Analysis of IOT-23 Datasets and Machine Learning Models for Malicious Traffic Detection

ISSM-581: Research Methods III
Winter 2021

Chibueze Victor Oha (coha@student.concordia.ab.ca)
Fathima Shakoora Farouk (ffarouk@student.concordia.ab.ca)
Pujan Pankaj Patel (pppatel@student.concordia.ab.ca)
Prithvi Meka (pmeka@student.concordia.ab.ca)
Sowmya Nekkanti (snekkant@student.concordia.ab.ca)
Bhageerath Nayini (bnayini@student.concordia.ab.ca)
Smit Xavier Carvalho (scarvalh@student.concordia.ab.ca)
Nisarg Desai (ndesai@student.concordia.ab.ca)
Manishkumar Patel (mpatel4@student.concordia.ab.ca)

Research Project

Submitted to the Faculty of Graduate Studies
Concordia University of Edmonton

In Partial Fulfilment of the
Requirements of ISSM-581 course

Concordia University of Edmonton
FACULTY OF GRADUTE STUDIES
Edmonton, Alberta

Advisor: Dr Sergey Butakov (sergey.butakov@concordia.ab.ca)
Department of Information Systems Security Management
Concordia University of Edmonton,
Edmonton T5B 4E4, Alberta, Canada

Table of Contents

List of Tables

List of Figures

Analysis of IOT-23 Datasets and Machine Learning Models for Malicious Traffic Detection

Chibueze Victor Oha (coha@student.concordia.ab.ca)
Fathima Shakoora Farouk (ffarouk@student.concordia.ab.ca)
Pujan Pankaj Patel (pppatel@student.concordia.ab.ca)
Prithvi Meka (pmeka@student.concordia.ab.ca)
Sowmya Nekkanti (snekkant@student.concordia.ab.ca)
Bhageerath Nayini (bnayini@student.concordia.ab.ca)
Smit Xavier Carvalho (scarvalh@student.concordia.ab.ca)
Nisarg Desai (ndesai@student.concordia.ab.ca)
Manishkumar Patel (mpatel4@student.concordia.ab.ca)
Adviser: Sergey Butakov (sergey.butakov@concordia.ab.ca)

Abstract— **Connected devices are penetrating the market with an unprecedented speed. Networks that carry Internet of Things (IoT) traffic need highly adaptable tools for traffic analysis in order to detect and suppress malicious agents. This has prompted researchers to explore the various benefits machine learning has to offer. By developing models to detect certain kinds of malicious traffic accurately, it will allow for better detection capabilities if implemented in an Intrusion Detection System (IDS) or Next-generation Firewall. This research paper focuses on harnessing the advanced features of Machine Learning (ML) in exploring the network traffic generated by IoT devices. The IoT-23 dataset was used and preprocessed into three different datasets for further exploration using machine learning algorithms.**

This enhances the easy detection of malicious traffic, thereby improving the security in IoT devices. The machine learning algorithms implemented in this paper include: Logistic Regression, Decision Tree, Random Forest Classifier, XGBoost and Artificial Neural Network. This research was able to achieve almost 100% accuracy across all the three datasets.

Keywords— Machine Learning, IoT-23, Internet of Things, Supervised Learning, TensorFlow

I. INTRODUCTION

In this recent era, the need for more effective and efficient techniques for detecting malicious connections in network system has arisen more than before. This has prompted researchers to explore the various benefits machine learning has to offer. In 2019, CIRA conducted a survey which revealed that 71% of organizations in Canada experienced at least one form of cyber-attack which led to losses in terms of finances, time, and resources. This costs Canadian organizations an average of $9.25 million in investigation and remediation [1] and clearly shows how important it is to put network security into consideration by investing in machine learning.

Internet of Things (IoT) refers to physical devices that are able to interconnect with other devices in a network and exchange information to perform some functionality for the user. These IoT devices have grown exponentially over the last few years and have uses in healthcare, industry and commercial areas with devices like smartwatches, home assistants and smart TVs. In a research published by Transformation Insights [2], they estimate that the number of active IoT devices were over 7.6 billion by the end of 2019. They further estimate that this figure will grow to 24.1 billion IoT devices by 2030.

A common issue with IoT devices is that many of them are not built with security in mind and thus are susceptible to attacks that can compromise the network, information or other devices. Some common issues with IoT devices are that they are poorly configured and firmware updates and patches are not regularly maintained and shipped out, which leads to security issues with these devices. Some of the possible attacks on IoT devices have been documented as follows:

Physical Attacks: Physical attacks on IoT devices target the hardware of an IoT device and by interfering with the device's hardware components, a malicious actor may gain control of the device. These attacks are carried out while the attacker is in close proximity to the network or IoT devices. Some examples of physical attacks on IoT devices include, Node tampering, Radio frequency (RF) Interference, Node attack, Node jamming, Physical damage and Social engineering attacks.

Software Attacks: Most IoT devices run some kind of software and operating system (OS) to perform functionality for users. An IoT device can become compromised on a software level by conducting phishing attacks, introducing malware, spyware or a backdoor into the device. A compromised IoT device can also be used to run malicious code, cause a buffer overflow or conduct side-channel attacks to extract sensitive data from protected memory space.

Network Attacks: Network attacks refer to the possible attacks that may occur during the transmission of data between devices in a network. If a compromised IoT device is present in a network it could cause attacks such as eavesdropping, man-in-the-middle attacks, Denial of Service (DOS) or Distributed Denial of Service (DDOS) by infecting the IoT device with a botnet malware. There have also been attacks documented on the routing protocol for low-power and Lossy networks (RPL) which include the Sybill Attack, Selective Forwarding, Wormhole attack, Sinkhole Attack, Blackhole Attack and Hello flooding attack. [3]

Approaches in detecting malicious traffic such as one caused by botnets using machine learning have increased exigently over the past few years with several techniques proposed with the sole purpose of enhancing malicious traffic detection using novel machine learning algorithms [4]. Machines learn by experience from models designed by humans to analyze data input to make accurate predictions. Additionally, the advancement of Artificial intelligence (A.I.) and Machine learning techniques have helped researchers and security professionals to implement these techniques into Intrusion Detection Systems (IDS). These IDS devices serve as a means of detecting malicious traffic and have only grown more sophisticated in recent years.

The sections of this research paper are structured as follows: Section II gives an overview of the traditional methods that were used in network traffic analysis and the existing machine learning approaches that can be employed in detecting malicious IoT traffic. Section III provides a detailed description of the IoT-23 dataset used and how it was engineered for further analysis. Section IV discusses the preprocessing stages of the IoT-23 dataset used. Section VI outlines the experimentational result of analyzing the IoT traffic using ML models identified in Section V. Section VII compares the result gotten to other similar research papers, detailing their differences. Section VIII gives a summary of the research, future improvement and practical implementation in industrial environment.

II. Literature Review

A. Traditional Methods for Network Traffic Analysis

The most common network traffic analysis methods involved the use of traffic outlier detection algorithms and the use of Intrusion Detection Systems (IDS).

1) Traffic Outlier Detection Algorithms

An outlier is a data point that is expressively different from other normal remarks [5]. There are multiple algorithms to identify the outliers in the urban network traffic. They are as mentioned below:

i) Flow outlier detection

Flow based outlier detection is used to find anomalies by inspecting the header information carried out by the flow analyzers [6]. The methods used to analyze the network traffic are statistical, machine learning, clustering, frequent pattern mining and agent based [7].

ii) Trajectory outlier detection

The use of trajectory outlier detection is to learn trajectories or their sections which alter noticeably from or are unreliable with the residual set. The trajectory outliers include offline processing and online processing [7].

