributed Smart Space Orchestration System (DS2OS) and real network traffic data was used in this approach. The main limitation of this approach is that it also considers the changes in network traffic caused due to random factors such as anomalies resulting in the reliability of the results of the approach being reduced. The scenario of big data and the working of the approach in real time were also not considered in this approach.

The performances of twelve machine learning models in detecting anomalous behaviours from networks were compared in (Elmrabit et al., 2020). The UNSW-NB15, CICIDS-2017 and Industrial Control System (ICS) datasets are used for training the machine learning models in this approach. It can be observed from this approach that the CICIDS-2017

can be used for effectively training machine learning in detecting network anomalies as it was observed that the RF achieves an accuracy of 99.99% in detecting anomalies, this also the best accuracy value achieved in this approach. So, it can be concluded that the CICIDS-2017 is more effective than the UNSW-NB15 and ICS datasets in training machine learning models for network anomaly detection.

Anomalies in IoT networks were detected using machine learning algorithms in (Liu et al., 2020). The KNN, RF , XGBoost , SVM and LR machine learning classifiers are trained using the IoT Network Intrusion Dataset .Although the best performance was showcased by the KNN with an accuracy of 99%, it can be seen from this approach that the XGBoost classifier is also effective in detecting anomalies in IoT networks.

The effectiveness of machine learning algorithms in detecting anomalies on IoT networks was examined in (Al-Akhras et al., 2020). The KNN, RF and Naïve Bayes algorithms were trained using UNSW-NB15 benchmark dataset. The KNN and RF exhibited the best performances in this approach achieving an accuracy of 100%. This approach also used precision as a metric for performance evaluation.

The RF, DT, MLP, SVM, AdaBoost , Naive Bayes and a variant of ANN was used for detecting anomalies from IoT networks in (Stoian, 2022). The IoT-23 dataset was used in this approach, and it was seen that the best performance was showcased by the RF algorithm as it achieved an accuracy of 99.5%. This approach reveals that the IoT-23 dataset can be used for effectively training machine learning models for detecting anomalies in IoT networks. One limitation associated with the approach is that the data in the dataset was not used for training machine learning models at once as the machine learning, models were trained by breaking up the data and training the models using the smaller samples. Also many of the results of the study, like the high accuracy attained by the DT, were not analyzed more deeply.

A system named Anomaly Detection-IoT (AD-IoT) was used for detecting anomalies from IoT networks in (Alrashdi et al., 2019). The system is based on the Random Forest algorithm, and it is trained using the UNSW-NB 15 data set. It is revealed in this approach that the system achieves an accuracy of 99.34% during classification. This approach again proves the effectiveness of machine learning algorithms in detecting anomalies in IoT networks.

An approach for detecting anomalies effectively and swiftly from IoT networks was proposed in (Alsamiri and Alsubhi, 2019). The Bot-IoT was used in this approach for training the QDA, RF, NB, ID3, MLP KNN and Adaboost classifiers. KNN exhibited the best performance in this approach with an accuracy of 99%. The data was pre-processed, and the feature selection was performed before training the machine learning algorithms. The training and pre-processing may have helped the machine learning model to effectively and quickly detect anomalies in IoT networks.

Machine learning algorithms were used for detecting malware based on a single IoT device in (Van Dartel, 2021). The approach was performed on an ESP32-chip that can classify data points present in the IoT-23 dataset. The data was classified as ‘benign’ or ‘malware’ in this approach. The data was used for training the DT and RF classifiers and both achieved an accuracy of 100%. The main limitation of this approach is that the system is not effective if a lot of malware comes from a specific location or IP range.

## 2.2 Malware detection using XGBoost.

The XGBoost classifier along with the decision tree is used for malware detection in (Kumar and S, 2020). The Ember dataset is used in this approach for training the machine learning model. The use of the XGBoost improves the performance of the machine learning model in detecting malware and it is observed to achieve an accuracy of 98.5%. The model is observed to be effective in detecting malware even with limited computational resources and in relatively less time. The feature selection was performed in this approach, and it was observed to not affect the performance of the model.

The XGBoost classifier with Vote based Backward Feature Elimination technique (XGB-VBFE) was used for detecting malware in (Munisamy Eswara Narayanan and Balasundaram Muthukumar, 2021). Feature selection is performed in this approach and is observed to improve the performance of the model. The model can effectively detect malware as it is able to achieve an accuracy of 99.5%. The model is observed to need only a less amount of time for performing computation in this approach too.

Randomized tree algorithm and XGBoost was used for detecting malware in (Palša et al., 2022). A dataset that combined samples from several other datasets was used for training the model. Both the static and dynamic analysis of malware was combined in this approach. The model was observed to achieve an accuracy of 91.92% while performing static analysis.

## 2.3 Summary

On studying the existing literature related to malware detection and IoT malware detection the machine learning classifiers are highly effective in detecting malware in general and malware in IoT environments. The feature selection and pre-processing of the data before training the machine learning models is observed to improve the performance of the model in detecting malware in IoT environments. It is observed from the literature that the datasets that contain the data about the network traffic can be used for effectively training the machine learning models. The approaches that used the datasets IoT-23 and CICIDS-2017 are observed to effectively train machine learning models in detecting malware based on the data from networks and IoT networks. The XGBoost algorithm is observed to require only a small amount of time for training and is also observed to require only a limited amount resources for performing computation. It was also observed to effectively detect malware.

The XGBoost algorithm was observed to be a machine learning classifier that had several advantages, but it had been used only in a limited number of approaches. No existing literature studied here was observed to use the classifier in detecting malware in IoT networks. So, for studying the capabilities of the XGBoost algorithm it will be used for the system proposed here in detecting malware from an IoT environment. The IoT-23 and CICIDS-2017, which were found to be highly effective in training the machine learning models for anomaly detection in

networks will be used for training the XGBoost machine learning classifier. The data in the datasets will be pre-processed and important features will be selected before training the XGBoost model.

