Journal of Electrical Engineering, Electronics, Control and Computer Science – JEEECCS, Volume 10, Issue 35, pages 31-48, 2024

Machine Learning-Based Anomaly Detection in IoT Devices: A Development Approach

1st SEUN Mayowa Sunday
ADEY Innovations Limited
St. Modwen Park, Stonehouse,
Gloucester, United Kingdom
rightmayowa@gmail.com

Abstract— **Many experts have explored the risks posed by Internet of Things (IoT) devices to large corporations and smart cities. Because of the rapid acceptance of IoT and the nature of these devices, their inherent mobility, and the limits imposed by standardization, sophisticated systems capable of detecting suspicious movement on IoT devices connected to a network are necessary. As the number of Internet of Things devices connected to the Internet increased, so did the capacity of Internet traffic. As a result of this change, typical methods and traditional data processing approaches for detecting attacks are no longer valid and should be avoided. Because of the increased volume of network data, detecting assaults in the Internet of Things (IoT) and identifying malicious activity in its early stages is a particularly tough problem to tackle. This article proposes and offers evidence for an approach for identifying malicious network traffic. For identifying malicious network traffic, the framework employs three commonly used classification-based approaches. The Support Vector Machine (SVM), Random Forest (RF), and logistic regression (LR) algorithms all execute with 100% accuracy. The dataset Botnet-IoT was employed in the model creation used in this study framework, and the results in terms of training, specificity, and accuracy were compared.**

Keyword: IoT; LR; SVM; RF; Botnet-IoT

I. INTRODUCTION

The Internet of Things (IoTs) is a revolutionary computing paradigm that has evolved rapidly over the past decade in almost every technological domain. These include smart homes, smart industries, smart transportation, and smart healthcare ([1], [2]), the use of sensors ([3], [4]), smart cities [5], and satellites [6], to name a few. It is composed of a large number of Internet of Things devices (Things) that are outfitted with a variety of sensors, actuators, storage, computing, and communicational capabilities to collect and exchange data through the use of the standard internet [7]. Because of the sensitive nature of the data that is recorded and processed within the IoT network, it is imperative that the network be protected from any potential breaches.

As the first line of defense, various security mechanisms such as firewalls, authentication schemes, various encryption methods, antiviruses, and so on are currently used to protect sensitive data from potential security attacks on vulnerable devices ([8], [9], [10], [11]). Such example could be distributed denial of service (DDoS) attacks ([5], [12]). IoT can be implemented with software-defined networking (SDN), future network architecture, named data networking (NDN), and cloud computing network, along with voice over Internet Protocol (VoIP), deep learning (DL) and machine learning (ML) ([13][14][15][16][17][18][19]).

The utilization of a massive quantity of data results in the rapid production of many new anomalies, each of which may be original or may represent a mutation of an existing abnormality. Therefore, an intrusion detection system (IDS) that is capable of functioning as a second line of defense can give additional protection against security assaults to an internet of things (IoT) network. It is possible to categorize an IDS according to the technique of deployment and the detection approach. An IDS can be a host-based IDS or a network-based IDS based on its deployment; however, depending on the detection method, it can be signature-based, anomaly detection-based, specification-based, or hybrid detection [20]. An IDS can also be a network-based IDS based on its deployment. Providing security to the Internet of Things (IoT) at its entrance points is the primary emphasis of this research project. This will be accomplished by implementing network-based intrusion detection systems (NIDS) with an anomaly detection-based detection technique.

The increase in the False Alarm Rate (FAR) in detecting zero-day anomalies is the primary issue with the current generation of intrusion detection systems [20]. Researchers have lately investigated the prospect of applying machine learning (ML) and deep learning (DL) techniques to improve the accuracy of NIDS detection while simultaneously lowering the FAR. Studies have demonstrated that both ML and DL techniques are effective tools for learning valuable patterns from network traffic in order to categorize the flows as either anomalous or benign ([21]). The importance of the DL's application within NIDS for IoT networks is brought home by the fact that it has demonstrated effectiveness in extracting useful characteristics from raw data thanks to its in-depth architecture, which eliminates the need for human intervention.