2) Intrusion Detection Systems

The unauthorized access to an information system can be referred as an intrusion. In short, any kind of threat that can affect the enterprise's information confidentiality, integrity and availability is an intrusion [8]. The software applications or devices were developed to detect the intrusion to the system which is called an intrusion detection system (IDS). The purpose of the IDS is not only to prevent the attack but to identify and report it to the network administrator [9].

There are two learning techniques according to which intrusion detection system behaves. They are as follows:

i) Signature based intrusion detection system.

Signature-based intrusion detection system uses known patterns or a signature of the malicious traffic to identify the attack traffic. The known patterns are stored in a database which includes the collection of the suspicious activities and operations that can exploit the weaknesses of the information systems. In this technique, the pattern of the incoming traffic is compared with the pattern stored in the database to differentiate the attack traffic from the legitimate traffic [8].

Due to its nature of comparing the pattern with the database, it is not possible to detect malicious traffic when the attack patterns are not available in the database. This issue can be overcome by an anomaly-based intrusion detection system. The SNORT tool is a great example of a signature-based intrusion detection [9].

ii) Anomaly intrusion detection system

To develop an anomaly-based intrusion detection system, the baseline for the network traffic needs to be decided. The deviation of network traffic behavior from the baseline is considered an intrusion. The behavior of the attack traffic which is not like the legitimate traffic can be treated as the intrusion [10]. This type of system can be attached to the network-based intrusion detection system (NIDS) as well as host-based intrusion detection system (HIDS).

The fundamental edge of the anomaly-based intrusion detection system is its ability in detecting unknown attacks and can be treated as the best solution against zero-day attacks. It is very difficult for the attackers to discover what is the normal behavior decided by the system [8]. Anomaly based intrusion detection system can be categorized in three types as per the training process as follows:

- Statistical based
- Knowledge-based
- Machine-learning based.

3) Conventional packet inspection

Conventional packet filtering reads only packet header information which is called a stateful packet inspection. This is not a sophisticated way to filter the packets as it does not look into the data part (payload) of the packet [11]. The stateful packet inspection can be done by using firewalls. The disadvantages of stateful packet inspection can be overcome by using deep packet inspection.

4) Deep packet inspection

Deep packet inspection is also called DPI. It is an information extraction or packet inspection method. Deep packet inspection carries out an inspection into the information from header of the packets and the data part of the packet at a particular examination point. It checks for the specified protocol, spam, viruses, intrusions and any other defined malicious factor to deny the packet from passing over the examination point [11]. Deep packet inspection makes decisions about whether a specific packet should be dropped or forwarded to the destination.

Deep packet inspection works by using a function of devices such as firewall to conduct packet filtering on received packets. In real time, it checks the contents of the network packets according to the rules or list of malicious signatures stored in a database assigned to the devices by the service providers or the network administrators [11].

B. *Approaches to Detecting IoT Malicious Traffic Using Machine Learning*

The various approaches to malicious traffic detection can be grouped into three categories which include supervised learning, unsupervised learning, and reinforcement learning.

1) Supervised Learning

In the Supervised Learning model, the algorithm is given a completely labelled dataset that it can use to test the accuracy of training data. The model uses the training dataset to learn and creates its own logic to determine that the right outcome is achieved. The testing data set is then fed to the model which can test the model to see how well it learnt during the training phase. [12].

Supervised learning can be categorized into two:

i) Classification

If performance falls within a group, it is known as a classification. Classification can be either binary or multi-class. Multiclass, multi groups can be expected in this sort of classification model. Binary, unless this is predicted by a Boolean value, i.e., 0 or 1, or whether the value is true or false, as in the multiclass classification form [12] [13].

ii) Regression

When the output variable is a real number, regression is used. Height, body mass index, currency and several more may be examples of actual value. Some day-to-day regression-related problems involve time series estimation, respectively [13]. Some examples of regression are Linear Regression, Random Forest and Support vector machines (SVM).

2) Unsupervised Learning

In unsupervised learning, the computer is trained with knowledge that is neither classified nor numbered. The algorithm then attempts to group the unsorted data by extracting useful features based on similarities, patterns, and differences [14]. Clustering and dimension reduction problems are two subtypes of unsupervised learning problems.

3) Reinforcement Learning

In reinforcement learning, the aim is to continually observe the environment and use the knowledge gained to improve upon the model. The model works towards the final goal this way by trial and error through observation of the surrounding environment[15].

All the ML methods mentioned above require massive datasets. Various research groups attempted to create good IoT datasets in order to provide common ground for researchers to sharpen ML models for malicious traffic detection in IoT networks. For example, the IoT-23 dataset, CGIAR dataset, TLESS dataset, etc. This research looked at one of the most recent traffic datasets generated on IoT devices as it represents wide variety of the traffic, attacks and provides good amount of data suitable for most ML algorithms.

III. IOT TRAFFIC DATASET

A. *Description of IoT Traffic Dataset*

IOT-23 is a recent dataset which comprises of network traffic acquired from internet of things (IoT) devices. It encompasses twenty-three captures (20 malwares and 3 benign traffic) all captured within the year 2018 to 2019 by Avast AIC laboratory in partnership with CTU University in Czech Republic. The IOT-23 dataset provides a large data source of properly labelled real malware and benign IoT traffic for machine learning research purposes. Traffic was generated from three hardware IoT devices namely, Amazon Echo Home device, Philips HUE smart LED lamp and a Somfy smart door lock. The generated traffic consisted of protocols such ad HTTP, SSL, DBS, DHCP, Telnet and IRC. The dataset has a capture of 764,735,276 traffic with 764,308,000 being malicious in nature. Table 1 below shows a summary of the types of labels contained in the IoT-23 data [16].

Labels	**Description**
Attack	It indicates malicious attack packets from infected host to another host.
Benign	It indicates genuine packets
C&C	It identifies that the infected devices had connections to a C&C server.
DDoS	DDoS attacks carried out on infected devices are indicated with this label.
HeartBeat	It indicates packets sent from the suspicious source to keep alive the connection on the infected host by the C&C server.
FileDownload	Files downloaded to the infected devices are indicated with this label.
Mirai	It indicates that data have similarities and features of a Mirai Bonet.
Okiru	It indicates that data have similarities and features of an Okiru Bonet.
PartOfAHorizontalPortScan	It indicates information gathered through horizontal port scan for further attacks.
Torii	It indicates that data have similarities and features of a Torii Bonet.

Table 1: Summary of labels in IOT-23 Dataset

The IOT-23 dataset comprises of conn_log_labelled files each containing 23 columns of data. A detailed description of the 23 columns can be found on the IOT-23 dataset website [16].

B. *Engineering of The IoT Traffic Dataset*

Among 23 sub datasets in IoT-23 dataset, some of them are very small in size i.e., in kilobytes and some are larger in size i.e., more than 2 GB. Also, all 23 datasets in IoT-23 are severely imbalanced. 12 malicious labels and a benign label are identified in all 23 datasets. Among these 12 malicious labels, Malicious PartOfHorizontalPortScan, Malicious C&C okiru and Malicious DDoS labels are huge in number(millions) and rest of the malicious labels are less in number. All these labels are not present in every dataset. The detailed description of each dataset is explained in [17]. So, it has been decided to create one file for one label such that all the data of that label is taken from the datasets containing that label and grouped into the file. So, in total 13 files are created for the 13 labels which contains the data from all the 23 datasets.

To balance the dataset and overcome the issue of underfitting and overfitting on the IoT-23 datasets, 3 datasets were created by taking random amount data from all the 13 files. In this process, randomness in data selection was always maintained.

For creating 'Dataset_1', N random data was selected from each of the 12 malicious files and mixed with the benign entries. It contains all 12 malicious labels and benign labels. In this dataset, benign labels are encoded as 1 and all malicious labels are encoded as 0.