# 3 Research Methodology

## 3.1 Research Questions.

Over the course of this research paper, we will be analyzing and answering the following research question:
Research Question
How effective is XGBoost model for detecting malware from IoT networks?

In order to address the research question, the data from the IoT-23 and CICIDS-2017 datasets will be pre-processed but the feature selection will be performed only on the CICIDS-2017 dataset. The XGBoost algorithm will be trained using the data from both the IoT-23 and CICIDS-2017 dataset separately and the performance of the XGBoost algorithm in detecting malware from IoT environments will be compared based on the two datasets. The performance of the XGBoost model will be evaluated by determining the accuracy and precision achieved by the model when it is trained using the two different datasets.

## 3.2 Aims

- Create a dataset that contains the details of IoT networks for training the XgBoost model.
- Train the XgBoost model using the data in the dataset.
- Evaluate the performance of the XgBoost model in detecting malware from IoT networks.
- Create a desktop application that can detect malware from IoT networks based on the network details given as input.

## 3.3 Sources of Data

The selection of the papers for this research was correlated to the Machine learning techniques, Malware detection, cyber security, IoT devices, Datasets, privacy issues in IoT which be use to of answer the research question. Following lists are used for referencing:

| No. | Libraries for References |
|---|---|
| 1 | IEEE Xplore Digital Library |
| 2 | Science Direct |
| 3 | MDPI |
| 4 | Research Gate |
| 5 | Springer Open |
| 6 | Google Scholar |
| 7 | ACM Digital Library |
| 8 | Kaggle Notebook |
| 9 | Elsevier |

**Table 1 : Library for References**

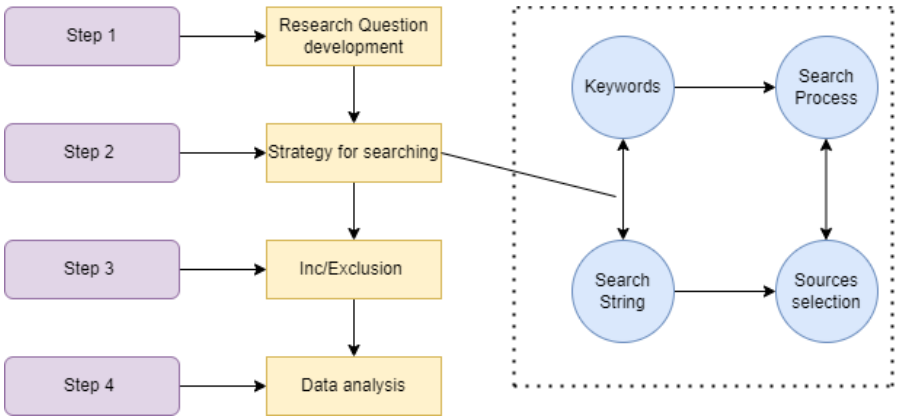


**Figure 1 : Search Proccess**

## 3.4 Datasets.

### 3.4.1 IoT-23

The data in the dataset is made up of the network traffic obtained from IoT devices (Sebastian Garcia, Agustin Parmisano and Maria Jose Erquiaga, 2020). The data is obtained from devices like smart door lock, Amazon Echo home intelligent personal assistant and Philips HUE smart LED lamp. The dataset contains 8 files containing the data of IoT networks that contain

malware and 3 files containing datasets that are benign or normal. The dataset is used because it contains network data and is highly effective in training machine learning algorithms for identifying anomalous networks (Van Dartel, 2021; Stoian, 2022; Thamaraiselvi and Anitha Selva Mary, 2020).

### 3.4.2 CICIDS-2017

The CICIDS-2017 dataset is made up of network data that contains data about network traffic. The dataset contains data of networks which are benign and data of networks in which an anomaly is present. Four files present in this dataset will be combined in this approach. Each of these four files will contain data of networks which are benign and data of networks in which an anomaly is present. Each of the 4 files contains data corresponding to 4 different kinds of anomaly. This dataset is used because the machine learning algorithms trained using this data set exhibits a good performance in detecting networks containing anomalies (Elmrabit et al., 2020).

## 3.5 Pre-processing the data.

The pre-processing of the data in the dataset will help in removing the unwanted data in the dataset. This will result in the XGBoost model being trained more effectively and the time taken for training the model will also be reduced.

### 3.5.1 Pre-processing the data in the IoT-23 dataset.

The data in the dataset needs to be converted into the form of a .csv file so that the XGBoost model can be trained. The unwanted data in the dataset also needs to be removed. The unwanted labels are removed from the dataset. The .csv file is created, and a column named ‘label’ is added in the .csv file. This column contains data that specifies if the network contains malware or if it’s benign. Each row of the .csv file corresponds to the data associated with the different networks and the columns will represent the features of the networks. The column ‘label’ will contain sting values as it specifies if a network contains an anomaly or if it’s benign. The machine learning model cannot be effectively trained using numerical data as it works like a machine and will be more effective when working with numerical data. So, the string values in the column ‘label’ are converted to numerical values. The rows having the value ‘benign’ as the value in the column ‘Label’ will be replaced with 0 and all the rows having values that represent names of malwares in the column ‘Label’, ie; the rows that represent the data of a network containing a malware is replaced with the value 1. Now the columns containing empty values are removed and the null values present in the columns and rows are also identified and removed. Some of the network features will have values as text or string this also has to be converted into numerical values. The columns that contain the values as strings are ‘proto’ and ‘conn_state’. The string values in the columns ‘proto’ are replaced with numerical values ranging from 0 to 2 for different values and the string values in ‘conn_state’ are replaced using

numerical values ranging from 0 to 12. The data will then be saved in a .csv file. The features or data and the labels in the dataset are separated.

