Journal of Harbin Engineering University Vol 44 No. 5
ISSN: 1006-7043 May 2023

Towards the Development of an Ensemble Intrusion Detection Model for DDoS and Botnet Mitigation using the IoT-23 Dataset

Shahbaz Ahmad Khanday[1][0000-0002-3256-894X],
Dr. Hoor Fatima[2][0000-0002-4285-2661]
Dr. Nitin Rakesh[3][0000-0002-1343-5244]
[1] Ph.D. Scholar, Department of Computer Science & Engineering, Sharda University, Knowledge Park III, Greater Noida India 201306
[2] Assistant Professor, Department of Computer Science & Engineering, Sharda University, Knowledge Park III, Greater Noida India 201306
[3] Professor Department of Computer Science & Engineering, Sharda University, Knowledge Park III, Greater Noida India 201306
https://www.sharda.ac.in/department/computer-science-and-engineering-cse

Abstract
The distribution of malware and bot formation of resource-constrained IoT networks has surged in the past few years. In IoT networks, prompt detection of intrusions is crucial due to the large-scale attacks pioneered by botmasters and botnets made up of unsecured IoT devices. The identification of such assaults has shown encouraging results using conventional machine learning models and customary deep learning approaches. Nevertheless, the use of certain algorithms may considerably benefit from a smaller feature set, because this could preclude the consequences of superfluous features and reduce the computation asset requisite for intrusion detection in such network systems having numerous restrictions. While most of the information insights in input data could be important and helpful but are excluded in the name of dimensionality reduction or feature selection process by IoT intrusion detection systems. In this study, an ensemble approach is proposed using Extra-Tree classifier for extracting important features, and a group of classifiers is tuned and tested for the classification attack and benign labels in the IoT-23 dataset. The manuscript proposes a novel intrusion detection model with a novel pre-processing technique along with a novel data preparation technique for binary and multiple classifications in IoT attack and malware mitigation. Comparative performance analysis of Linear Support Vector Classifier, Gaussian Naïve Bayes, and Ada-Boost is conducted in two case studies a) for Binary Classification and b) for Multiple Classification.
Keywords: Extra Tree Classifier, Linear Support Vector Classifier, Gaussian Naïve Bayes, Ada-Boost, and XG-Boost

1. Introduction

Improvements in real-life situations are being facilitated by IoT developments having major contributions to intelligent applications (e.g., Smart cities, IoT healthcare, autonomous vehicles, and IoT equipment for education). IoT technology's concurrent advancement and broad usage have created new security difficulties. Adhering to IoT security laws is challenging due to the intricate arrangements of new, occasionally ad hoc scenarios and their ambiguity. Most Internet of Things (IoT) devices are wirelessly connected and commonly utilized in an unattended way. In such situations, an attacker often gets access to IoT devices or networks easily, whether be it physically or logically. [1] By using hundreds of IoT devices as bots without the users' knowledge, an attacker with purportedly malicious intent could cause a serious, even fatal, impact on a target. Also, IoT devices are commonly distinguished by a low-cost manufacturing process (encompassing software and embedded design decisions, such as hazardous upgrade techniques, outmoded equipment, and obsolete security policies) and a lack of configuration concern on the part of typical users. Due to the numerous and serious flaws in IoT devices, everyday malware releases increasingly focus on infecting IoT networks as their major target.[2] Furthermore, an ever-growing collection of exploitation assets is available thanks to the fast spread of unsecured IoT devices and the simplicity with which attackers may find them via web services. Intruders nowadays can operate massive attacks like spam emailing, phishing URLs, and Distributed Denial of Service (DDoS) against Internet resources by exploiting and infecting a huge number of such exploitable IoT

devices. [3] IoT network penetration has become the principal purpose of ongoing malware campaigns as a result of the myriad and serious security flaws in IoT devices. Most modern IoT malware exploits low-level weaknesses in the infected devices to target IoT networks. It is vital to focus on effective IoT-specific antimalware countermeasures and identify the vulnerabilities introduced by malware advancements to IoT meta-systems. [4] To further disclose the security challenges in IoT meta systems we present some of the real-world examples of IoT-associated malware and botnet attacks via stealthy DDoS

I. Mirai: - A brute-force assault using the IoT devices may be remotely affected by Mirai malware on the Linux operating system and with Telnet (port 23) or port 2323 exposed. The infected equipment is used as a component of a botnet to launch a significant DDoS attack. Recent disruptive operations, like DDoS-enabled Rian Krebs' website in September 2016 but also a hit on Dyn in October 2016, were significantly influenced by the Mirai botnet. [5] On September 30, 2016

II. BASHLIE:- categorized as a DDoS-enabled malware to target Linux systems, Its early iteration from 2014 attacked BusyBox-enabled machines by leveraging a security flaw titled Shellshock inside the bash shell. A few weeks later, a new version was identified that might infect more devices that were susceptible.

III. One million devices were reportedly contaminated by 2016 as a result of the source code's publication in 2015, which sparked the development of further variants. Hajime: - Hajime is a development in IoT botnets since it uses a Peer - to - peer Command and control system that is significantly more evolved and differentiates bot conduct relying on the configuration of the device. The earliest Hajime samples came from Spain. [6] Although this worm creates a sizable Peer - to - peer botnet (almost 300,000 devices), its true function is yet unknown. [7]

IV. Torii: - A sophisticated version of Mirai was reported first in September 2018. Avast named the botnet strain Torii because of the telnet assaults Dr. Vesselin Bontchev found the incoming strain reached his honeypot through Tor exit nodes. [8]. Persirai and BrickerBot are among other upgraded variants of Mirai.

V. BrickerBot: - Brickerbot on the other hand, launches a succession of destructive Linux commands to break the unit irreversibly. Several of these instructions affect the memory space, kernel configuration, internet connectivity, gadget speed, and even the ability to delete all files from the device. [9]

VI. Other Bots: - Other DDoS-enabled bot attacks like Meris in 2021, have led to new dimensions and security challenges to the modern IoT metasystem. Many of these malware further evolved in upgrades according to the report cited by Stratosphere lab. [[10]]

VII.