Machine learning (ML), are an important type of AI algorithm that has received a lot of attention from academics in a variety of domains, including natural language processing, computer vision, and network security, amongst others. ML have fared particularly well in those sectors because of their dynamic system of prediction, which provides many abstractions for the purpose of efficiently learning complex characteristics ([22]). Due to the participation of a vast quantity of data created by IoT devices, ML has become a suitable methodology to be adopted for an IDS designed for an IoT network because of its qualities, which have made it an ideal candidate for adoption. The purpose of this research is to investigate whether it is possible to make use of ML to suggest an effective solution for NIDS while working within the context of IoT.

A. Aims And Objectives

The purpose of this research is to investigate whether it is possible to make use of ML to suggest an effective solution for NIDS while working within the context of IoT.

The aims of this research are:

- To implement a novel technique for anomaly detection in IoT-based smart devices.
- To demonstrate the significance of ML for developing data security for Smart homes.
- To assess how effective the proposed model is by evaluating it using the IoT-Botnet 2020 dataset, and then to compare the performance of the proposed model to the other works by previous researcher.

B. Research Questions

The study will attempt to answer the following questions:

- What are the key and perennial security issues in IoT-based smart environments?
- What are the recent and unique techniques for anomaly detection in IoT-based smart devices?
- What is the significance of ML in the development of a data security for IoT network?

C. Scope Of Study

The purpose of this research is to comprehend the concept of a machine learning (ML) method that may significantly increase the accuracy of an IoT network's anomaly detection.

D. Significance Of The Study

Predicting intrusions is essential for good decision-making and transparency, as a machine learning model, unrestricted by some of the assumptions of the standard statistical models, can provide far more accurate insights than a human analyst could deduce from the data. If a machine learning (ML) model can identify an intrusion on an Internet of Things (IoT) device that is not generally detected by conventional detection, then the likelihood of a protected IoT network flow increases, reducing the danger of cyberattacks.

E. Limitation Of Study

This study employs the Bot-IoT-2020 dataset for the creation of the model. Due to the recent distribution of this dataset, the number of available works is extremely limited. Consequently, the research relies on a small number of writers for comparisons of precision. Another drawback of this research is that, despite the careful development of the model, it will not be integrated into end-user apps like web, desktop, or mobile devices. Lastly, there are various machine learning techniques, this research will contribute to the knowledge domain by only considering the support vector machine (SVM), Logistic Regression (LR) and the Random Forest (RF).

II. Literature Review

With the exponential development of network usage, the importance of security has increased. A security mechanism protects data against security breaches. "The CIA triangle (Confidentiality, Integrity, and Availability) is the most common security service that any network can provide to defend against security breaches".

This section will examine ML and DL-based efforts on IDS. The chapter will finish by analyzing several datasets that have contributed to network intrusion detection and can be utilized for IoT over the year.

A. Accuracy Based Review

Accuracy is crucial to the creation of a model, the greater the accuracy, the better. This section examines past works and documents the accuracy.

Research by [23] aimed to improve IoT security by experimenting with anomaly detections on the IoT Network Intrusion Dataset using several machine learning algorithms. On the IoT Intrusion Network Dataset, they were able to attain both high accuracy and efficiency. Using KNN, they were able to achieve a 99% accuracy with an average runtime of 2 minutes. With an accuracy of 97% and a runtime of just 10.8 seconds, XGBoost demonstrated excellent results. According to the researchers, the F1 scores generated by several machine learning algorithms were consistent with accuracy. Using all of the features in the dataset, their preliminary experimental results showed significant promise as they seek to expand their work beyond binary classification to multiclass classification. Before training the model, the authors normalized the data to overcome the low accuracy of 86% that the LR technique provided.

The research conducted by [26], proposed an optimized ML-based framework that combined Bayesian optimization Gaussian Process (BO-GP) and decision tree (DT) classification model to detect botnet attacks on IoT devices. Their goal was to develop a dynamic, effective, and efficient IoT attack detection framework. Experimental results showed that their proposed optimized DT-based framework improved the accuracy, precision, recall, and F-score. More specifically, their highest result were able to achieved values of 99.99%, 0.99, 1.00, and 1.00 for these four metrics respectively. The authors concluded that the result illustrated that their proposed framework was both effective and robust in detecting botnet attacks in IoT environments.

According to [27], there has been an exponential increase in interest in Internet of Things (IoT) applications and services since their widespread adoption. Organizations have begun developing a variety of IoT-based products, ranging from personal devices such as smart watches to a network of smart grid, smart mining, smart manufacturing, and autonomous driverless vehicles. Various machine learning methods, including K-Nearest Neighbor (KNN), the Naive Bayes model, and the Multi-layer Perception Artificial Neural Network (MLP ANN), were utilized to construct a model in which data were trained using the BoT-IoT dataset. According to the author, their model addresses the security problem posed by bot threats. The author claimed the accuracy of the NB, KNN, and MLPANN to be 100%, 99.6%, and 87.4%, respectively.