### 3.5.2 Pre-processing the data in the CICIDS-2017 dataset

Only four of the files from the dataset are read and these files are combined for creating a .csv file for training the XGBoost model. The 4 .csv files are read and then combined to form a single .csv file. A number of columns in this combined dataset contains values that are not in numerical form, so the values of these columns are converted into numerical form and then the unwanted columns in the dataset are removed. The dataset in the .csv form here must also contain a column named 'Label' that represents the label or the class corresponding to the data that represents the features of a network. So before storing the data in a single .csv file the data that must be stored in the column 'label' must be replaced with numerical values as the column contains data as string values. If the value in the column 'Label' corresponding to a row is 'Benign' it is replaced with 0 and if the value in the column 'Label' corresponding to a row is anything other than 'Benign', ie; the name of malwares or anomalies, it is replaced with 1. The infinite values in the dataset are also replaced with zeroes.

### 3.5.3 Feature selection

The feature selection will only be performed on the CICIDS-2017 dataset as the IoT-23 contains only a small number of features so performing feature selection on the dataset will prove to be insignificant. The feature selection will help in reducing the training time taken by the model as the model will now be trained using the most important features in the dataset. Here the feature selection will be performed by using the analysis of variance (ANOVA) method. After feature selection the best 14 features are obtained from the dataset and these features along with the labels are saved in a .csv file.

## 3.6 Model Creation.

After pre-processing and feature selection the data will be used for training the XGBoost model. Before training the XGBoost model 20% of data in both the datasets will be separated into a testing set, and the remaining data will be used for training the model, the data separation will be performed separately for each dataset. After the separation the data in both the datasets must be balanced as both the datasets are imbalanced and training the model using imbalanced data might affect the performance of the model negatively. So, under sampling is performed on both datasets. The number of rows having the label 'Benign' is much greater than the rows having label values that correspond to malwares in the CICIDS-2017 dataset, so the number of rows having the label 'Benign' is under sampled and made equal to the number of rows having label values that correspond to malware. But in the IoT-23 dataset the number of rows having label values that correspond to malware is greater than the number of rows having label values as 'Benign'. So, in this case the number of rows having label values corresponding to malware is reduced. After balancing both the datasets the data in the datasets will be scaled before

training the model. XGBoost is an application of the gradient boosting algorithm, and it is based on the decision tree classifier. The classifier has been used in many approaches as it has been found to be scalable, fast and efficient. The XGBoost is loaded and trained using the data from the dataset assigned for training the model. The XGBoost model is trained to use the data in both the datasets and both instances of the model will be saved.

### 3.7 Implementation, Evaluation and Results.

The performance of the trained XGBoost model is evaluated by determining the precision and accuracy of the model. The trained XGBoost model can be tested to find out the values for accuracy and precision. The values of accuracy and precision are found out for the XGBoost model that has been trained with both the CICIDS-2017 and IoT-23 datasets.

## 4 Design Specification

### 4.1 Design

The system proposed here will be implemented using Python. The platform used for developing the system contained an 16 GB RAM and i7 processor. The system uses several machine learning support libraries in Python and the desktop application is designed using the 'Tkinter' library in Python. The XGBoost classifier was used as the classifier and the ANOVA technique was used for feature selection from the dataset.

The XGBoost model is optimized using the Gradient Boosting framework. It is portable, flexible, and efficient. In boosting the weak classifiers having low accuracies are combined to create a strong classifier which can exhibit a better performance during classification. The weak learner at each step works based on the direction of gradient associated with the loss function in gradient boosting machines (Chen et al., 2019). The XGBoost is a tree structure model. The XGBoost contains a second-order Taylor expansion as it's loss function (Nasiri and Alavi, 2022).

The ANOVA is a statistical method for comparing several means that are independent. The rank features are determined by the ANOVA by computing the ratio of variances between and among groups (Nasiri and Alavi, 2022). This technique works based on variances associated with different groups of data in the dataset (gajawada, 2019).

The architecture of the system defines the important tasks that are completed at each of the system. The training stage involved the pre-processing, feature selection and training of the XGBoost classifier using the important features so that the classifier can detect malware (Figure (1)). After training the system developed here was tested in the prediction stage here, the architecture consisted of the pre-processing of the data, passing the data to the trained XGBoost

classifier and the prediction by the trained XGBoost classifier. The result showed the prediction of the classifier (Figure 2).