For creating ‘Dataset_2’, N random data was selected from 3 big malicious labels files and mixed with the benign entries in a balanced way. This dataset contains Benign, Malicious DDoS, Malicious PartOfAHorizontalPortScan and Malicious Okiru labels.

For creating 'Dataset_3', the other malicious labels which are small are taken randomly and mixed with the benign label. An oversampling technique was used to make the dataset balanced. This dataset includes 10 different types of labels. The labels taken in this dataset are Benign, Malicious C&C-FileDownload, MaliciousC&C, Malicious C&C-Mirai Malicious FileDownload, Malicious Attack, Malicious C&C-HeartBeat, Malicious Torii, Malicious C&C-PartofAHorizontalPortscan, and Malicious C&C-HeartBeat-FileDownload.

In feeding the engineered dataset to the different machine learning models, the dataset labels were encoded as Benign traffic with the encoding of 1 and various Malicious traffic was encoded from 2 to 12 including 0.

Since the datasets created were in the raw format, it needed to be converted into more understandable and meaningful format. This was achieved by some dataset pre-processing steps along with removing unwanted features.

IV. Dataset Preprocessing

i) Dropping of non-unique and unimportant columns:

In the datasets, columns ‘local_orig’ and ‘local_resp’ are non-unique. So, these columns are dropped along with ‘u_id’ column as it does not have any importance in building a model.

ii) Converting Categorical columns to Numerical datatype:

Columns like ‘id_orig_h’, ‘id_resp_h’, ‘proto’, ‘service’, ‘conn_state’, ‘history’ are categorical, and are therefore converted to numerical datatype using Label Encoder.

iii) Checking for Missing values:

The columns ‘duration’, ‘orig_bytes’ and ‘resp_bytes’ contains missing values. These missing values were replaced by ‘Mean’ of their respective column.

iv) Splitting of data into Train and Test data:

After the above steps were completed, the dataset was split into train and test datasets in the ratio of 7:3.

v) Feature Scaling:

Feature scaling limits the data variables of each column to certain range to be compared on common grounds. Standard Scaler technique was implemented on the datasets to standardize the range of data with zero mean and standard deviation of one.

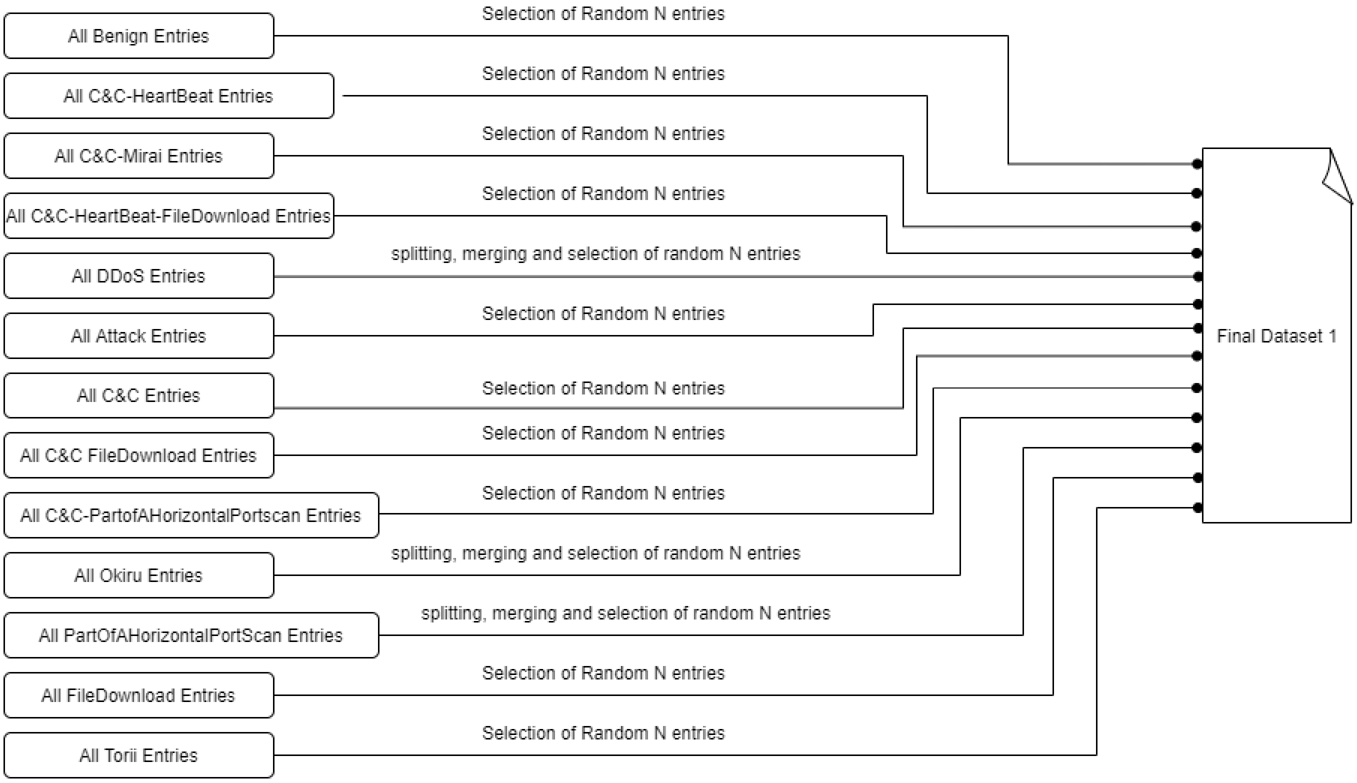
All Benign Entries
Selection of Random N entries
All C&C-HeartBeat Entries
Selection of Random N entries
All C&C-Mirai Entries
Selection of Random N entries
All C&C-HeartBeat-FileDownload Entries
Selection of Random N entries
All DDoS Entries
splitting, merging and selection of random N entries
All Attack Entries
Selection of Random N entries
All C&C Entries
Selection of Random N entries
All C&C FileDownload Entries
Selection of Random N entries
All C&C-PartofAHorizontalPortscan Entries
Selection of Random N entries
All Okiru Entries
splitting, merging and selection of random N entries
All PartOfAHorizontalPortScan Entries
splitting, merging and selection of random N entries
All FileDownload Entries
Selection of Random N entries
All Torii Entries
Selection of Random N entries
Final Dataset 1

Figure 1: Dataset preparation for Dataset_1

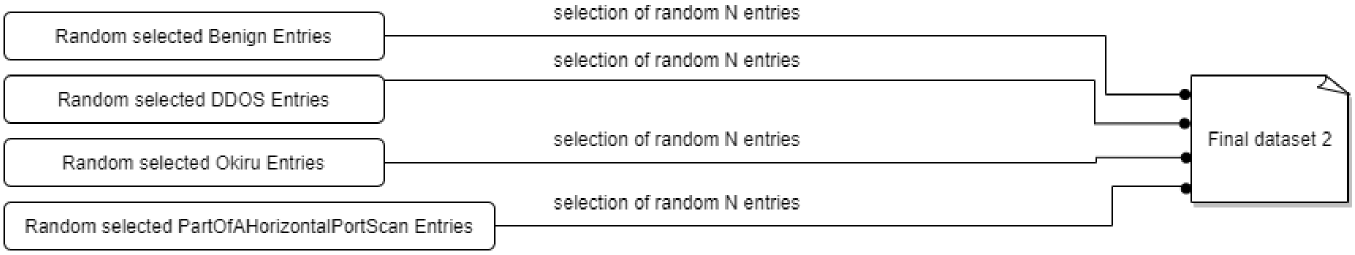
Random selected Benign Entries
selection of random N entries
Random selected DDOS Entries
selection of random N entries
Random selected Okiru Entries
selection of random N entries
Random selected PartOfAHorizontalPortScan Entries
selection of random N entries
Final dataset 2