The [29] proposed a real-time hybrid intrusion detection strategy in which the intrusion technique was used to detect well-known attacks and the anomaly approach was used to detect novel attacks. Using the anomaly detection technique, patterns of intrusion that avoided abuse detection were identified as attacks, resulting in a high detection rate. On the final day of the experiment, the model's accuracy reached an impressive 92.65%. Moreover, as the model learns and trains the system daily, the proportion of false negatives decreases drastically. The issue of slow detection rate persists when the model is applied to extremely big datasets.

Combining feed forward and pattern recognition neural networks, In addition [30] trained the IDS based on an artificial neural network with Bayesian regularization and scaled conjugate gradient training methods. Numerous performance criteria were utilized to evaluate the effectiveness and capacity of the proposed job. On the basis of the outcome, the two models were proven to outperform one another for a variety of attack detections. The overall accuracy of the feed-forward artificial neural network was 98.07%. The efficiency of the process can be improved by testing the model on multiple datasets.

B. Dataset Based Review

This section analyses and evaluates certain existing datasets, highlighting their characteristics, vulnerabilities, and downsides. These datasets have been utilized for both intrusion detection and the evaluation of anomaly-based intrusion detection machine-learning techniques. To compare these

datasets, the standards for establishing a state-of-the-art dataset are utilized.

The KDD Cup 99 dataset was generated using tcpdump for the 1999 Knowledge discovery in data mining competition. It is an enhanced version of the DARPA dataset that contains 41 features.

There are 78% redundant and 75% duplicate records in the training data and testing data, respectively. This results in unbalanced and biassed classification outcomes. In his research, [28] noted that the probability distribution in both the testing and training sets is highly variable, which may result in an imbalance between attack types and routine traffic. The biggest difficulty with the KDD99 dataset, according to [28], is that it contains obsolete attack types such as smurf, teardrop, and Neptune that are no longer prevalent in modern traffic patterns. Consequently, they are not updated to reflect the most recent assault trends and footprints. Despite these obstacles, numerous researchers utilize the dataset as a standard. While these methodologies and tactics improve the Intrusion Detection System's detection accuracy, they are not yet as accurate and effective in the actual world as they claim to be.

NSL-KDD: In 2009, NSL-KDD was modified from KDD'99 to address the issue of redundancy and duplication in the KDD'99 dataset. The training set consists of 21 known attack kinds, while the testing set has an additional 16 unknown attack types. After cleaning, the NSL-KDD training and testing data were decreased from 4,900,000 to 125,973 and from 2,000,000 to 22,544 records, respectively. This reduction provided a sufficient number of instances for experimentation without removing a chunk as was done in KDD99.

Research by [31], through their obtained results, they confirmed that supervised ML can be used to analyze traffic data and accurately expose the network that are maliciously over IoT devices. To identify that traffic accurately, they used the NSLKDD dataset to perform critical evaluation by applying ML techniques. NSLKDD dataset is used for the comparison of the given framework by employing functions such as selection and classification. Overall, the authors claimed that the RF algorithm provided the best accuracy of 85.34% on the fog layer in comparison with the SVM and GDBT of 32.38% and the 85.34% respectively.

The Canadian Institute of Cybersecurity Intrusion Detection System (CICIDS-2017) dataset was designed specifically for intrusion detection in 2017. SourceIP, SourcePort, DestinationIP, DestinationPort, and Protocol are some of the labels. CICIDS satisfies nearly all of the criteria for actual assaults and gives the most recent attack scenarios. However, 32] revealed throughout their research of the dataset that it had some significant flaws. Any standard IDS must handle these concerns to guarantee proper impartiality. This dataset is fairly extensive, spanning eight files, and represents five days' worth of Canadian Institute of Cybersecurity traffic data over the course of one year. This is likely one of the most significant flaws in the dataset, as it is arduous to analyse eight distinct files. The IDS might be designed as a single dataset, however this would generate a massive amount of data, hence increasing processing overhead. In addition, the dataset of 288,602 occurrences with no label and 203 instances with missing information contains numerous repetitive records. This redundancy renders the data unsuitable for IDS training [32].