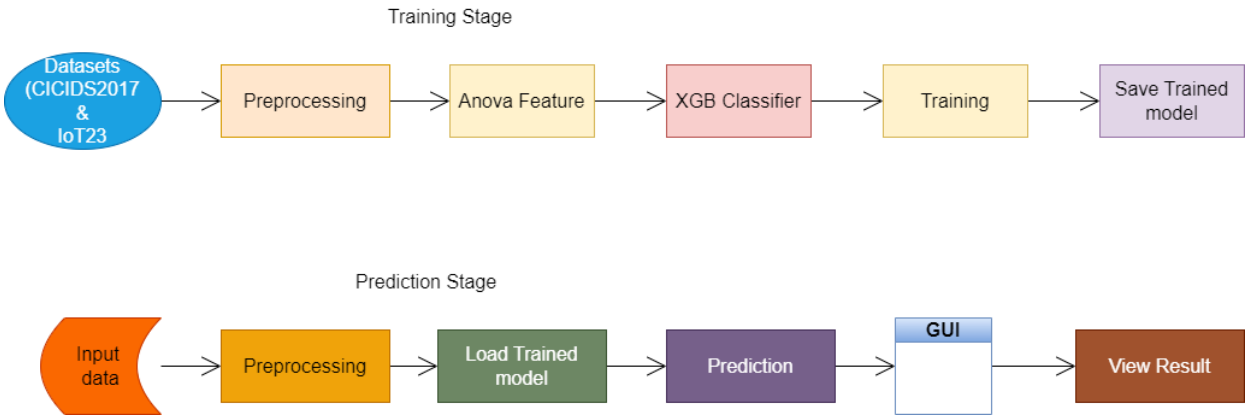


**Figure 2 : The overall architecture of the system**

# 5 Implementation

The pre-processing of the two datasets used in this approach was mainly carried out using the data form the 'pandas' library in Python. The feature selection was performed on the CICIDS-2017 dataset using the method 'SelectKBest' imported from the 'sklearn' library. The is loaded by importing the method 'XGBClassifier()' from the 'sklearn' library. The model is trained using the 'fit ()' method and the trained model will be tested using the 'predict ()' method. The model will be trained separately using both the datasets. Both the trained models will be used for creating a desktop application that detects malware in an IoT environment. The user interfaces of both the desktop applications will consist of input spaces where network features corresponding to the dataset using which the model used in the desktop application has been trained , can be given as input. After entering the network features as input the both the desktop applications perform detection when a button is clicked, and the given inputs are passed on to the trained XGBoost classifier loaded here and the classifier produces an output. Based on the output generated by the classifier the result of detection will be displayed on the interface of the desktop application. If the network features given as input correspond to a network containing malware the result will contain the text 'Malware detected' and if the network does not contain any malware the text 'Normal' will be displayed.

Figure (3) displays the desktop application's ui when malware is found in the data provided as input. Figure (4) displays the desktop application's ui when no malware is discovered in the data provided as input. Figures 3 and 4 use data from the IOT-23 Dataset as input. These two values are important because they represent the appropriate functioning of the desktop app whenever the data from the IOT-23 dataset is provided as input.

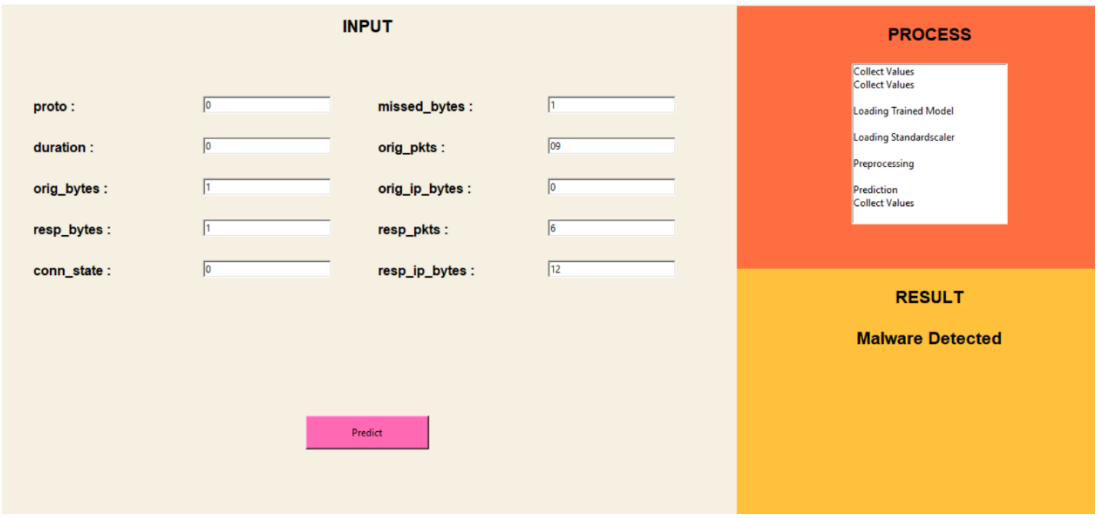


**Figure 3 : GUI for predicting the Malware in IoT-23**

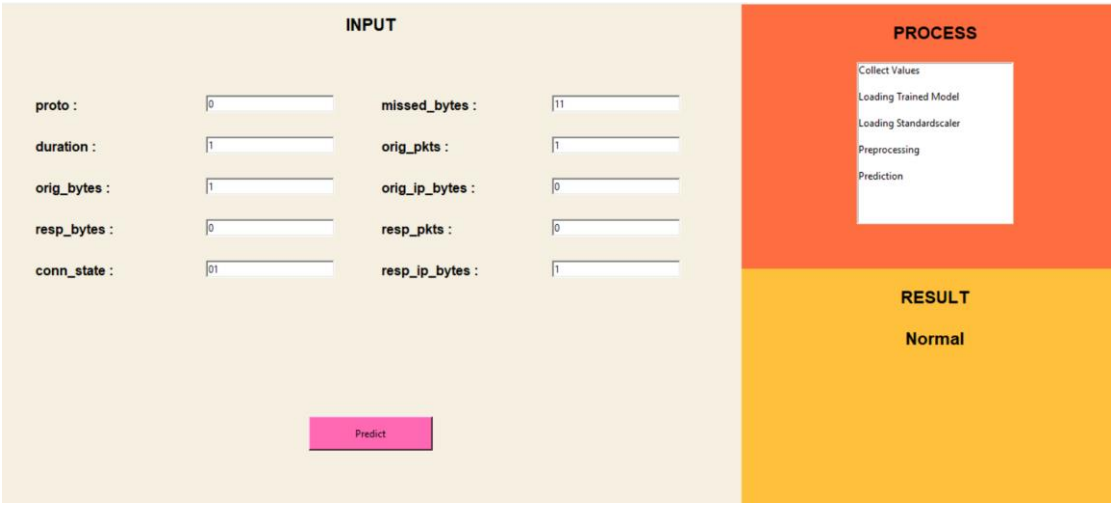


**Figure 4: GUI for predicting Normal in IoT-23**

Figure (5) represents the desktop application's ui when malware is detected in the entered data. Figure (6) represents the desktop application's ui when no malware is detected in the entered data. The data from the CICIDS-2017 Dataset is provided as input for figures 5 and 6. These two findings are significant because they demonstrate that the desktop software works properly when the CICIDS-2017 dataset is being used as input.