1.1 Motivation and Contributions

One of the key goals for researchers is to increase the success rate of detecting harmful activity against the IoT ecosystem by initiating resistant workarounds for the IoT metasystem to spot malicious interventions. IoT security must recognize unwanted and harmful traffic to track, analyze, and prohibit improper data flows in the IoT network. Intervention and development of anti-malware distribution and prevention from botnet formation need to be addressed by the researchers. Furthermore, the involvement of extensible artificial intelligence should be brought into practice in building robust and lightweight intrusion detection systems suitable for IoT networks. Perhaps the existing intrusion detection systems to restrain the modern DDoS attack variants for IoT needs further assessment and development. Numerous academics have developed a variety of intrusion detection strategies leveraging machine learning and deep learning techniques to prevent fraudulent traffic flows in the network architecture. However, the most harmful traffic flows are frequently misclassified by some ML models because of erroneous and poor feature selection. The key query is: What favorable properties should be chosen for exact fraud traffic assessment in IoT networks? More research must be done on this topic.

In the proposed investigation, we suggested a novel data frame pre-processing technique that employs an aggregation tree-based importance and impurity to extract the feature importance of each attribute in the data frame. Then followed by dropping redundant attributes of the data frame to eliminate features with a lesser importance. Feature selection using tree-based importance (Extra Tree Classifier) is discussed in section 3.1.2 of the article. The research's major focus continues to be on identifying attack tactics and malware with DDoS functionality dissemination. The study emphasizes the determination of DDoS attack labels and malware proliferation in underlying IoT network data flow. Using the offered pre-processing approach and data frame preparation, of the

manuscript. Some of the findings of the manuscript are: -

- A unique data pre-processing strategy to extract the important and maximum features from the IoT-23 dataset.
- Two novel data-frame preparation models are proposed from the IoT-23 dataset, one for binary classification (between DDoS and Benign attack vectors) and the other resembling multiple classifications with various attack vectors contained within the dataset.
- DDoS attack mitigation model for IoT is proposed in the article.
- Comparative performance analysis of various classifiers (Linear Support Vector Classifier, Gaussian naïve Bayes, and Ada-Boost used for binary classification as well as multiple classification of attack vectors.

Table 1. Acronyms used throughout the manuscript

Acronyms	Full Form
KNN	K-Nearest Neighbour
SMOTE	Synthetic Minority Over-sampling Technique
IDS	Intrusion Detection Systems
MLP	Multilayer Perceptron
SDIoT	Software Defined Internet of Things
LNN	Lightweight Neural Networks
DoS	Denial of Services
GNN	Generative Neural Network
MQTT	MQ Telemetry Transport
ROC	Receiver Operator Characteristic
IG/GR	Information Gain/Gain ratio

The manuscript roadmap is given below in Figure 1.

1. Introduction
1.1 Motivation and Contribution
1.2 Manuscript Roadmap
2. Literature Survey
3. IoT Intrusion detection
3.1 Proposed Intrusion detection model
3.2 Data Pre-processing
3.3 Data Modelling
3.4 Data Evaluation
3.4.1 Anomaly Detection by a classifier
4. Results
I. Linear SVM Classifier
II. Gaussian Naive Bayes Classifier
III. Ada-Boost Classifier
IV. Comparison of the proposed model with existing research
V. Discussion
5. Conclusion and Future Scope
6. References

Figure 1. Manuscript Structure

2. Literature Survey

3. To defend against DDoS attacks as well as malware trafficking via the Internet, a variety of malware prevention tools were made available, used, and tested. To evaluate the most effective countermeasures for a potential solution to the test against malware dissemination via covert DDoS attack in IoT networks we are discussing the up-to-date state of the art in Table 2 of the manuscript. It also includes recent work in the area of IoT malware security measures put into practice. The IoT's characteristics and potential risks from utilizing botnets and IoT networks improperly are discussed in the literature review section.

4.

Table 2. Literature Survey

Citations and Year of Publication	Proposed Methodology used	Classifiers and methods used	Experimental data used
[11]	A CNN inspired model put into practice on BoT-IoT dataset for training and used the same model on smaller TON-IoT dataset for testing, With the help of	Convolutional Neural Network	Bot-IoT and TON-IoT dataset by USNW

	DL-IDS employing, we can train up and fine-tune previously learned models on smaller samples with unexpected attack patterns.		
[12]	The authors introduce a unique NID strategy suitable for IoT metasystems employing a compact deep neural network (LNN)	Deep Neural Network	-
[13]	In a variety of ML methods, a quantitative bijective soft-set technique is used to determine which ML model is the most productive.	Naïve Bayes classifier	Bot-IoT
[14]	This paper recommends a method for choosing the optimum characteristics by using IG and GR do identify the DDoS and DoS assaults.	JRipclassifier, Information Gain and Gain Ratio	Bot-IoT and KDD-Cup 99 dataset
	The proposed approach constructs feature subsets by executing insertion and union operations mostly on units to harvest for 50% IG and GR attributes.		
[15]	Decision tree models constructed from collected Bot-IoT dataset with a maximum of three features serve as a definition of the authors' contributions.	Decision Tree classifier	Bot-IoT
[16]	This study proposes a CNN-based approach for anomaly-based intrusion detection systems (IDS). By providing the ability to properly analyse IoT traffic logs, this strategy has taken advantage of the potential	Convolutional Neural Network	NID and Bot-IoT dataset

	of IoT infrastructure.		
[17]	In order to solve this problem, this research offers a labeled contextual IoT data set on a moderate IoT testbed that comprises both legitimate and fraudulent botnet network traffic. Data from botnet infection, spread, and interactions with Command and control server stages are obtained after the deployment of three well-known botnet infections (Mirai, BashLite, and Torii).	K-Nearest Neighbour, Decision Tree, and Random Forrest	MedBIoT dataset
[18]	Our suggested model addresses the safety challenge regarding the hazards posed by bots. The BoT-IoT dataset was used to train a variety of machine learning algorithms with and without class balancing. The researchers propose two case studies and the highest accuracy is 88% obtained by using KNN on a balanced dataset.	K-Nearest Neighbour, Naïve Bayes, and MLP	Bot-IoT
[19]	The paper uses a unique design deep learning method known as the Bidirectional -Gated Recurrent Unit-Convolutional Neural Network (Bi-GRU-CNN for classifying the assaults in IoT.	As input to the Bi-GRU-CNN model, the model uses the binary file byte sequence. Moreover, the t-distributed Stochastic Neighbor Embedding (t-SNE) visualisation method is employed.	Taiwan Information Security Centre.
[20]	To aggregate IoT malicious application code and evaluate	Script-based analysis.	IoTPOT dataset.

	their underlying relationships by removing hazardous IPs and/or additional instructions, the manuscript recommends creating and implementing a multi-level strings-based malware similarity evaluation approach.				DDoS attacks. The cosine resemblance of the parameters of the packet-in messaging rate at frontier SD-IoT network switches is used to assess that DDoS instances are emerging.		
[21]	A monitoring system for the SD-IoT switches for the SD-IoT connected to an Application interface, and Sensor nodes make up the suggested testbed configuration. Then, utilizing the suggested SD-IoT infrastructure, the authors describe a method for identifying and thwarting	_	Dataset by the National Natural Science Foundation of China	[22]	In this research, we suggest a framework for detecting application-layer denial-of-service (DoS) attacks against the MQTT protocol and evaluate the system using real-world, configuration DoS attack scenarios. A conventional machine framework was created for the MQTT protocol to guard against such attacks.	AODE (Averaged one-dependence estimators), C4.5 and MLP	Using a specially created MQTT attack design relying on the Ecliplse-Paho library, DoS attack traffic was produced (Eclipse, 2018).