Figure 2: Dataset preparation for Dataset_2

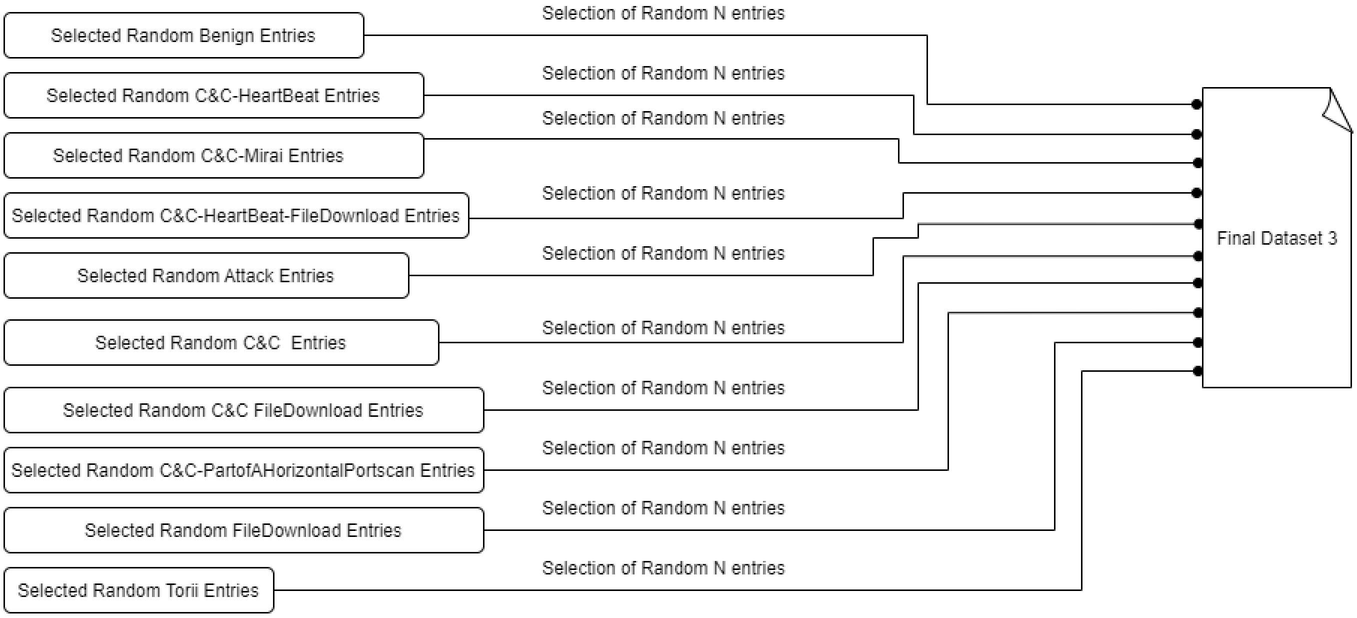
Selected Random Benign Entries
Selection of Random N entries
Selected Random C&C-HeartBeat Entries
Selection of Random N entries
Selected Random C&C-Mirai Entries
Selection of Random N entries
Selected Random C&C-HeartBeat-FileDownload Entries
Selection of Random N entries
Selected Random Attack Entries
Selection of Random N entries
Selected Random C&C Entries
Selection of Random N entries
Selected Random C&C FileDownload Entries
Selection of Random N entries
Selected Random C&C-PartofAHorizontalPortscan Entries
Selection of Random N entries
Selected Random FileDownload Entries
Selection of Random N entries
Selected Random Torii Entries
Selection of Random N entries
Final Dataset 3

Figure 3: Dataset preparation for Dataset_3

V. Machine Learning Models

In analyzing each of the engineered datasets, five machine learning models were used to carry out the data analyses. These includes Logistic Regression, Decision Tree Classifier, Random Forest Classifier, XGBoost Classifier and Artificial Neural Networks

A. *Decision Tree Classifier*

Dataset_1 and Dataset_2 decision tree models were configured with default decision tree configuration (without any parameters). While Dataset_3 decision tree model parameters were configured with criterion to ‘gini’, min_sample_leaf with 1, mean_sample_split with 2, presort with ‘deprecated’ and splitter with ‘best’.

B. *Logistic Regression*

Logistic regression model used in each dataset consist of two hidden layers. In Dataset_1, first hidden layer's weight TensorFlow variable is calculated using shape as (17,12) and biases TensorFlow variable with shape as (12,). Second hidden layer's weight TensorFlow variable is calculated using shape as (12,10) and biases TensorFlow variable with shape as (10,). In Dataset_2, first hidden layer's weight TensorFlow variable is calculated using shape as (17,8) and biases TensorFlow variable with shape as (8,). Second hidden layer's weight TensorFlow variable is calculated using shape as (8,4) and biases TensorFlow variable with shape as (4,). Lastly on Dataset_3, first hidden layer's weight TensorFlow variable was calculated using shape as (17,4) and biases TensorFlow variable with shape as (4,). Second hidden layer's weight TensorFlow variable was calculated using shape as (4,2) and biases TensorFlow variable with shape as (2,). Moreover, all logistic regression model's first hidden layer has 'Relu' as an activation function.

C. *Random Forest Classifier*

Random Forest Classifier model of all three datasets is configured without any parameters.

D. *XGBoost Classifier*

XGBoost classifier model in all three datasets was configured with different random state and learning rate values. For Dataset_1 and Dataset_2, this model’s values were set with random state as 1 and learning rate as 0.01. While for Dataset_3, this model’s values were set to random state as 42 and learning rate as 0.1.

E. *Artificial neural networks*

ANN sequential model for all 3 datasets were configured with 1 input layer, 1 hidden layer and 1 output layer. First hidden layer of all models has (17, null) as input shape, and 'relu' as an activation function. The output layer of all models for Dataset_1, Dataset_2 and Dataset_3 was 10,4 and 2 respectively. All models for three of the different datasets were compiled with optimizer = 'Adam', loss = 'categorical cross entropy' and metrics = 'accuracy'.

VI. Experimental Results

To determine the effectiveness of the machine learning algorithms employed in this research, machine learning metrics such as accuracy, precision, recall, AUC_ROC and F1 score were used. Accuracy gives an overview of how effective the ML algorithms are in detecting malicious and non-malicious traffic. It demonstrates the algorithm’s ability to differentiate false positives and false negative. F1 score helps to determine the precision of the classifier. A higher F1 score indicates a better performance of the analysis. The number of true positives to the total number of real positives is expressed as Recall.

Dataset_1 contains random data selected from each of the 12 malicious files and mixed with benign entries. It contains all 12 malicious labels and benign labels. Results from analysis of Dataset_1 are shown in Table 2.

Algorithm	Evaluation Metrics			
	Accuracy	Precision	Recall	F1 score
Logistic Regression	96.1%	0.93	0.99	0.96
Random Forest Classifier	99.99%	1	1	1
Decision Tree Classifier	99.99%	1	1	1
XGBoost Classifier	99.90%	0.99	0.99	0.99
Artificial Neural Network	99.99%	1	1	1

Table 2: Result Analysis Summary of Dataset I

As shown in Table 2 for Dataset_1, Logistic regression produced the least accuracy of 96.1% and F1 score of 0.96. On the other hand, artificial neural network and random forest classifier produces a much better accuracy of 99.99% and F1 score of 1 because of their ability to evaluate complex inputs.