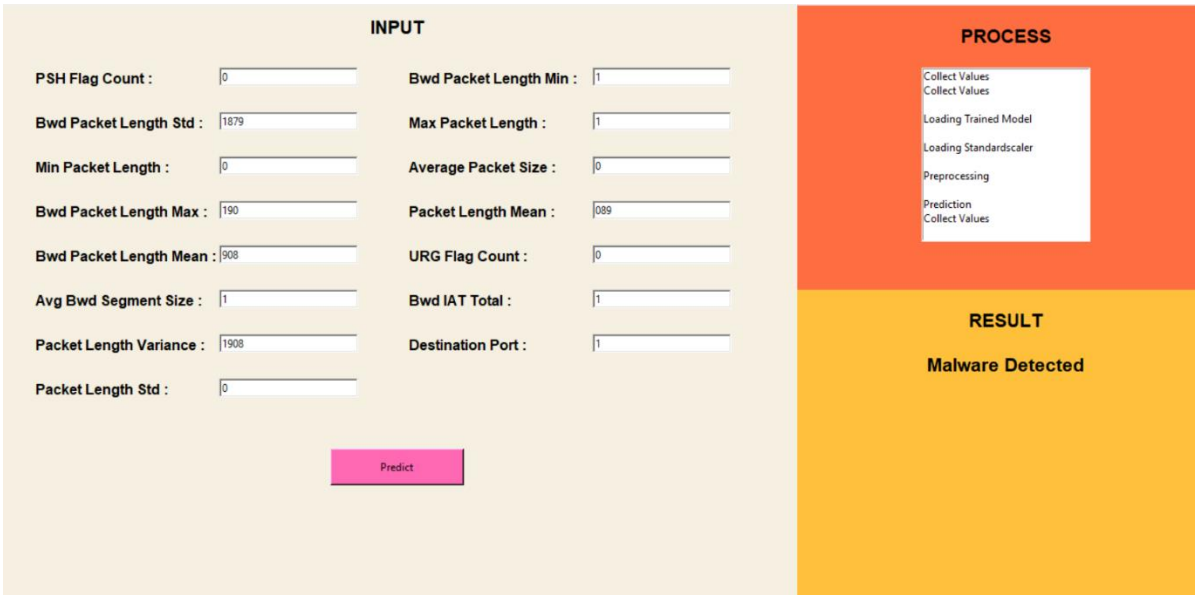


**Figure 5: GUI for predicting Malware in CICIDS-2017**

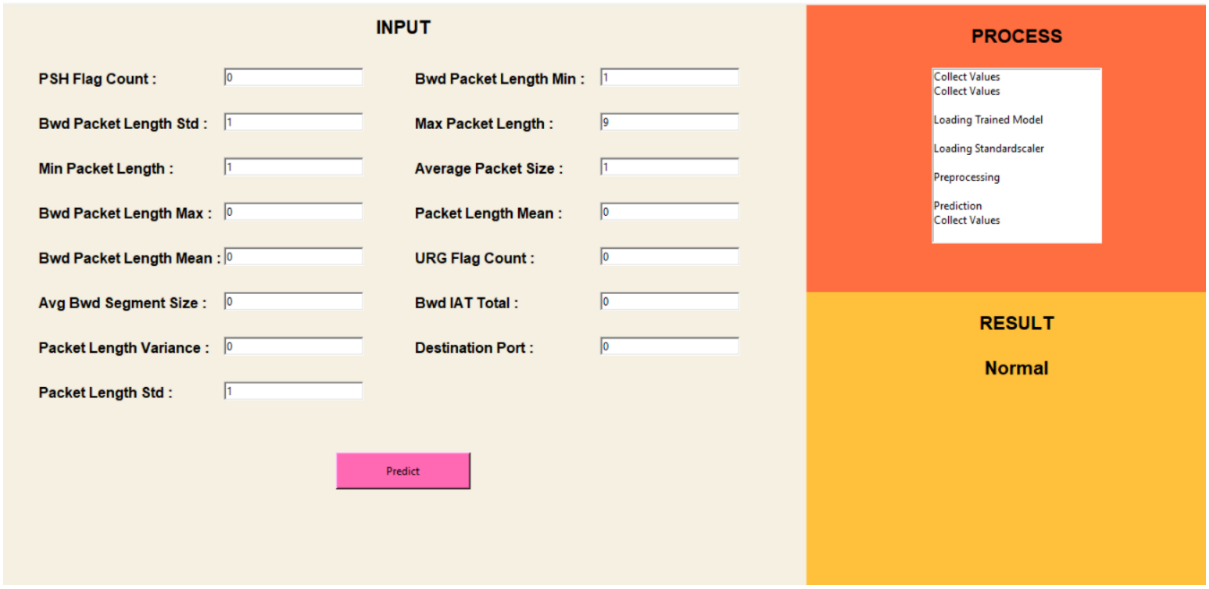


**Figure 6: GUI for predicting Normal in CICIDS-2017**

# 6 Evaluation

The main focus of the proposed solution is that Create a dataset that contains the details of IoT networks for training the XGBoost model, Train the XGBoost model using the data in the dataset & evaluate the performance of the XGBoost model in detecting malware from IoT networks and also create application for desktop to detect malware from IoT networks.

## 6.1 Results and Evaluation

The performance of the XGBoost model is evaluated by computing the accuracy and precision achieved by the model. The accuracy and precision of the model when it is trained using different datasets are found out.