[23]	To repel and safeguard the distributed denial of service (DDoS) assaults in an IoT meta-system, the researchers have suggested a multi-objective optimization -based strategy to select negligible features.	ELM feed-forward neural network	CICIDS2017 dataset

3. IoT Intrusion Detection Systems.

Given that most of the research focuses on developing IoT intrusion detection systems leveraging extensible artificial intelligence domains, this approach is the best one for researchers. The Intrusion detection model presented in this manuscript uses an Extra Tree Classifier for extraction of important features (Select from Model) feature selection techniques are used) and an MLP classifier for discriminating attack vectors. Researchers are encouraged to create reliable and efficient intrusion detection and intrusion prevention systems by considering machine learning techniques. Additionally, it has been demonstrated that DL's capacity for enhanced intellect may be utilized to detect newly distributed attacks in IoT systems thanks to its improved correctness and application performance. In addition to the usual classification and regression issues, intrusion detection systems must be extremely important in monitoring and examining IoT network activity in comparison to firewalls and ultimately the services offered by the firewalls. This is crucial when discussing the security of thin-client IoT networks because they are liable to several protection vulnerabilities, such as malware deployment, botnet formation, and eavesdropping, as well as having capacity limits, such as low capacity memory and processing chips, risky developmental tools, etc. [24] The lack of a feature set selected by humans, the ability for autonomous pre-training, and the compaction qualities of deep learning, on the other hand, allow it to be applied even in networks with resource limitations.[25], [26]. The proposed model is further defined below in Figure 3.

IDS (Intrusion Detection Systems) receive a variety of data in the form of network traffic logs, application logs, binaries or unprocessed notifications, incident footprints, and attack data are a few examples. An IDS uses standard logging or methods like disc, memory, packet, function, and code to collect data systematically or proactively from various sources like applications or systems. It is frequently possible to carry out methods for data collection and recognition separately by an IDS. An IDS can look at and detect abnormal activities in a network and infiltration can also be found. [27] Before providing the raw data to any machine learning or deep learning model, the above approach uses some basic data pre-processing stages. The manuscript's section 3.1 goes into more detail about the building blocks of the suggested Intrusion detection model. The given architecture of the model remains the same for both binary and multiple classification except the multiple classification model has more than two target classes.

3.1 Proposed Intrusion detection model

The various building blocks of the proposed model are given in Figure.2 below

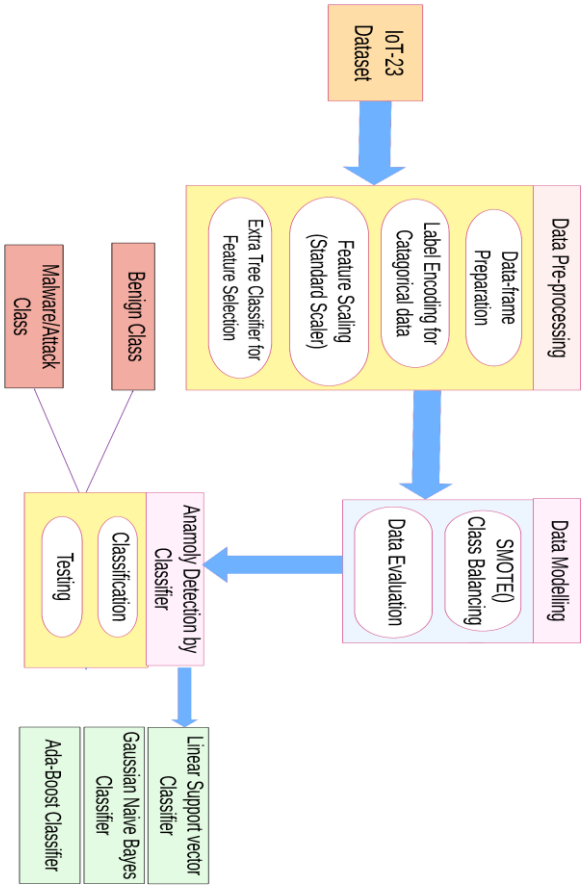

Figure 2. Intrusion detection model

3.1.1 Input Dataset

The Aposemat IoT 23 dataset consists of two distinct categories for experimentation: IoT malware containing 20 data frames and IoT Honeypot comprising 3 data frames. [28] The five data frames specifically IoT-Malware-Capture-43-1, IoT-Malware-Capture-49-1, IoT-Malware-Capture-52-1, IoT-Malware-Capture-36-1, as well as numerous additional data frames, comprise a multitude of collected instances, in which we have extracted approximately ten thousand from other data frame with the PCAP file sizes spanning within 1.5 - 6 Gigabytes and might necessitate high-end computational resources.

3.1.2 Data Pre-processing

The Aposemat IoT 23 dataset consists of two distinct categories for experimentation: IoT malware containing 20 data frames and IoT Honeypot comprising 3 data frames. [28] The five data frames specifically IoT-Malware-Capture-43-1, IoT-Malware-Capture-49-1, IoT-Malware-Capture-52-1, IoT-Malware-Capture-36-1, as well as numerous additional data frames, comprise a multitude of collected instances, in which we have extracted approximately ten thousand from other data frame with the PCAP file sizes spanning within 1.5 - 6 Gigabytes and might necessitate high-end computational resources. Figures 3 and 4 depict the data preparation model for binary and multiple classification.