A Confusion matrix provides a summary of the performance of classification algorithm when evaluating a machine learning classification problem. It is the ratio of accurate prediction to the overall prediction made. The confusion matrix diagram for dataset_1 is shown in Table 3.

Key:

🟥 Logistic Regression
🟦 Decision Tree Classifier
🟨 Random Forest Classifier
🟩 XGBoost Classifier
🟩 Artificial Neural Network

		Predicted Label			
		0		1	
Actual Label	0	92.71%	99.99%	7.29%	0.01%
		99.99%	99.98%	0.01%	0.02%
		99.79%		0.21%	
	1	0.57%	0.01%	99.43%	99.99%
		0.01%	0.17%	99.99%	99.83%
		0.19%		99.81%	

Table 3: Confusion Matrix summary of Dataset_1

Confusion matrixes of Dataset_1 shows that Logistic regression had the least performance due to a higher number of false positives of 0.57% and false negative of 7.29% when compared to other machine learning algorithms. Nevertheless, Decision Tree classifier had the best performance with a lower false positive and false negative of 0.01%.

In Dataset_2, random data was selected from 3 large malicious labels files and mixed with the benign entries in a balanced way. Results from the analysis of Dataset_2 are as shown in Table 4.

Algorithm	Evaluation Metrics			
	Accuracy	Precision	Recall	F1 score
Logistic Regression	98.8%	0.989	0.988	0.988
Radom Forest Classifier	99.99%	1	1	1
Decision Tree Classifier	99.99%	1	1	1
XGBoost Classifier	99.99%	0.99	0.99	0.99
Artificial Neural Network	99.99%	1	1	1

Table 4: Result Analysis Summary of Dataset II

Like the analysis of Dataset_1, logistic regression produced lesser accuracy of 98.9% compared to Random Forest classifier, Decision Tree Classifier and Artificial Neural Network due to high false positives and false negatives when analyzing Dataset_2.

Key:

🟥 Logistic Regression
🟦 Decision Tree Classifier
🟨 Random Forest Classifier
🟩 XGBoost Classifier
🟩 Artificial Neural Network

		Predicted Label							
		0		1		2		3	
Actual Label	0	99.99%	100%	0.03%	0	0.03%	0	0.03%	0
		100%	99.99%	0	0	0	0.01%	0	0
		99.99		0.05%		0		0.05%	
	1	0.22%	0	95.50%	99.99%	0	0.01%	4.28%	0
		0	0	99.99%	100%	0.01%	0	0	0
		0		99.99%		0.01%		0	
	2	0	0	0.02%	0.01	99.98%	99.99%	0	**0**
		0	0	0.01%	0.02%	99.99%	99.98%	0	**0**
		0		0.01%		99.98%		0.01%	
	3	**0**	0	0.02%	0	0	0	99.98%	100%
		0	0	0	0	0	0	100%	100%
		0		0		0		100%	

Table 5: Confusion Matrix Summary of Dataset_2

For the confusion matrixes of Dataset_2, Logistic Regression Analyses had a lower performance with higher false negatives and false positives compared to Decision Tree Classifier and Artificial Neural Network which performed much better with more precise predictions.

Dataset_3 contains the other malicious labels which are small. They are taken randomly and mixed with the benign label. An oversampling technique was used to make the dataset balanced. This dataset includes 10 different types of labels. Results from analysis of Dataset_3 are shown in Table 6.

Algorithm	Evaluation Metrics			
	Accuracy	Precision	Recall	F1 score
Logistic Regression	98.1%	0.982	0.981	0.981
Random Forest Classifier	99.98%	0.99	0.99	0.99
Decision Tree Classifier	99.98%	0.99	0.99	0.99
XGBoost Classifier	99.95%	0.99	0.99	0.99
Artificial Neural Network	99.98%	0.99	0.99	0.99

Table 6: Result Analysis Summary of Dataset III

Logistic Regression

		Predicted Label									
		0	1	2	3	4	5	6	7	8	9
Actual Label	0	99.51%	0	0.04%	0.17%	0	0.08%	0.13%	0	0.04%	0
	1	0	99.06%	0	0	0	0	0	0	0	7.3%
	2	0.23%	0	90.80%	0	0	0.64%	0	0%	7.90%	0
	3	0	0	0	100%	0	0	0	0	0	0
	4	0	0	0	0	100%	0	0	0	0	0
	5	0	0	1.87%	0.04%	0	97.99%	0	0	0.09%	0
	6	0	0	0.04%	0	0	0	99.96%	0	0	0
	7	0	0	0	0	0	0	0	100%	0	0
	8	0	0	0	0	0	0	0	0	100%	0
	9	0	0	0	0	0	0	0	0	0	100%

Decision Tree Classifier

		Predicted Label									
		0	1	2	3	4	5	6	7	8	9
Actual Label	0	100%	0	0	0	0	0	0	0	0	0
	1	0	100%	0	0	0	0	0	0	0	0
	2	0.08%	0.04%	99.88%	0	0	0	0	0	0	0
	3	0	0	0	100%	0	0	0	0	0	0
	4	0	0	0	0	100%	0	0	0	0	0
	5	0	0	0	0	0.04%	99.92%	0	0	0.04%	0
	6	0	0	0	0	0	0	100%	0	0	0
	7	0	0	0	0	0	0	0	100%	0	0
	8	0	0	0	0	0	0	0	0	100%	0
	9	0	0	0	0	0	0	0	0	0	100%

Random Forest Classifier

		Predicted Label									
		0	1	2	3	4	5	6	7	8	9
Actual Label	0	99.92%	0	0.08%	0	0	0	0	0	0	0
	1	0	100%	0	0	0	0	0	0	0	0
	2	0	0	100%	0	0	0	0	0	0	0
	3	0	0	0	100%	0	0	0	0	0	0
	4	0	0	0	0	100%	0	0	0	0	0
	5	0	0	0	0	0.04%	99.96%	0	0	0	0
	6	0	0	0	0	0	0	100%	0	0	0
	7	0	0	0	0	0	0	0	100%	0	0
	8	0	0	0	0	0	0	0	0	100%	0
	9	0	0	0	0	0	0	0	0	0	100%

XGBoost Classifier

		Predicted Label									
		0	1	2	3	4	5	6	7	8	9
Actual Label	0	99.81%	0	0.08%	0	0	0.08%	0.03%	0	0	0
	1	0	100%	0	0	0	0	0	0	0	0
	2	0	0	99.85%	0	0	0.15%	0	0	0	0
	3	0	0	0	100%	0	0	0	0	0	0
	4	0	0	0	0	100%	0	0	0	0	0
	5	0	0	0.04%	0	0.04%	99.84%	0	0.04%	0.04%	0
	6	0	0	0	0	0	0	100%	0	0	0
	7	0	0	0	0	0	0	0	100%	0	0
	8	0	0	0	0	0	0	0	0	100%	0
	9	0	0	0	0	0	0	0	0	0	100%

Artificial Neural Network

		Predicted Label									
		0	1	2	3	4	5	6	7	8	9
Actual Label	0	99.88%	0	0	0.04%	0	0.04%	0	0	0.04%	0
	1	0	100%	0	0	0	0	0	0	0	0
	2	0	0	99.07%	0	0	0.85%	0	0	0.08%	0
	3	0	0	0	100%	0	0	0	0	0	0
	4	0	0	0	0	100%	0	0	0	0	0
	5	0	0	0.07%	0.04%	0	99.89%	0	0	0	0
	6	0	0	0	0	0	0	100%	0	0	0
	7	0	0	0	0	0	0	0	100%	0	0
	8	0	0	0	0	0	0	0	0	100%	0
	9	0	0	0	0	0	0	0	0	0	100%

🟥 Logistic Regression

🟦 Decision Tree Classifier

🟨 Random Forest Classifier

🟩 XGBoost Classifier

🟩 Artificial Neural Network

Table 7: Confusion Matrix Summary of Dataset_3

Confusion matrixes of Dataset_3 shows that Radom forest classifier algorithm had a better performance with the least number of false positives and false negatives when compared to other algorithms used.