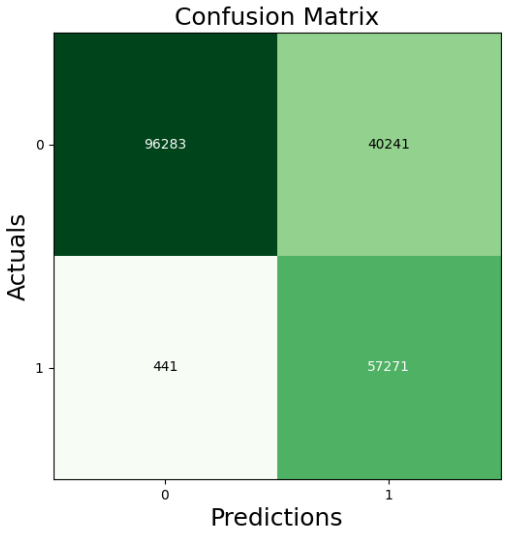


**Figure 7: Confusion matrix of the CICIDS-2017**

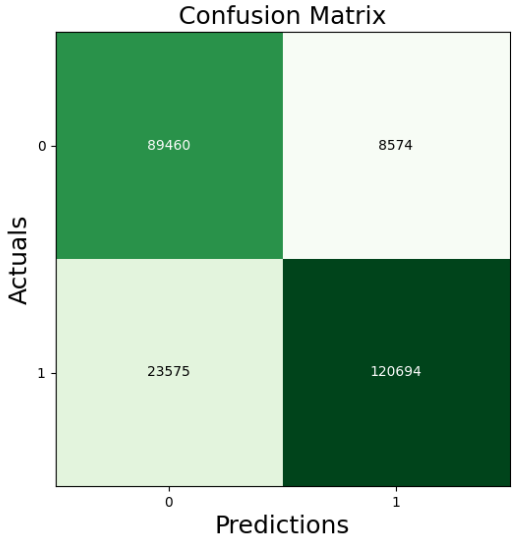


**Figure 8: Confusion Matrix of the IoT-23**

### 6.1.1 Accuracy

The accuracy of the model is found out and it can be observed that the model trained using the IoT-23 dataset has a higher accuracy than the model trained using the CICIDS-2017 dataset (Figure 9).

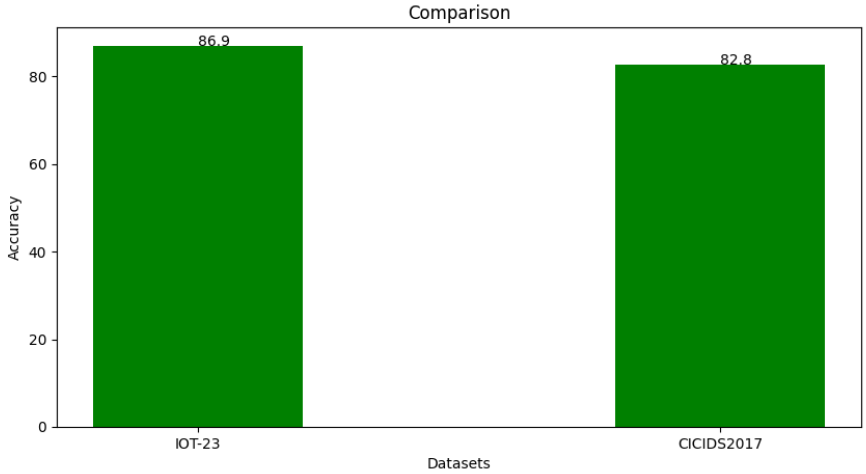


**Figure 9 : The accuracies achieved by the XGBoost model when trained with different datasets.**

### 6.1.2 Precision

The value of precision achieved by the XGBoost model when trained using two different datasets is computed and the model trained using the IoT-23 dataset has a higher precision than the model trained using the CICIDS-2017 dataset (Figure 10).

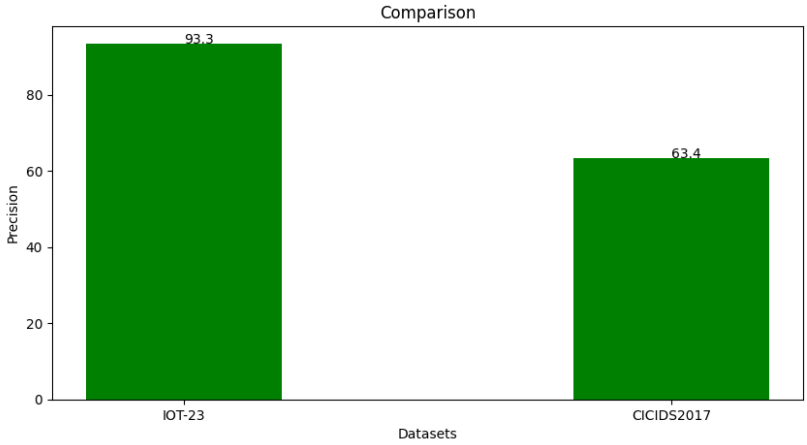


**Figure 10 : The precision achieved by the XGBoost model when trained with different datasets.**

## 6.2 Discussion

The metrics precision and accuracy are used for evaluating the performance of the XGBoost model and it can be observed from the results produced by the XGBoost model that the accuracy and precision achieved by the model trained with the IoT-23 dataset is better than the accuracy and precision achieved by the model trained using the CICIDS-2017 dataset. The smaller size of the IoT-23 dataset may have contributed to the model producing a better

performance than the performance it showcases when trained with the CICIDS-2017 dataset. The machine learning model is observed to be effective in detecting malware from IoT environments which is in line with the observations made during the study of literature. The use of datasets containing features of networks were found to effectively train the machine learning models on studying the existing literature and the XGBoost algorithm was also found to be effective in detecting malware , these observations are in line with the results of the study here as the XGBoost model is able to effectively detect malware in IoT environments but the results of the study here does not produce anything relevant that can support the observations made regarding the fast and efficient nature of the XGBoost model. But the performance of the XGBoost model in detecting malware in the study here is lower than some of the existing studies that performed malware detection using XGBoost algorithm (Table 2). The combination of the datasets used here and the XGBoost classifier may have caused the accuracy of the system developed here to be lower than several existing systems. No existing systems were found to have used the XGBoost classifier along with the IoT-23 and the CICIDS-2017 dataset, so a conclusive reasoning for the lower accuracy achieved by the model cannot be provided from the results of the study performed here. Also, in none of the existing approaches the XGBoost model was used for detecting malware from IoT environments.