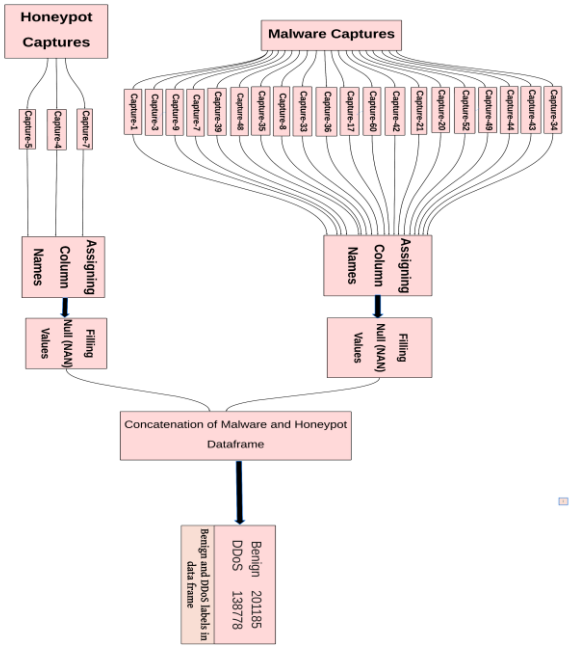

Figure 3. Data Preparation Model for Binary Classification

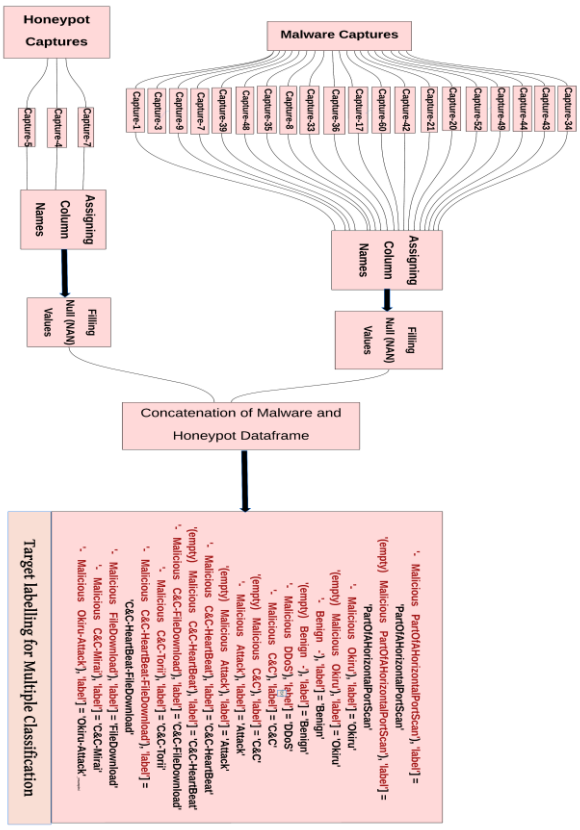

Figure 4. Data Preparation Model for Multiple Classification

In the data frame preparation modelling, we are labelling the records using column names. NaN and duplicate values are removed from every malware and honeypot capture. Then all captures are concatenated into single data-frame. Target labelling for multiple classification block depicts the labels assigned to attack vectors of the concatenated data frame, while in data frame preparation for binary classification only DDoS and Benign samples are used.

Features namely 'ts'. 'uid', 'id.orig_h', 'id.resp_h', 'proto', ''service, 'duration', 'org_bytes', 'resp_bytes' 'conn_state', 'local_orig', 'local_resp', 'history' and 'label' are categorical features and converted to integers using label encoding. For feature scaling the data in the columns must be normalized into a range and a standard scaler is used in the pre-processing. For feature selection, the importance of the best twenty features is plotted using an Extra-Tree classifier with a meta-estimator and the removal of redundant features on the bases of feature importance.

This module for feature selection defines a meta-estimator that uses averaging to improve projected accuracy and reduce overfitting when modeling a number of randomly generated decision trees (also known as extra trees) on different sub-samples of the data frame. [29] Also, each tree is given a randomized set of various input data parameters from which it must choose the feature that will most profitably segment the input data using the Gini Index. [30] The determining element in this situation is going to be Information Gain. finding the entropy of the data first. The entropy is given by

$Entropy(S) = \sum_{i=1} c - pi\log2(pi)$

(1)

Where c is the class labels, $\boldsymbol{p}$i represents the proportion of rows and i indicate output labels

Calculations of Gain at every tree are given by;

$Gain\ (S, A) = \sum v\varepsilon Values(A)$ |Sv| / |S| Entropy (Sv)

(2)

A represents an individual feature of data. Similarly, the Gain for each feature is calculated by:

Gain (Gn) = (Entropy (S) , An) represents the feature gain named An at Decision tree Dn

Therefore, After the process, the total gain of every attribute is determined, and the attributes having the highest Gain are designated to be vital features or important features. The extra tree classifier is used to extract the fifteen best features of the Iot-23 data frame and the feature importance is plotted in Figure 5 below.

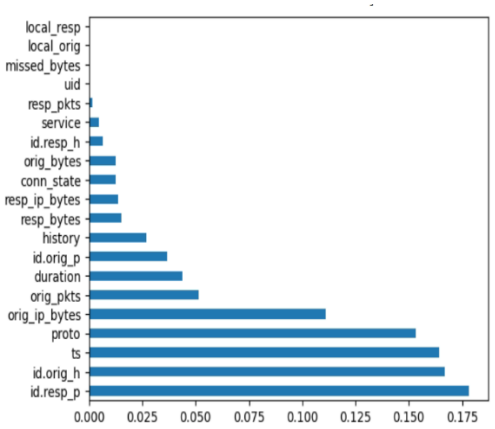

Figure. 5 Feature importance of the fifteen best features

3.1.3 Data Modelling

- Synthetic Minority Over-sampling Technique (SMOTE)

Synthetic Minority Over-sampling Technique (SMOTE) is a machine learning approach to oversample the minority class of data frame irrespective of the fact that most machine learning models produce ineffective and biased results when classes are not equally distributed. In our research, we used SMOTE() in IoT-23 datasets to balance labels in both case studies. The target class in IoT-23 dataset is heavily imbalanced as there exist only 201185 normal (Benign) labels over 138778 DDoS (Malicious) instances for binary classification. The target variable of the datasets represents class 0 as normal instances and class 1 as malicious/attack instances. In multiple classifications, we have obtained an attack vector of multiple attack types evenly sampled by SMOTE oversampling. The statistics of IoT attacks for multiple classifications are given in Figure 6 of the manuscript. Labels not exceeding ten thousand in number are excluded.

```
PartOfAHorizontalPortScan      878471
Okiru                          262693
Benign                         201185
DDoS                           138778
C&C                             15101
Attack                           6063
C&C-HeartBeat                     349
C&C-FileDownload                   43
C&C-Torii                          30
FileDownload                       13
0                                  10
C&C-HeartBeat-FileDownload          8
C&C-Mirai                           1
Name: label, dtype: int64
```

Figure. 6 Attack labels in the IoT-23 dataset for multiple classifications.