MAWILab: The dataset is tagged from the MAWI archives to identify anomalies, hence the name. The MAWI archives capture 15 minutes of daily traffic every day. The dataset is acquired using tcpdump on a network testbed. MAWILab lacks ground-truth information. However, the data classification is divided into three categories: notice, suspicious, and abnormal. Notice is the label provided to data that is deemed abnormal without consensus from all anomaly detectors, suspicious appears to be anomalous, and anomalous is indeed anomalous.

In her research, [33] noted that the primary constraint of the MAWILab dataset is that the packet trace may only be accessed for 15 minutes every day. In addition, whether data is labelled positive or negative relies on the classification methods utilized and the frequency with which they generate false positives. [33] created the UGR'16 dataset using these. The packet trace was separated into calibration and test portions. The lengthy calibration capture captures only the actual background traffic. The test capture consists of both authentic background traffic and synthetic traffic containing common attack types. The purpose of the test capture is to determine whether the IDS will detect the attacks. To evaluate the accuracy of the detectors at varying hours and days, a batch of attacks is carried out at various times.

This dataset was intended to give users with an acceptable and standard dataset, with the creators of UGR-16 hoping to address the issue of representative sample. It accomplishes this by capturing various hues and statuses of a four-month network's traffic at various times and days to simulate a real huge network. They mimicked contemporary network

traffic with simulated cyberattacks. UGR16's data set consists of real and synthetic v9 netflow data collected by sensors within a tier-3 ISP. ISP is a cloud service provider that runs virtualized services such as WordPress, Joomla, and email. During the massive capture, an estimated 600 million external IP addresses, 10 million corresponding subdomains, and 16 billion individual data packets are inspected [34].

Cyber-attack prevention and detection skills have been hampered because of the vast number of IoT devices, their diverse nature, and their limited resource availability. Due to these qualities, monitoring Internet of Things devices at the device level is not practicable; instead, monitoring occurs at the network level. Because of this, anomaly detection is in a better position to safeguard the Internet of Things network. Anomaly detection is regarded as an essential tool for the protection of the system because it assists in locating and notifying of anomalous activity within the system. Anomaly detection methods in information technology and internet of things have been modified to make use of machine learning. On the other hand, the implementations of anomaly detection systems that make use of machine learning in IT systems have been more successful than those in the IoT ecosystem due to their resource capabilities and their location inside the perimeter. Despite this, the machine learning-based anomaly detection that is now in use is susceptible to attacks from adversarial systems. In this context, [29] offered a complete assessment of anomaly detection in the IoT system using machine learning. A discussion of the significance of anomaly detection, the difficulties associated with the development of anomaly detection systems, and an examination of the machine learning methods that were applied is presented. The authors has advocated that blockchain technology can be employed to prevent model corruption by adversaries in situations where IoT devices can cooperatively build a single model using blockchain consensus methods. This is in accordance with the findings of the authors of the study.

C. Challenges In The IOT Based ML

Enhanced Attack Detection Connecting billions of electronic devices (e.g., sensors) and machines to IoT systems. This connectivity trend is projected to persist long into the foreseeable future, particularly with respect to wireless IoT applications such as smart cities. Each piece of equipment added to the network has various opportunities to serve as a Zero-day attack vector. Given the number of network entry points, machine learning algorithms must be able to train continually and adaptively in order to thwart these predicted attacks.

The present ML paradigm entails two steps: first, a specific training dataset executes a machine-learning algorithm to build a model, and then the model is applied to its intended IoT application ([24, 25]). The process of continuous ML is still in its infancy, so applying continuous learning to IoT presents obstacles.

The research by [24], the authors demonstrate lifelong ML by considering systems that can learn various tasks from multiple domains throughout their lifetime. The purpose is to assimilate sequentially acquired knowledge in order to selectively apply it when learning a new scheme in order to build more refined assumptions and/or policies. In addition, the authors advise that the AI community should study more seriously the nature of systems that can learn throughout the course of their operational lifetime, rather than focusing solely on one-time learning algorithms. In the future, it will be necessary to explore how to implement lifetime machine learning to detect cyber intrusions in IoT. Data Characteristics of IoT. To train and produce an accurate model, ML algorithms rely on data. Since data constitute the foundation for extracting knowledge, it is essential to ensure their accuracy. Data quality, availability, and integrity are crucial factors in the training and testing of AI algorithms. However, IoT systems generate huge volumes, high velocities, and diverse data varieties, sometimes known as the 3 Vs. Various types of equipment and devices are employed to generate data, resulting in heterogeneous data. Consequently, ensuring data authentication in IoT is difficult. Nevertheless, machine learning techniques offer considerable promise for addressing these IoT security issues.