VII. Discussion

Researchers in [18] conducted an investigation into the applicability of machine learning algorithms in the identification of anomalies in IoT networks. It compared different machine learning algorithms against the IoT-23 dataset which contains both malicious and benign traffic. The following machine learning algorithms were used in the comparison: Random Forest (RF), Naïve Bayes (NB), Multi-Layer Perceptron, a variant of the Artificial Neural Network (ANN), Support Vector Machine (SVM) and AdaBoost (ADA). As per the results of the research, the Random Forest algorithm returned 100% accuracy and was deemed the best algorithm for this dataset.

Matrices		Classifiers				
		RF	NB	ANN	SVM	ADA
Precision	weighted	1.00	0.76	0.71	0.60	0.86
	Macro	0.88	0.27	0.33	0.23	0.55
Recall	weighted	1.00	0.23	0.66	0.67	0.87
	Macro	0.85	0.38	0.14	0.14	0.35
F1_score	weighted	1.00	0.25	0.52	0.59	0.83
	Macro	0.84	0.10	0.10	0.13	0.37
	Accuracy	1.00	0.23	0.66	0.67	0.87

Table 8: The results from various ML methods from Stoian [18]

Hussain et al. [19] proposed a universal feature set for machine learning models to distinguish botnet traffic from benign traffic regardless of the underlying dataset. With regards to the IoT-23 dataset, the following features were selected as the ‘most universal’ features using logistic regression. They are as follows; Pkt Len Mean, Bwd Pkt Len Min, Pkt Len Min, Pkt Size Avg, Bwd Header Len, Bwd IAT Max, Bwd Pkt Len Mean, Flow Byts/s, Flow IAT Max, Fwd Pkt Len Mean. Four ML algorithms were used to classify the traffic, they are: Naïve Bayes, K nearest neighbor, Random forest and Logistic Regression. Of the four, Random Forest performed the best, but all other algorithms used returned above 98% in accuracy, precision, recall and F1- score. When article [19] is compared with this research, this research performed better with a percentage of 99.9% because of how the IoT-23 dataset was engineered.

In a research conducted by Kumar et al., [20] they proposed and developed EDIMA, a lightweight IoT botnet detection solution deployed at edge gateways which provides early detection of botnets before they can coordinate an attack. EDIMA consists of a two-stage detection mechanism. The first uses Machine Learning techniques to classify aggregate network traffic and the second stage uses ACF based tests to detect individual bots. Three types of ML algorithms were used in this experiment, Gaussian Naive Bayes’, Support Vector Machine and Random Forest. Again, the random forest algorithm performed the best and had the highest accuracy, recall, precision and F1 score out of the three algorithms used.

Practically, the proposed machine learning models can be used for intrusion detection systems for IoT devices. It can be done in two main phases namely detection and classification. Detection phase is used to extract the features and its value from the packets of IoT network. Specifically, it looks at the features and its value which the machine learning model is trained for. It puts all the information of the packets in the form of a record. Secondly, the classification phase can be used to classify the data records obtained in the pre-processed phase. Classification can be done according to the nature of machine learning method used. This phase recognizes that the identified record is benign or malicious.

As IoT devices continue to become more popular in corporate and home environments, there is a growing need to maintain the security of the network these devices are connected to. Specific malware that targets vulnerable IoT devices can create a point of weakness and compromise the security of the network. As a result, designing and implementing such Machine Learning models and integrating them with IDS devices will allow for better detection of malicious IoT traffic.

Although the engineering and preprocessing of the IoT-23 dataset implemented here differ from other research papers compared above, Random Forest classifier also performed better in anomaly detection and classification. As described in above Section III, 3 different datasets were created namely Dataset_1, Dataset_2 and Dataset_3 from the IoT-23 dataset. Moreover, the IoT-23 dataset was too big in size and some of the attack categories were small, which would easily result into overfitting of models. So, dividing this big dataset into three smaller datasets with different categories has presented a chance to train models with all the attack labels.

The advantage of this approach is that it gives an opportunity to train all models in a similar way to produce better performance matrix results of all three datasets with all labels. In addition, less computational power is required in training all three sub datasets as compared to training the full dataset at once. In real time, it can give good detection capability for all kinds of attacks which will make detection more accurate, and systems more secured.

VIII. Conclusion

After deep examination of IoT-23 datasets, it was observed that most of the datasets are imbalanced. A lot of data preprocessing was conducted to prepare three final datasets out of the 23 smaller datasets present in the IoT-23 dataset. Various machine learning models were implemented using these final datasets but only five models, namely Logistic Regression, Decision Tree Classifier, Random Forest Classifier, XG Boost Classifier and Artificial Neural Network are considered for this research based on the time taken to train the models. Among these five models, the time taken to train Decision tree model and Artificial Neural network model is less when compared to the Logistic Regression, Random Forest and XG Boost models. In terms of model evaluation metrics, with the exception of Logistic Regression, the other four model's performance is almost 100 percent across the three datasets.

To improve the models, further research can be done for the security of IoT device which should be conducted in the post-processing phase. An algorithm can be developed which can further classify the data in groups of records for the second time to reduce the false ratio of the proposed research. A possible future research would be to implement these trained ML models in an IDS device or a next generation firewall in an IoT network and monitor how well the models perform in a real-world environment.

Transformed datasets and scripts used in this research can be found at [21].

IX. References

[1] CIRA, "2019 CIRA Cybersecurity Survey," CIRA, 2019. [Online]. Available: https://www.cira.ca/resources/cybersecurity/report/2019-cira-cybersecurity-survey. [Accessed 2 March 2021].

[2] Transforma Insights , "Global IoT Market Will Grow to 24.1 Billion Devices in 2030, Generating $1.5 Trillion Annual Revenue," CISON PR Newswire, 19 May 2020. [Online]. Available: https://www.prnewswire.com/news-releases/global-iot-market-will-grow-to-24-1-billion-devices-in-2030--generating-1-5-trillion-annual-revenue-301061873.html. [Accessed 20 March 2021].

[3] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity* , vol. 4, no. 18, 2021.

[4] E. B. Beigi, H. H. Jazi, N. Stakhanova and A. A. Ghorbani, "Towards effective feature selection in machine learning-based botnet detection approaches," in *2014 IEEE Conference on Communications and Network Security*, San Francisco, CA, USA, doi: 10.1109/CNS.2014.6997492, 2014, pp. 247-255.

[5] L. Jae-Gil, H. Jiawei and . L. Xiaolei, "Trajectory Outlier Detection:A Partition-and-Detect Framework," in *2008 IEEE 24th International Conference on Data Engineering*, Cancun, Mexico, 25 April 2008.

[6] R. Sharma, A. Guleria and R. K. Singla, "An overview of flow-based anomaly detection," *International Journal of Communication Networks and Distributed Systems*, vol. 21, no. DOI: 10.1504/IJCNDS.2018.10014505, pp. 220-240, July 2018.

[7] Y. Djenouri, A. Belhadi, J. C.-W. Lin, D. Djenouri and A. Cano, "A Survey on Urban Traffic Anomalies Detection Algorithms," *IEEE Access,* vol. 7, no. 2169-3536, pp. 12192 - 12205, 15 January 2019.