Figure 6 shows the attack vector in the target variable, which is unbalanced. In our approach of multiple classifications, the attack labels below ten thousand are excluded to effectively balance the target classes.

- Data Evaluation

In the Splitting procedure, the pre-processed data frame is split into train and testing partitions with 80% of the data frame for training and 20% for testing.

Anomaly detection by the classifier: - The binary classification test is performed on the input pre-processed data by using Linear Support Vector Classifier, Ada-Boost, Naïve Bayes Classifier, and Convolutional Neural Network classifiers, to determine if the instance belongs to either the Normal or attack class. Normal Traffic class and Attack Traffic class are the outcome labels of the classifier evaluated in a classification report. In the result section of the manuscript, a classification report contains ROC Curve, Confusion Matrix, Precision, Recall, F1 Score, and Support derived from the classifier. On the other hand, Linear Support Vector Classifier, Logistic Regression, Ada-Boost, Gaussian Naïve Bayes Classifier, and Convolutional Neural Networks are also used in multiple classifications. The performance of the classifiers is evaluated by Accuracy percentage, Precision, Recall, F1 Score, and Support (with weighted and macro average).

4. Results

4.1 Linear Support Vector Classifier

A hyperplane in space with N dimensions that can categorize observations is what the SVM algorithm looks for. To get the desired results, the major objective of the experiment was to use SVM for both basic binary and multiple classifications. The support vector machine (SVM) model is trained using a linear support vector classifier algorithm. Below is a description of the form that linear SVM adopts:

$$H(x) = w^T x + b \sum_{i=1}^{n} wixi + b$$

(I)

x is the feature vector represented with feature dimensionality n to obtain the optimum weight vector w and bias b.

Comparable to SVC using kernel as linear, but built using linear library instead of SVM, giving it greater versatility about the penalized and loss functions that can be used. As a result, it ought to adapt to a significant amount of data points more effectively. Both dense and sparse supply is supported by this class, and the multiple classes support is dealt with using a one-against-the-rest strategy. [31] The results achieved using Linear SVC are given below in Table 3 and ROC curve in Figure 7.

Table 3. Classification Report of Linear SVC as Binary and Multiple Classification

Classification Test		Precision	Recall	F1-Score	Support	Confusion Matrix (TP, FN FP, TN)
Linear SVC for Binary Classification	Accuracy			0.999	67993	40226 32 17 27718
	Macro Average	1.0	0.99	1.0	67993	
	Weighted Average	1,0	1.0	1.0	67993	
Linear SVC for Multiple Classification	Accuracy			0.68	287460	
	Macro Average	0.47	0.64	0.47	287460	
	Weighted Average	0.61	0.90	0.67	287460	

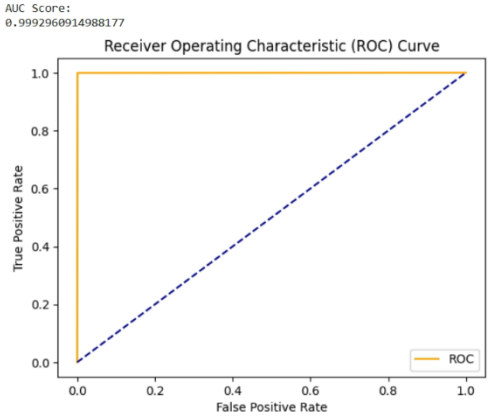

Figure 7. ROC Curve of Binary Classification using Linear Support Vector Classifier

4.2 Gaussian Naïve Bayes Classifier

Gaussian Naive Bayes can use the partial fit method for performing current changes to model variables. Current information about the algorithm for updating means and variance of the feature set. To achieve online or out-of-core learning, the partial fit method is anticipated to be invoked repeatedly on various dataset segments. [32] This is particularly helpful whenever the entire dataset cannot be stored in the CPU at once and most opted for imbalanced datasets, henceforth a prime choice in our study. It is speculated that the likelihood of the features is Gaussian, employing a Gaussian Naïve Bayes classifier.

$$P(x^i \mid y) = \left(\frac{1}{\sqrt{2\sigma_y^2}}\right)^{\left(\frac{x_i - \mu_y{}^2}{2\sigma_y^2}\right)} \quad (1)$$

$\boldsymbol{\sigma_y}$ and $\boldsymbol{\mu_y}$ are calculated by the maximum likelihood. The results achieved using Linear SVC are given below in Table 4 and Figure 8.

Table 4. Classification Report of Gaussian Naïve Bayes as Binary and Multiple Classification

Classification Test		Precision	Recall	F1-Score	Support	Confusion Matrix (TP, FN FP, TN)	
Gaussian Naive Bayes for Binary Classification	Accuracy			0.95	67993	36559 3	3699 27732
	Macro Average	0.94	0.95	0.94	67993		
	Weighted Average	0.95	0.95	0.95	67993		
Gaussian Naive Bayes for Multiple Classification	Accuracy			0.83	287460		
	Macro Average	0.74	0.59	0.58	287460		
	Weighted Average	0.85	0.83	0.78	287460		

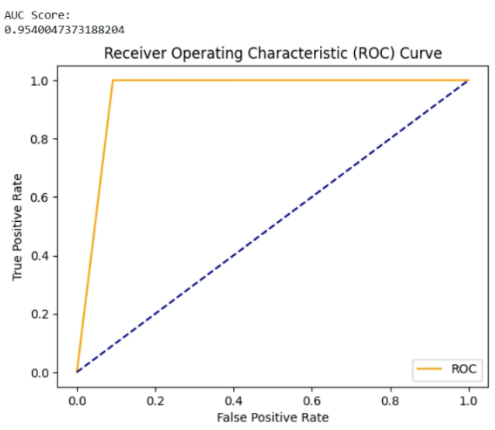

Figure 8. ROC Curve of Binary Classification by Gaussian Naïve Bayes

4.3 Ada-Boost Classifier

AdaBoost's main tenet is to train a series of inadequate learners (i.e., models which are just marginally superior to arbitrary speculation, like decision trees) on frequently altered renditions of the data. After then, a weighted consensus voting (or total) is used for incorporating all of their guesses to come up with the best guess of all. Weights (w1, w2, w3.....wn) are applied to each of the training sets during each boosting iteration, which modifies the data. Since the initial choices for all weights are, the very initial step merely instructs a mediocre trainee on the raw data. $\omega_i = 1/N$ The collection values are individually changed for any further iteration, and the acquiring process is then performed once more to the reweighted input. [33] In the proposed estimation, an Ada-boost classifier is trained by the Malware and Benign samples with n-estimators equal to 300. The results obtained using Ada-Boost as a classifier in the proposed intrusion detection model are given in Table 5 and Figure 9.