D. Conclusion

Several efforts have been conducted in network attack over the past few years, as demonstrated by the previous review. There have been a significant number of high accuracy developments. As stated previously, the development dataset has significant limitations. [35] argued that the obvious limitations are the outdated datasets, their inability to detect a wide range of network attacks, most of them are not publicly available, and that most of them do not take into account the importance of message encryption in modern communication make it difficult to evaluate the majority of the datasets. The IoT-Botnet2020 dataset is the most recent state-of-the-art dataset for

IoT network intrusion detection. Although this dataset is new, our study will contribute by constructing three different ML models based on the IoT-Botnet2020 and comparing their accuracy to that of other models by previous researchers.

III. METHODOLOGY

Discussion of the features of the dataset used for this project, performances of the exploratory data analysis (EDA), feature selection, measuring metrics, model construction and visualization are discussed in this section.

A. Dataset

Machine learning (ML) is successful due to the availability of datasets, and this research is no exception. The dataset utilized for this study is the Bot-IoT-2020 dataset [28]. The definition of the dataset's characteristics is tabulated (table 3) as presented in the official website [28].

B. Bot-IoT-2020 Dataset

In the Cyber Range Lab at UNSW Canberra, a realistic network environment was designed to create the BoT-IoT dataset. The network environment consisted of both regular and botnet traffic. The source files for the dataset are supplied in various formats, including the original pcap files, the generated argus files, and csv files. The data were divided depending on attack category and subcategory to facilitate the labelling procedure.

The size of the recorded pcap files is 69.3 GB, and they contain over 72,000,000 records. The size of the captured flow traffic in csv format is 16.7 GB. The dataset consists of DDoS, DoS, OS and Service Scan, Keylogging, and Data exfiltration attacks, with DDoS and DoS attacks further categorised by protocol [28].

However, for academic purpose, the author created 5% of the whole dataset which consists of the approximately 3 million records. The author also selected best top 19 features out of the total 47 features which were used to gather the 5% of the dataset. Table 1 shows the definition of the used features according to the author.

No.	Feature	Description
1	pkSeqID	Number of row
2	proto	Textual representation of transaction protocols included in network flow.
3	Saddr	IP Source
4	Sport	Port of source
5	Daddr	Destination IP
6	Dport	Destination port
7	Seq	Argus sequence number
8	stddev	Agr. Standard deviation
9	N_IN_Conn_P_SrcIP	Total number of connections per IP source.
10	min	Minimum duration of aggregated records
11	state_number	Statistical representation of feature state
12	mean	Average length of a compiled record
13	N_IN_Conn_P_DstIP	Connections to destinations per IP.
14	drate	destination-to-source packets per second
15	srate	Source-to-destination packets per second
16	max	Maximum aggregated record duration
17	category	Category of traffic
18	subcategory	Subcategory of traffic
19	attack	When the traffic is normal=0; attack traffic = 1

Table 1: Bot-IoT Dataset

C. Exploratory Data Analysis (EDA)

To enhance the performance of a machine learning (ML) model, "the dataset must be organized so that the model can utilized it easily" [36]. EDA is essential in ML since it allows to interpret our dataset before pre-processing. This section explains the EDA and the data pre-processing procedures utilized in this research.

Figure 1 shows that the dataset does not contain null values while figure 2 shows that the dataset is highly imbalanced. As explained by the author of the dataset (table 3), most of the traffic are attack traffic (1) while very few are normal traffic (0).

D. Feature Selection And Preprocessing

Since feature selection is performed as a pipeline step in ML, it is essential to model development. Depending on the characteristics of the data, this process may be manually, automatic, or both. Major

advantage of the technique is the ability to select a collection of attributes that help to the improvement of models and forecast output [37]. This phase is required since it has a substantial impact on the model's development time and precision. "Irrelevant features in the dataset can be detrimental to training by allowing the model to learn from data unrelated to the projected output; hence, feature selection is necessary". With accuracy, the effect is readily apparent, as undesired data serves as background noise. The benefits of feature selection include "minimal overfitting, accuracy improvement, and training time reduction".