[8] A. Khraisat, I. Gondal and P. e. a. Vamplew, "Survey of intrusion detection systems: techniques, datasets and challenges," 25 July 2019. [Online]. Available: https://doi.org/10.1186/s42400-019-0038-7. [Accessed 10 October 2020].

[9] M. K. Asif, . T. A. Khan, T. A. Taj and U. Naeem, "Network Intrusion Detection and its strategic importance," in *IEEE Business Engineering and Industrial Applications Colloquium (BEIAC). doi:10.1109/beiac.2013.6560100*, 2013.

[10] Prasad, J. Veeramreddy and M. Koneti, "Anomaly-Based Intrusion Detection System," no. DOI: 10.5772/intechopen.82287, June 11th 2019.

[11] C. Brook, "What is Deep Packet Inspection? How It Works, Use Cases for DPI, and More," 2018 Decemner 5. [Online]. Available: https://digitalguardian.com/blog/what-deep-packet-inspection-how-it-works-use-cases-dpi-and-more. [Accessed 15 November 2020].

[12] Data Robot, "Supervised Machine Learning," [Online]. Available: https://www.datarobot.com/wiki/supervised-machine-learning/. [Accessed 28 October 2020].

[13] J. Brownlee, "Supervised and Unsupervised Machine Learning Algorithms," 20 August 2020. [Online]. Available: https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/. [Accessed 28 October 2020].

[14] S. Shai and B. Shai, *Understanding Machine Learning: From Theory to Algorithm,* Cambridge University Press, 2014.

[15] K. Ansam and A. Ammar, "A critical review of intrusion detection," no. doi.org/10.1186/s42400-021-00077-7, pp. 1 - 27, 2021.

[16] . P. Agustin, G. Sebastian and J. E. Maria, "Stratosphere Laboratory. A labeled dataset with malicious and benign IoT network traffic," [Online]. Available: https://www.stratosphereips.org/datasets-iot23.

[17] P. Agustin, G. Sebastian and . J. E. Maria, "Stratosphere Laboratory. A labeled dataset with malicious and benign IoT network traffic," [Online]. Available: https://www.stratosphereips.org/datasets-iot23.

[18] N. A. Stoian, "Machine Learning for Anomaly Detection in IoT networks: Malware analysis on the IoT-23 Data set," University of Twente, Enschede, Netherlands, 2020.

[19] 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

IEEE 23rd International Multitopic Conference (INMIC), 2020.

[20] A. Kumar, M. Shridhar, S. Swaminathan and T. J. Lim, "Machine Learning-Based Early Detection of IoT Botnets Using Network-Edge Traffic," Singapore University of Technology and Design,, Singapore, 2020.

[21] GitHub, "Machine-Learning-Models-for-Detecting-Malicious-Traffic-in-IoT-Devices-using-IoT-23-Dataset," [Online]. Available: https://github.com/Bhageerath123/Machine-Learning-Models-for-Detecting-Malicious-Traffic-in-IoT-Devices-using-IoT-23-Dataset. [Accessed 20 March 2021].