Table 4. Classification report of Ada-boost classifier as Binary and Multiple Classification

Classification Test		Precision	Recall	F1-Score	Support	Confusion Matrix (TP, FN FP, TN)
Ada-boost classifier for Binary Classification	Accuracy			0.99	67993	40058 1 277 34
	Macro Average	1.0	1.0	1.0	67993	
	Weighted Average	1.0	0.99	1.0	67993	
Ada-boost classifier for Multiple Classification	Accuracy			0.84	287460	
	Macro Average	0.55	0.56	0.55	287460	
	Weighted Average	0.76	0.84	0.78	287460	

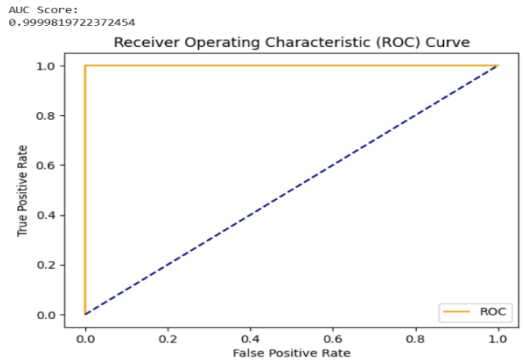

Figure 9. ROC Curve of Binary Classification by Ada-boost

Precision, Recall, F-1 Score, and Support values in multiple classifications are given below in Table 6, Table 7, Table 8, and Table 9 of the article.

Table 6. Precision, Recall, F1-Score, and Support in Linear Support Vector Classifier for Multiple Classification

Target Labels	Label Count	Precision	Recall	F1-Score	Support
Benign	191969	0.98	0.30	0.46	38204
C&C	14295	0.69	0.11	0.18	2870
DDoS	132574	0.97	0.82	0.89	26660
Okiru	251627	0.0	0.0	0.0	50591
Part Of A Horizontal Port Scan	846835	0.66	0.95	0.78	169135

Table 7. Precision, Recall, F1-Score and Support in Gaussian Naïve Bayes for Multiple Classification

Target Labels	Label Count	Precision	Recall	F1-Score	Support
Benign	191969	0.97	0.6	0.12	38204
C&C	14295	0.14	0.11	0.12	2870
DDoS	132574	0.79	0.82	0.81	26660
Okiru	251627	1.0	0.98	0.99	50591
Part Of A Horizontal Port Scan	846835	0.80	0.97	0,87	169135

Table 8. Precision, Recall, F1-Score, and Support in Ada-Boost for Multiple Classification

Target Labels	Label Count	Precision	Recall	F1-Score	Support
Benign	191969	0.0	0.6	0.0	38204
C&C	14295	0.0	0.0	0.0	2870

DDoS	132574	1.0	0.82	0.90	26660
Okiru	251627	0.93	1.0	0.96	50591
Part Of A Horizontal Port Scan	846835	0.80	1.0	0,89	169135

4.4 Comparison of the proposed model with existing research.

Table 9. of the manuscript shows a detailed description of the research articles in which the IoT-23 dataset is used. The table highlights the significance of the feature selection test, classification metric, and dropped and used features from the IoT-23 dataset.

Table 9. Comparison of proposed model vs existing research

Author/ Citation	Features used from the dataset	Features dropped from the dataset	Feature selection method used	Classification Type	Classification Metric
Stoian in 2020 [34]	'id.resp_h','id.resp_p',' proto', 'service',' duration', 'orig_bytes','resp_bytes','conn_state',' history','orig_pkts','orig_ip_bytes','resp_pkts','resp_ip_bytes',' label'	'ts', 'uid', 'id.org_h', 'local_org', local_resp', 'missed_bytes','tunnel_parents	Pearson's Correlation and Heatmap	Multiple	Accuracy using Naïve Bayes = 0.23, Support Vector machine = 0.67 and Ada-Boost = 0.87
Abdalgawad et al, in 2022 [35]	'proto', 'service', 'duration', 'orig_bytes', 'resp_bytes', 'conn_state', 'orig_pkts' and 'resp_pkts'	'local_orig', 'local_resp' 'orig_ip_bytes' 'resp_ip_byte'; 'id.org_h', id.org_p', 'id.resp_h', 'id.resp_p' ,'uid, 'ts' , 'missed_bytes', 'org_ip_byte','org_resp_bytes', 'history' and 'missed_bytes'	Correlation and Boxplots	Multiple classification	The average Macro-F1 score using K-Nearest Neighbour is 0.6019 and using Random Forest achieved 0.5518

[36] (Thavasimani & Kasturirangan Srinath in 2022	'id.orig_h', 'id.orig_p', 'id.resp_h', 'id.resp_p', 'proto', 'service', 'duration', 'orig_bytes', 'resp_bytes', 'conn_state', 'history', 'orig_pkts', 'orig_ip_bytes', 'resp_pkts', 'resp_ip_bytes', and 'tunnel_parents'	ts' , 'uid', 'id.org_h', 'local_org', local_resp', 'missed_bytes'	Chi-Square Test	Binary Classification	Accuracy obtained using Custom Genetic Algorithm =0.988
Proposed Model	'ts' 'id.orig_h' 'id.orig_p'.'id.resp_h' 'id.resp_p' 'proto 'service' 'duration' 'orig_bytes' 'resp_bytes' 'conn_state' 'history' 'orig_pkts' 'orig_ip_bytes' 'resp_ip_bytes' 'label'	'uid', 'resp_pkts','missed_bytes', 'local_orig', 'local_resp'	Tree-Based Importance Using Extra Tree Classifier	Both Binary classification for DDoS attack mitigation and multiple classification	Accuracy using Linear SVC obtained 0.999 for binary classification and 0.845 for multiple classification 0.99 and 0.83 by Ada-Boost Classifier and 0.956 and 0.83 using Gaussian naïve Bayes Classifier