Several aspects in the dataset were eliminated using feature selection. This is commonly misconstrued as dimensionality reduction [37]. Dimensionality reduction techniques frequently combine characteristics to minimize their dimensions, whereas feature selection selectively removes features without altering the remaining ones [38]. In this research, two distinct methodologies for selecting features are described in the following subsections.

E. Manual Feature Selection

According to the author of the dataset, the 5% data utilised shows that they reduced the features to only 19 out of the total 47 features. However, inpecting the 19 features in table 1, it shows that there are some features that can have the same effect on both normal traffic and attack traffic.

The pkSeqID,saddr,daddr,proto,subcategory,dport And sport, were removed since they have equal effect on both normal or abnormal traffic (figure 4).

F. Automatic Feature Selection

In this instance, the Mutual Information (MI) algorithm is utilized. How dependent and independent variable are connected is obtained [39].

Figure 5 illustrates the outcome of the MI utilizing the scikit learn library.

The first step was generation of MI (figure 5) then features of lower importance (0.000) to the dependent variable (attack) were dropped. Total used features dropped to 5 (figure 6) with over 3 million entries (figure 7).

G. Changes in Dataset

As shown in Figure 3, category is the categorical variable, which must be converted into the numerical variable. Using pandas built-in library, the feature is converted into numerical variable which later increased the number of the features as shown in Figure 8. After the categorical variable is converted to a numeric variable, feature scaling is applied to the dataset to assist model creation and prevent one independent variable from dominating another [36]. Random over sampling techniques is used to fix the imbalanced dataset in this research [40].

H. Train Test Split

Splitting the dataset into training and test set is very important in ML development. It is recommended to use very high percentage of the data for training and use the remaining few percentage for testing. Testing data is what is passed to evaluate the model for good performance. 80% of the dataset is used for training while the remaining 10% is used for testing [41].