1. Appendix A

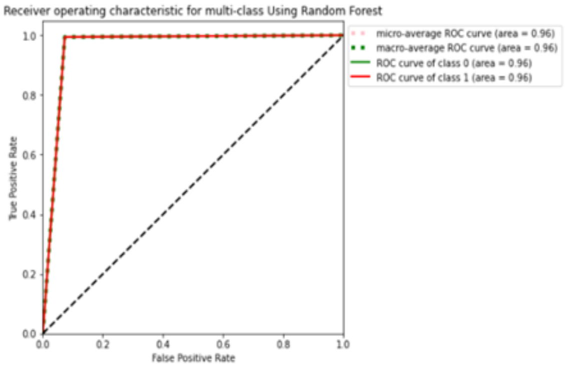

Figure 4 ROC-AUC Curve of Dataset 1 using Logistic Regression

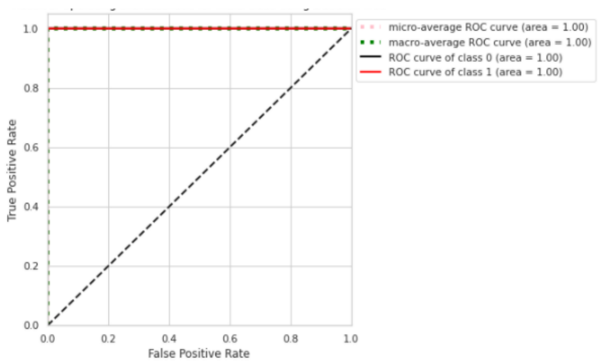

Figure 5: ROC_AUC Curve of Dataset 1 using Decision Tree Classifier

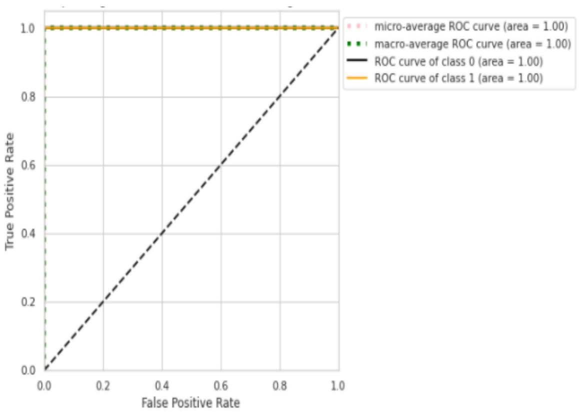

Figure 6: ROC_AUC Curve of dataset 1using Random Forest Classifier

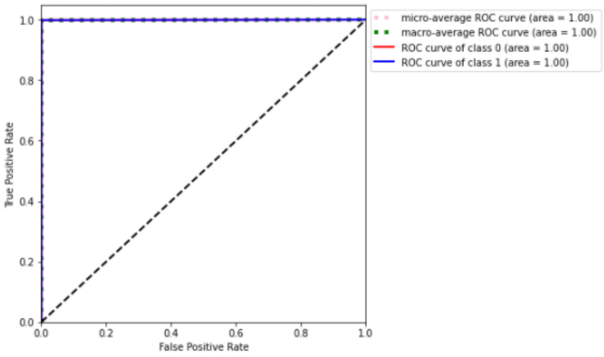

Figure 7: ROC_AUC Curve of Dataset 1 using XGBoost Classifier

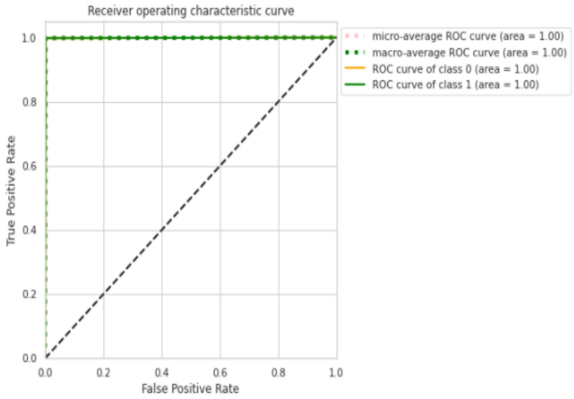

Figure 8: ROC_AUC Curve of Dataset 1 using Artificial Neural Network

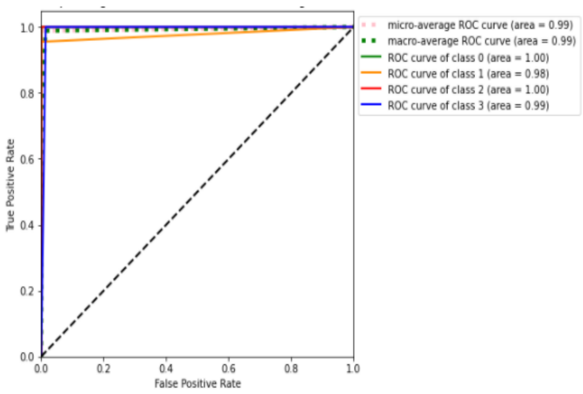

Figure 9: ROC_AUC Curve of Dataset II using Logistic Regression

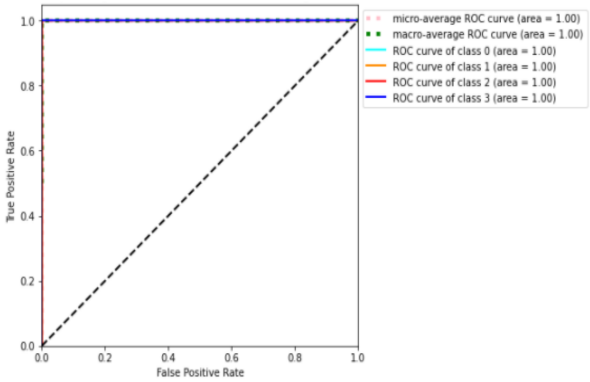

Figure 10: ROC_AUC Curve of Dataset II using Decision Tree Classifier

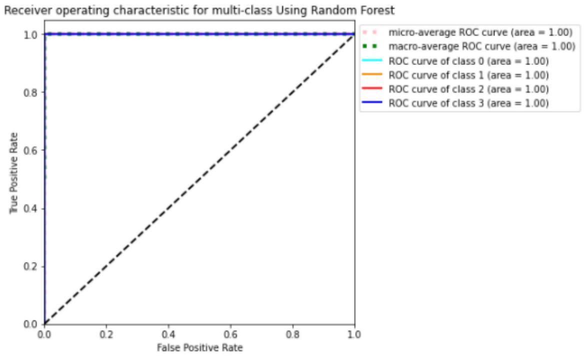

Figure 11:ROC_AUC Curve of Dataset II using Radom Forest Classifier

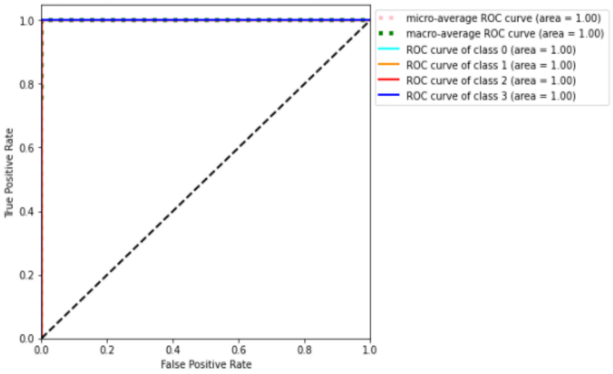

Figure 12: ROC_AUC Curve of Dataset II using XGBoost Classifier

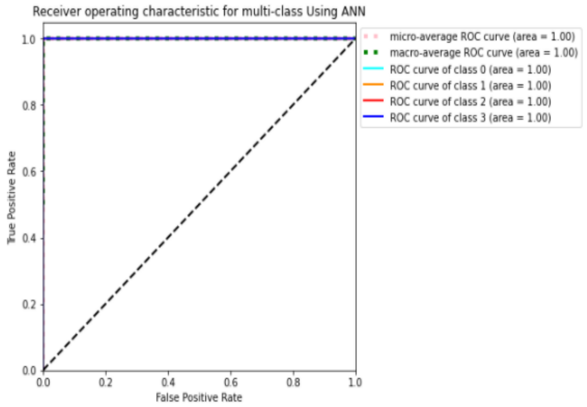

Figure 13: ROC_AUC Curve of Dataset II using Artificial Neural Network

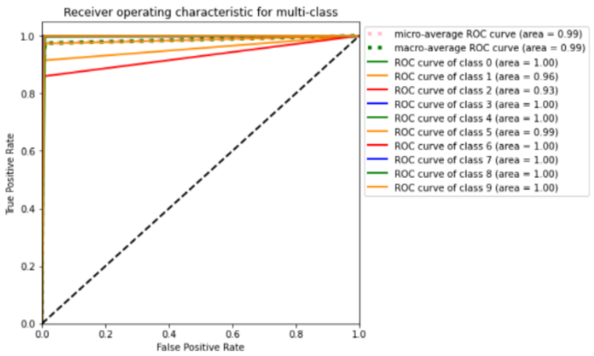

Figure 14: ROC_AUC Curve of Dataset III using Logistic Regression

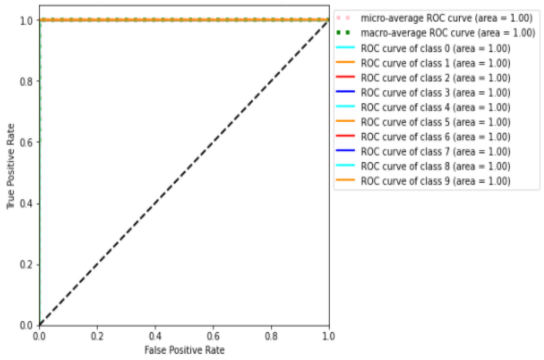

Figure 15: ROC_AUC Curve of Dataset III using Decision Tree

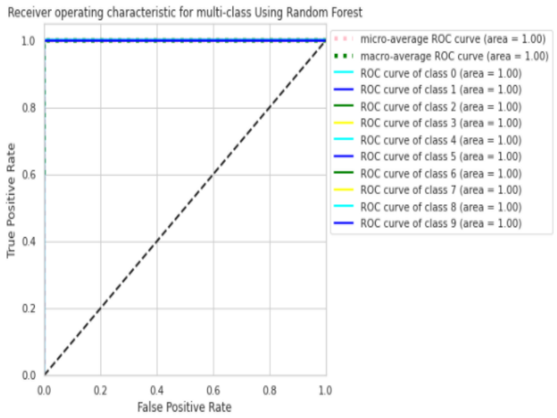

Figure 16: ROC_AUC of Dataset III using Random Forest Classifier

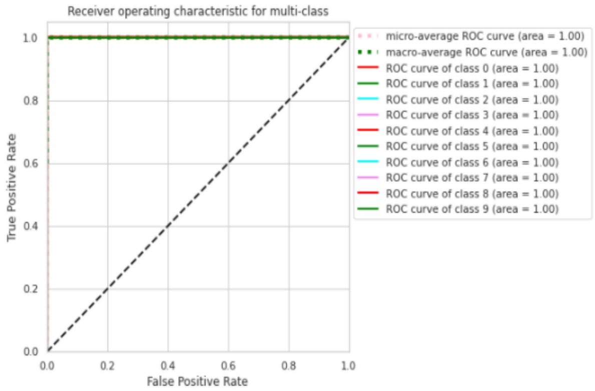

Figure 17: ROC-AUC Curve of Dataset III using XGBoost Classifier

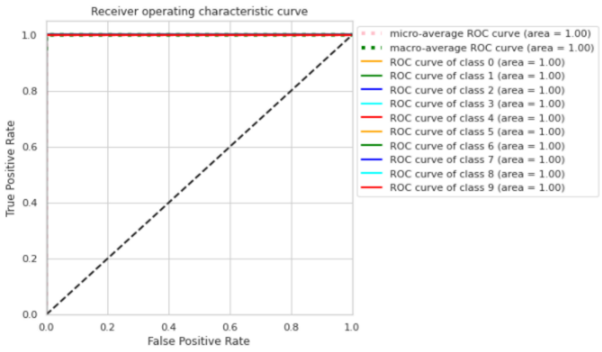

Figure 18: ROC_AUC Curve of Dataset III using Artificial Neutral Network