5. Discussion

According to the performance measurements, all four classifiers demonstrated extremely effective performance for DDoS attack detection. Contrarily, the LSVC performed poorly for multiple classifications, only scoring 68%, and completely disregarded the labels for Okiru assault in the test data. The Ada-boost classifier for multiple classifications similarly disregarded classes 0 and 1. Ada-boost has proven to be a capable classifier, yet it is sufficient. In comparison to the other members of the group, Gaussian Naive Bayes has demonstrated the best overall performance for all attack vectors. The novelty of the proposed model is a) the proposal of DDoS attack mitigation in IoT infrastructure to defend against unintended bot recruitment of resource-constrained IoT devices and has obtained better results

compared to the Custom Genetic Algorithm used by Thavasimani & Kasturirangan Srinath in 2022 [36] b) various kinds of IoT malware mitigation making the attack vector of the dataset and c) an innovative approach of selecting a feature set for IDS model. Research using IoT-23 conducted by various researchers has taken advantage of a reduced data frame dropping the majority of features from the IoT-23 dataset. Likewise [37] in his research article dropped seven features (ts, uid, id.orig_h, local_orig, local_resp , missed_bytes, tunnel_parents), a research article by [35] dropped thirtteen features and researchers of the article [38] dropped more than 50% of the features from the Iot-23 dataset. In comparison to that we have dropped only five features with minimum importance out of twenty features in the proposed model and classifiers within have performed better than the compared models.

6. Conclusion and Future Scope

To monitor and limit unwanted traffic flows, IoT well-being necessitates the recognition and investigation of malicious activities. Due to the increasing number of weak IoT devices, botnet-based malware injections pose a serious security threat to the IoT ecosystem. Many researchers have worked and proposed machine and deep learning-based autonomous solutions for developing anti-malware systems to analyze IoT network traffic and data. However, because of inadequate feature selection, some machine learning algorithms are prone to incorrectly identifying mostly damaging traffic flows. In the proposed IDS model implementation, a unique feature selection technique is to select distinct feature sets from the input data and fed them to the classifier. Finding out the best feature set from the input data for a classifier is the goal of the proposed research. From the set of classifiers, it is concluded that the Ada-Boost and Gaussian Naïve Bayes classifiers outperform Linear Support vector classifier. Also, it is observed that Gaussian Naïve Bayes Classifier has worked efficiently for binary classification as well as for multiple attack class detection. The future work is reserved to include various ensemble and hybrid classifiers, where the performance of the proposed model can be improved and upgraded. For the same objective, different Deep Learning approaches might be taken into consideration as well. Due to the requirement of high-end processing expenditure, we were only able to extract a small chunk of the IoT-23 dataset from the enormous log files. We intend to deploy the proposed model for large-size datasets, including the Iot-23 dataset.

Refrences

[1] K. Albulayhi, A. A. Smadi, F. T. Sheldon, and R. K. Abercrombie, ‘IoT Intrusion Detection Taxonomy, Reference Architecture, and Analyses’, *Sensors*, vol. 21, no. 19, Art. no. 19, Jan. 2021, doi: 10.3390/s21196432.

[2] G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescapé, ‘A Hierarchical Hybrid Intrusion Detection Approach in IoT Scenarios’, in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, Dec. 2020, pp. 1–7. doi: 10.1109/GLOBECOM42002.2020.9348167.

[3] C. D. McDermott, F. Majdani, and A. V. Petrovski, ‘Botnet Detection in the Internet of Things using Deep Learning Approaches’, in *2018 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2018, pp. 1–8. doi: 10.1109/IJCNN.2018.8489489.

[4] D. Vasan, M. Alazab, S. Venkatraman, J. Akram, and Z. Qin, ‘MTHAEL: Cross-Architecture IoT Malware Detection Based on Neural Network Advanced Ensemble Learning’, *IEEE Trans. Comput.*, vol. 69, no. 11, pp. 1654–1667, Nov. 2020, doi: 10.1109/TC.2020.3015584.

[5] M. Kan, ‘Hackers create more IoT botnets with Mirai source code’, *Computerworld*, Oct. 18, 2016. https://www.computerworld.com/article/3132570/hackers-create-more-iot-botnets-with-mirai-source-code.html (accessed Oct. 26, 2022).

[6] S. Herwig, K. Harvey, G. Hughey, R. Roberts, and D. Levin, ‘Measurement and Analysis of Hajime, a Peer-to-peer IoT Botnet’, in *Proceedings 2019 Network and Distributed System Security Symposium*, San Diego, CA: Internet Society, 2019. doi: 10.14722/ndss.2019.23488.

[7] ‘Hajime, the mysterious evolving botnet’. https://securelist.com/hajime-the-mysterious-evolving-botnet/78160/ (accessed Oct. 27, 2022).

[8] M. Smith, ‘New vicious Torii IoT botnet discovered’, *CSO Online*, Oct. 01, 2018. https://www.csoonline.com/article/3310222/new-vicious-torii-iot-botnet-discovered.html (accessed Oct. 28, 2022).

[9] ‘BrickerBot Malware Emerges, Permanently Bricks IoT Devices - Security News’. https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/brickerbot-malware-permanently-bricks-iot-devices (accessed Oct. 28, 2022).

[10] 'A Study of IoT Malware', *Stratosphere IPS*. https://www.stratosphereips.org/a-study-of-iot-malware (accessed Dec. 03, 2022).

[11] Kawale, S., Dhabliya, D., & Yenurkar, G. (2022). Analysis and Simulation of Sound Classification System Using Machine Learning Techniques. 2022 International Conference on Emerging Trends in Engineering and Medical Sciences (ICETEMS), 407–412. IEEE.

[12] I. Idrissi, M. Azizi, and O. Moussaoui, 'Accelerating the update of a DL-based IDS for IoT using deep transfer learning', *Indones. J. Electr. Eng. Comput. Sci.*, vol. 23, no. 2, p. 1059, Aug. 2021, doi: 10.11591/ijeecs.v23.i2.pp1059-1067.

[13] R. Zhao *et al.*, 'A Novel Intrusion Detection Method Based on Lightweight Neural Network for Internet of Things', *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9960–9972, Jun. 2022, doi: 10.1109/JIOT.2021.3119055.

[14] M. Shafiq, Z. Tian, Y. Sun, X. Du, and M. Guizani, 'Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city', *Future Gener. Comput. Syst.*, vol. 107, pp. 433–442, Jun. 2020, doi: 10.1016/j.future.2020.02.017.