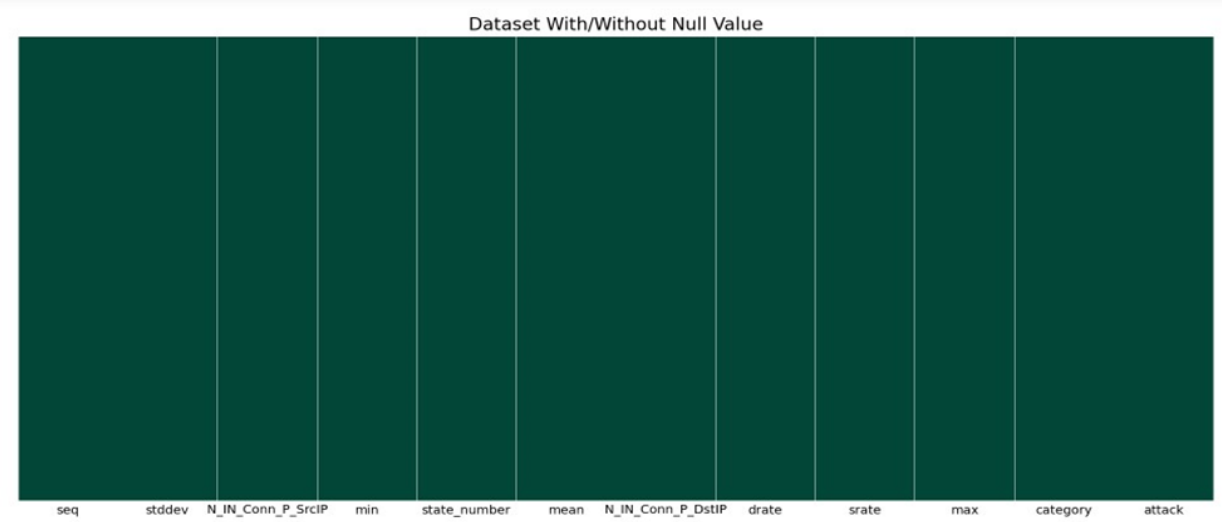

Figure 1: Dataset null value

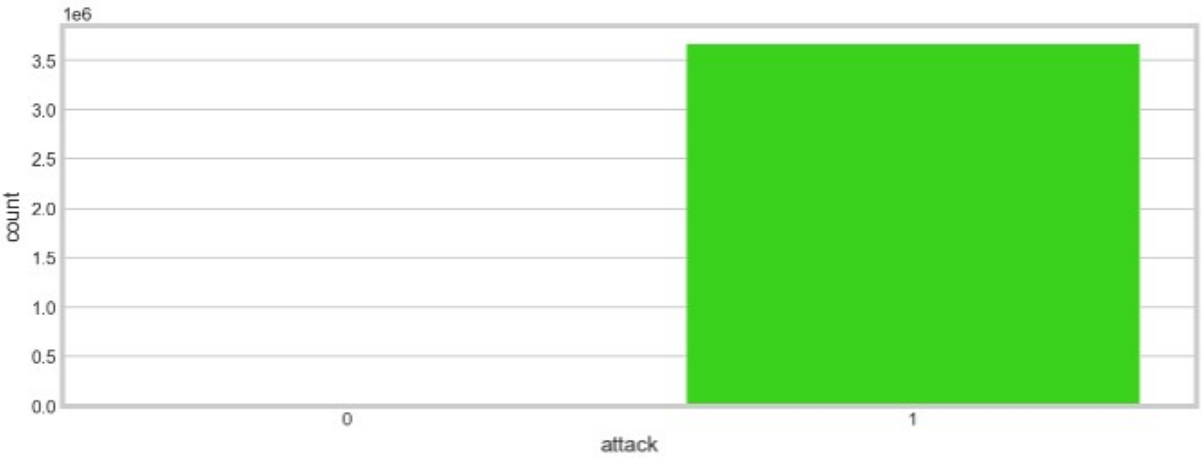

Figure 2: Imbalanced dataset

stddev	N_IN_Conn_P_SrcIP	min	state_number	mean	N_IN_Conn_P_DstIP	drate	srate	max	DoS	Normal	Reconnaissance	Theft	attack
068909	75	0.000000	1	0.068909	96	14.511893	0.566862	0.137818	1	0	0	0	1
000000	2	0.000131	2	0.000131	1	0.000000	0.000000	0.000131	1	0	0	0	1
064494	75	0.000000	1	0.064494	96	15.505319	0.567549	0.128988	1	0	0	0	1
064189	75	0.000000	1	0.064189	96	15.578993	0.567570	0.128378	1	0	0	0	1
063887	75	0.000000	1	0.063887	96	15.652637	0.567630	0.127774	1	0	0	0	1

Figure 3: Features after fixing the categorical variable.

I. Random Forest (RF)

Figure 9 depicts the random forest technique, a "supervised learning system that can be used for classification and regression". However, it is typically employed for categorization jobs. A forest is composed of trees, and a forest with more trees is stronger (figure 9). Similarly, the RF method constructs DTs from data samples, obtains predictions from each, and then votes on the best option.

In contrast to the more common practice of bagging, it is anticipated that the merged trees will be de-correlated in order to make it possible for each tree to produce decisions that are more streamlined and specialized. The implementation of random feature splits for each decision tree is the standard method for accomplishing this de-correlation [42]. If there is a significant degree of correlation between the trees, then a single decision tree can be used as a suitable substitute for the random forest. This model also includes an approach for quantifying the importance of variables based on the estimations or assessments of the feature space provided by the various trees contained within the model [43]. The random forest method, similar to decision trees, requires fine-tuning of a number of hyper-parameters in order to achieve the best possible equilibrium between classification

RF pseudo-code:

- Step 1 – select randomly chosen sample from the dataset.
- Step 2 – get the prediction result from every tree
- Step 3 – perform voting for every predicted result.
- Step 4 – most voted prediction is the final prediction result.

J. Regression

This is one of the most straightforward methods for modelling relationships between variables. The objective of the linear regression technique is to create a line that best represents the connection between the predictors and the dependent variable. The linear regression for a single predictor is formally given by equation 1.

$$y = B_0 + B_1 x \quad (1)$$

Logistic regression (equation 3.2) is an approach for supervised learning that is applicable when the dependent variable is dichotomous (binary). In contrast to linear regression, real-world problems frequently necessitate nonlinear models. Real-world problems can be quadratic, exponential, or logistic. Logistic regression means that potential outcomes are categorical rather than numerical [47]. Through logistic regression, we may predict categorical outcomes, such as "yes or no it is dangerous traffic" or "0 or 1 will be dangerous traffic." In reality, in the context of this study, decision-making frequently simplifies down to a simple yes or no. In logistic regression, we may make far more fundamental predictions, such as "will this traffic be dangerous at all? Using scikit-learn library, L2 regularization with SAGA approach was used for the gradient descent.