| **System** | **Accuracy (in %)** |
|---|---|
| (Kumar and S, 2020) | 98.5% |
| (Munisamy Eswara Narayanan and Balasundaram Muthukumar, 2021) | 99.5% |
| (Palša et al., 2022) | 91.92% |
| System proposed here | 86.9% |

**Table 2: Comparison of the accuracy achieved by the system proposed here to the accuracy achieved by the existing systems that perform malware detection using XGBoost model.**

The system proposed here can effectively detect malware from IoT environments, but it has its own set of limitations. The first major limitation of the system developed here is that the performance of the system developed here is lower than several existing systems. The second limitation is that the size of the datasets used for training the classifier is small as a large portion of the initial large data set was discarded during pre-processing. Another limitation of the system is that it only classifies the data as benign, and malware and the types of malwares are not detected.

# 7 Conclusion and Future Work

The answers to the main research questions of the study performed here are found out. The answer to the research question was found out as it was seen that the machine learning based XGBoost classifier can effectively detect malware in IoT environments. The second research question is also answered as the performance of the XGBoost classifier is evaluated by computing the accuracy and precision. The XGBoost classifier was trained in two sperate ways using two datasets, IoT-23 and CICIDS-2017 and the result of the study shows that the

XGBoost classifier trained using the IoT-23 dataset achieves a higher accuracy and precision, 86.9% and 93.3% respectively, than the XGBoost classifier trained using the CICIDS-2017 dataset which achieves accuracy and precision values of 82.8% and 63.4% respectively. The data in both the datasets is pre-processed, while feature selection is performed using the ANOVA technique on only the CICIDS-2017 dataset. The data is then used for training the classifier and the classifier trained using the two datasets are used for creating two different desktop application that share the same aim of detecting malware from IoT networks based on the network features given as input.

The XGBoost model was shown to have a strong performance in identifying malware from IoT networks, with an accuracy of 86.9% and a precision of 93.3%. Even if the method established here was less accurate than several previous systems, it may still be regarded good.

The accuracy and precision of the XGBoost model, which are performance measures that contribute in evaluating the performance of the trained XGBoost model, were determined to assess its effectiveness in identifying malware from IoT networks.

The system here uses a machine learning model for detecting malware in IoT networks. Deep learning model can be used in the future for detection as it may improve the performance of the system. The system can also be enhanced to detect and classify malware into different types.

# References


Al-Akhras, M., Alawairdhi, M., Alkoudari, A. and Atawneh, S. (2020). Using Machine Learning to Build a Classification Model for IoT Networks to Detect Attack Signatures. International journal of Computer Networks & Communications, 12(6), pp.99–116. doi:10.5121/ijcnc.2020.12607.

Alrashdi, I., Alqazzaz, A., Aloufi, E., Alharthi, R., Zohdy, M. and Ming, H. (2019). AD-IoT: Anomaly Detection of IoT Cyberattacks in Smart City Using Machine Learning. [online] IEEE Xplore. doi:10.1109/CCWC.2019.8666450.

Alsamiri, J. and Alsubhi, K. (2019). Internet of Things Cyber Attacks Detection using Machine Learning. International Journal of Advanced Computer Science and Applications, 10(12). doi:10.14569/ijacsa.2019.0101280.

Ayaz, M., Ammad-uddin, M., Sharif, Z., Mansour, A. and Aggoune, el-Hadi M. (2019). Internet-of-Things (IoT) based Smart Agriculture: Towards Making the Fields Talk. IEEE Access, pp.1–1. doi:10.1109/access.2019.2932609.

Bagui, S., Wang, X. and Bagui, S. (2021). Machine Learning Based Intrusion Detection for IoT Botnet. International Journal of Machine Learning and Computing, 11(6), pp.399–406. doi:10.18178/ijmlc.2021.11.6.1068.

Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C. and Faruki, P. (2019). Network Intrusion Detection for IoT Security Based on Learning Techniques. IEEE Communications Surveys & Tutorials, 21(3), pp.2671–2701. doi:10.1109/comst.2019.2896380.

Chen, M., Liu, Q., Chen, S., Liu, Y., Zhang, C.-H. and Liu, R. (2019). XGBoost-Based Algorithm Interpretation and Application on Post-Fault Transient Stability Status Prediction of Power System. IEEE Access, [online] 7, pp.13149–13158. doi:10.1109/ACCESS.2019.2893448.

devi D, Santha. (2020). IoT Malware Detection using Machine Learning Ensemble Algorithms. Available at:https://www.researchgate.net/publication/342765046_IoT_Malware_Detection_using_Machine_Learning_Ensemble_Algorithms

Downey, A. (2019). Outdated software leaves NHS ‘vulnerable to cyber attack’ new research says. [online] Digital Health. Available at: https://www.digitalhealth.net/2019/04/outdated-software-leaves-nhs-vulnerable-to-cyber-attack-new-research-says/ [Accessed 20 Dec. 2019].