[15] P. Nimbalkar and D. Kshirsagar, 'Feature selection for intrusion detection system in Internet-of-Things (IoT)', *ICT Express*, vol. 7, no. 2, pp. 177–181, Jun. 2021, doi: 10.1016/j.icte.2021.04.012.

[16] J. L. Leevy, T. M. Khoshgoftaar, and J. Hancock, 'IoT attack prediction using big Bot-IoT data', *Int. J. Internet Things Cyber-Assur.*, vol. 2, no. 1, pp. 45–61, Jan. 2022, doi: 10.1504/IJITCA.2022.124373.

[17] T. Saba, A. Rehman, T. Sadad, H. Kolivand, and S. A. Bahaj, 'Anomaly-based intrusion detection system for IoT networks through deep learning model', *Comput. Electr. Eng.*, vol. 99, p. 107810, Apr. 2022, doi: 10.1016/j.compeleceng.2022.107810.

[18] A. Guerra-Manzanares, J. Medina-Galindo, H. Bahsi, and S. Nõmm, 'MedBIoT: Generation of an IoT Botnet Dataset in a Medium-sized IoT Network':, in *Proceedings of the 6th International Conference on Information Systems Security and Privacy*, Valletta, Malta: SCITEPRESS - Science and Technology Publications, 2020, pp. 207–218. doi: 10.5220/0009187802070218.

[19] S. Pokhrel, R. Abbas, and B. Aryal, 'IoT Security: Botnet detection in IoT using Machine learning'. arXiv, Apr. 05, 2021. doi: 10.48550/arXiv.2104.02231.

[20] R. Chaganti, V. Ravi, and T. D. Pham, 'Deep learning based cross architecture internet of things malware detection and classification', *Comput. Secur.*, vol. 120, p. 102779, Sep. 2022, doi: 10.1016/j.cose.2022.102779.

[21] S. Torabi, M. Dib, E. Bou-Harb, C. Assi, and M. Debbabi, 'A Strings-Based Similarity Analysis Approach for Characterizing IoT Malware and Inferring Their Underlying Relationships', *IEEE Netw. Lett.*, vol. 3, no. 3, pp. 161–165, Sep. 2021, doi: 10.1109/LNET.2021.3076600.

[22] D. Yin, L. Zhang, and K. Yang, 'A DDoS Attack Detection and Mitigation With Software-Defined Internet of Things Framework', *IEEE Access*, vol. 6, pp. 24694–24705, 2018, doi: 10.1109/ACCESS.2018.2831284.

[23] N. F. Syed, Z. Baig, A. Ibrahim, and C. Valli, 'Denial of service attack detection through machine learning for the IoT', *J. Inf. Telecommun.*, vol. 4, no. 4, pp. 482–503, Oct. 2020, doi: 10.1080/24751839.2020.1767484.

[24] M. Roopak, G. Y. Tian, and J. Chambers, 'Multi-objective-based feature selection for DDoS attack detection in IoT networks', *IET Netw.*, vol. 9, no. 3, pp. 120–127, 2020, doi: 10.1049/iet-net.2018.5206.

[25] X. Ma *et al.*, 'A Survey on Deep Learning Empowered IoT Applications', *IEEE Access*, vol. 7, pp. 181721–181732, Jan. 2019, doi: 10.1109/ACCESS.2019.2958962.

[26] W. Wang, J. Liu, G. Pitsilis, and X. Zhang, 'Abstracting massive data for lightweight intrusion detection in computer networks', *Inf. Sci.*, vol. 433–434, pp. 417–430, Apr. 2018, doi: 10.1016/j.ins.2016.10.023.

[27] W. Wang *et al.*, 'HAST-IDS: Learning Hierarchical Spatial-Temporal Features Using Deep Neural Networks to Improve Intrusion Detection', *IEEE Access*, vol. 6, pp. 1792–1806, 2018, doi: 10.1109/ACCESS.2017.2780250.

[28] Kumbhkar, M., Shukla, P., Singh, Y., Sangia, R. A., & Dhabliya, D. (2023). Dimensional Reduction Method based on Big Data Techniques for Large Scale Data. 2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS), 1–7. IEEE.

[29] S. Ahmad Khanday, H. Fatima, and N. Rakesh, 'Implementation of intrusion detection model for DDoS attacks in Lightweight IoT Networks', *Expert Syst. Appl.*, p. 119330, Nov. 2022, doi: 10.1016/j.eswa.2022.119330.

[30] S. Garcia, A. Parmisano, and M. J. Erquiaga, 'IoT-23: A labeled dataset with malicious and benign IoT network traffic', *Stratos. Lab Praha Czech Repub. Tech Rep*, 2020.

[31] 'sklearn.ensemble.ExtraTreesClassifier', *scikit-learn*. https://scikit-

learn/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html (accessed Mar. 16, 2023).

[32] 'ML | Extra Tree Classifier for Feature Selection - GeeksforGeeks'. https://www.geeksforgeeks.org/ml-extra-tree-classifier-for-feature-selection/ (accessed Aug. 12, 2022).

[33] 'sklearn.svm.LinearSVC', *scikit-learn*. https://scikit-learn/stable/modules/generated/sklearn.svm.LinearSVC.html (accessed Jun. 04, 2023).

[34] 'sklearn.naive_bayes.GaussianNB', *scikit-learn*. https://scikit-learn/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html (accessed Jun. 04, 2023).

[35] '1.11. Ensemble methods', *scikit-learn*. https://scikit-learn/stable/modules/ensemble.html (accessed Jun. 04, 2023).

[36] N. A. Stoian, 'Machine Learning for anomaly detection in IoT networks : Malware analysis on the IoT-23 data set', Jul. 03, 2020. https://essay.utwente.nl/81979/ (accessed Jan. 05, 2023).

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

[38] K. Thavasimani and N. Kasturirangan Srinath, 'Hyperparameter optimization using custom genetic algorithm for classification of benign and malicious traffic on internet of things–23 dataset', *Int. J. Electr. Comput. Eng. IJECE*, vol. 12, no. 4, p. 4031, Aug. 2022, doi: 10.11591/ijece.v12i4.pp4031-4041.

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

[40] I. Ullah and Q. H. Mahmoud, 'Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks', *IEEE Access*, vol. 9, pp. 103906–103926, 2021, doi: 10.1109/ACCESS.2021.3094024.

[41] Access 9:103906–103926. https://doi.org/10.1109/ACCESS.2021.3094024