Elmrabit, N., Zhou, F., Li, F. and Zhou, H. (2020). Evaluation of Machine Learning Algorithms for Anomaly Detection. [online] IEEE Xplore. doi:10.1109/CyberSecurity49315.2020.9138871.

Fizza, K., Banerjee, A., Mitra, K., Jayaraman, P.P., Ranjan, R., Patel, P. and Georgakopoulos, D. (2021). QoE in IoT: a vision, survey and future directions. Discover Internet of Things, 1(1). doi:10.1007/s43926-021-00006-7.

gajawada (2019). ANOVA for Feature Selection in Machine Learning. [online] Medium. Available at: https://towardsdatascience.com/anova-for-feature-selection-in-machine-learning-d9305e228476 [Accessed 21 Dec. 2020].

García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G. and Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, [online] 28(1-2), pp.18–28. doi:10.1016/j.cose.2008.08.003.

Hasan, M., Milon Islam, Md., Islam, I. and Hashem, M.M.A. (2019). Attack and Anomaly Detection in IoT Sensors in IoT Sites Using Machine Learning Approaches. Internet of Things, p.100059. doi:10.1016/j.iot.2019.100059.

Kumar, R. and S, G. (2020). Malware classification using XGBoost-Gradient Boosted Decision Tree. Advances in Science, Technology and Engineering Systems Journal, 5(5), pp.536–549. doi:10.25046/aj050566.

Liu, Z., Thapa, N., Shaver, A., Roy, K., Yuan, X. and Khorsandroo, S. (2020). Anomaly Detection on IoT Network Intrusion Using Machine Learning. [online] IEEE Xplore. doi:10.1109/icABCD49160.2020.9183842.

Munisamy Eswara Narayanan, Balasundaram Muthukumar .(2021). Malware Classification Using XGBoost With Vote Based Backward Feature Elimination Technique. Turkish Journal of Computer and Mathematics Education Vol.12 No.10 ,pp. 5915-5923.

Nasiri, H. and Alavi, S.A. (2022). A Novel Framework Based on Deep Learning and ANOVA Feature Selection Method for Diagnosis of COVID-19 Cases from Chest X-Ray Images. Computational Intelligence and Neuroscience, 2022, pp.1–11. doi:10.1155/2022/4694567.

Palša, J., Ádám, N., Hurtuk, J., Chovancová, E., Madoš, B., Chovanec, M. and Kocan, S. (2022). MLMD—A Malware-Detecting Antivirus Tool Based on the XGBoost Machine Learning Algorithm. Applied Sciences, 12(13), p.6672. doi:10.3390/app12136672.

Podder, P., Mondal, M.R.H., Bharati, S. and Paul, P.K. (2020). Review on the Security Threats of Internet of Things. International Journal of Computer Applications, 176(41), pp.37–45. doi:10.5120/ijca2020920548.

Saleem, M. (2019). Brexit Impact on Cyber Security of United Kingdom. 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security). doi:10.1109/cybersecpods.2019.8885271.

Sebastian Garcia, Agustin Parmisano, & Maria Jose Erquiaga. (2020). IoT-23: A labeled dataset with malicious and benign IoT network traffic (Version 1.0.0) [Data set]. Zenodo. http://doi.org/10.5281/zenodo.4743746

Shen, X., Fantacci, R. and Chen, S. (2020). Internet of Vehicles [Scanning the Issue]. Proceedings of the IEEE, 108(2), pp.242–245. doi:10.1109/jproc.2020.2964107.

Stevanovic, M. and Pedersen, J.M. (2014). An efficient flow-based botnet detection using supervised machine learning. 2014 International Conference on Computing, Networking and Communications (ICNC). doi:10.1109/iccnc.2014.6785439.

Stoian, N.-A. (2022). Machine Learning for Anomaly Detection in IoT networks: Malware analysis on the IoT-23 Data set. [online] Available at: https://essay.utwente.nl/81979/1/Stoian_BA_EEMCS.pdf [Accessed 20 Nov. 2022].

Thamaraiselvi, Dr.R. and Anitha Selva Mary, S. (2020). Attack and Anomaly Detection in IoT Networks using Machine Learning. International Journal of Computer Science and Mobile Computing, 9(10), pp.95–103. doi:10.47760/ijcsmc.2020.v09i10.012.

Tyagi, H. and Kumar, R. (2021). Attack and Anomaly Detection in IoT Networks Using Supervised Machine Learning Approaches. Revue d'Intelligence Artificielle, 35(1), pp.11–21. doi:10.18280/ria.350102.

Van Dartel, B. (2021). Malware detection in IoT devices using Machine Learning. [online] Available at: https://essay.utwente.nl/86854/1/VanDartel_BA_EEMCS.pdf [Accessed 20 Nov. 2022].

YIN, Y., Zeng, Y., Chen, X. and Fan, Y. (2016). The internet of things in healthcare: An overview. Journal of Industrial Information Integration, 1, pp.3–13. doi:10.1016/j.jii.2016.03.004.

Dwivedi, R. (2021) *Complete tutorial on Tkinter to deploy machine learning model*, *Analytics India Magazine*. Available at: https://analyticsindiamag.com/complete-tutorial-on-tkinter-to-deploy-machine-learning-model/ (Accessed: December 15, 2022).

Erenkervan (2020) *XGBoost classification*, *Kaggle*. Kaggle. Available at: https://www.kaggle.com/code/erenkervan/xgboost-classification.

*XGBoost algorithm: XGBoost in machine learning*, *Analytics Vidhya*. Available at: https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/#h2_8.

Madhukar, B. (2021) *Using near-miss algorithm for imbalanced datasets, Analytics India Magazine*. Available at: https://analyticsindiamag.com/using-near-miss-algorithm-for-imbalanced-datasets/.