K. Support Vector Machine (SVM)

SVM is used to examine classification and regression patterns [44]. A SVM generates a classification model for new data. Given a set of labelled training data with one or two categories, it transforms into a non-probabilistic binary linear classifier. It is used to solve classification issues [44]. A SVM model is defined as "a representation of the samples as space points that has been mapped so that samples of different categories can be separated by a simple, as large as possible distance [45].

This algorithm plots each data point as a point in n-dimensional space (where n is the number of features), with the value of each feature corresponding to a specific coordinate. Using the scikit-learn package, create an arrangement by identifying the hyper-plane that most effectively divides the two classes (dependent and independent variables).

L. Evaluation Metrics

Well-known measuring metrics such as recall, accuracy, f1-score, roc-curve, etc. will be used [43].

M. Confusion Matrix

Confusion matrix is used to understand what the model is getting correctly and what it is getting wrongly. Figure 10 shows how confusion matrix generally works then table 5 and figure 7 shows the confusion matrix of all the proposed models.

True Positive (TP) and True Negative (TN): Real value at the diagonal axis

False Negative (FN): Other values along the horizontal

False Positive (FP): Other value present in the vertical column

N. Classification Report

This is the recall, precision, accuracy, f1-score, and support as explained below.

- **Recall**: The TP divided by how many times the classifier predicted that class.

$$\frac{TP}{TP + FN} \quad (2)$$

- **Precision**: Number of correct predictions divided by how many occurrences of that class were in the test data.

$$\frac{TP}{TP + FP} \quad (3)$$

- **F1-score**: The weighted harmonic mean of the precision and recall values for the test is the F1-score. A high f1-score indicates that the precision is more balanced [46].

$$\frac{2 * Precision * recall}{precision + recall} \quad (4)$$

- **Support**: The total number of true response samples in the class.

$$TP + FN \quad (5)$$

- **Accuracy**: Combination of TP and TN divided by other values [47].

$$\frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

O. Visualization Techniques

The discussion of area under the curve (AUC) and the receiver operating character (ROC) will be discussed here.

- **Area under Curve (AUC) and Receiver Operating Character (ROC)**

Both of these terms refer to performance statistics that can be used to detect issues over a spectrum of threshold levels. The ROC curve is defined as "the graphical depiction of the degree or measure of separation" from the threshold (figure 5), whilst the area under the curve (AUC) refers to the mathematical formula for calculating the value (table 5). This demonstrates that the model is able to differentiate between different types [46]. The model is considered to be of greater quality when its AUC value is higher. In a similar vein, the model's ability to differentiate between fake and legit networks improves in direct proportion to the model's [36]. When the area under the curve (AUC) gets close to one, a model is excellent since it indicates a high degree of separability. If the AUC value of a model is very close to zero, which indicates that the model has the lowest possible measure of separability, then the model is thought to be incorrect. In point of fact, this suggests that the effects are mutually reinforcing. It refers to the act of incorrectly recognizing the digits 0 and 1 as being each other. A value of 0.5 for the AUC indicates that the model is utterly unable to distinguish between classes [46].

IV. FINDINGS

Python, together with its associated frameworks, was the language of choice for all of the work that was put into practice. This section will begin with a review of the programming language, programming framework, and integrated development environment (IDE) that were utilized in this study. Documenting the most important findings from the experiment is the final step in completing this part.

A. Programming Language And Integrated Developemen Environment (IDE)

Data scientists overwhelmingly favor using Python as their primary programming language of choice. It is equipped with a number of libraries that make it simple to construct machine learning and deep learning applications. The Python programming language and its associated libraries is heavily used during the course of this research.

	seq	stddev	N_IN_Conn_P_SrcIP	min	state_number	mean	N_IN_Conn_P_DstIP	drate	srate	max	category	attack
0	9	0.068909	75	0.000000	1	0.068909	96	14.511893	0.566862	0.137818	DoS	1
1	10	0.000000	2	0.000131	2	0.000131	1	0.000000	0.000000	0.000131	DoS	1
2	11	0.064494	75	0.000000	1	0.064494	96	15.505319	0.567549	0.128988	DoS	1
3	12	0.064189	75	0.000000	1	0.064189	96	15.578993	0.567570	0.128378	DoS	1
4	13	0.063887	75	0.000000	1	0.063887	96	15.652637	0.567630	0.127774	DoS	1

Figure 4: Remaining features after manual feature Selection

```
<matplotlib.legend.Legend at 0x24b146fb438>
```

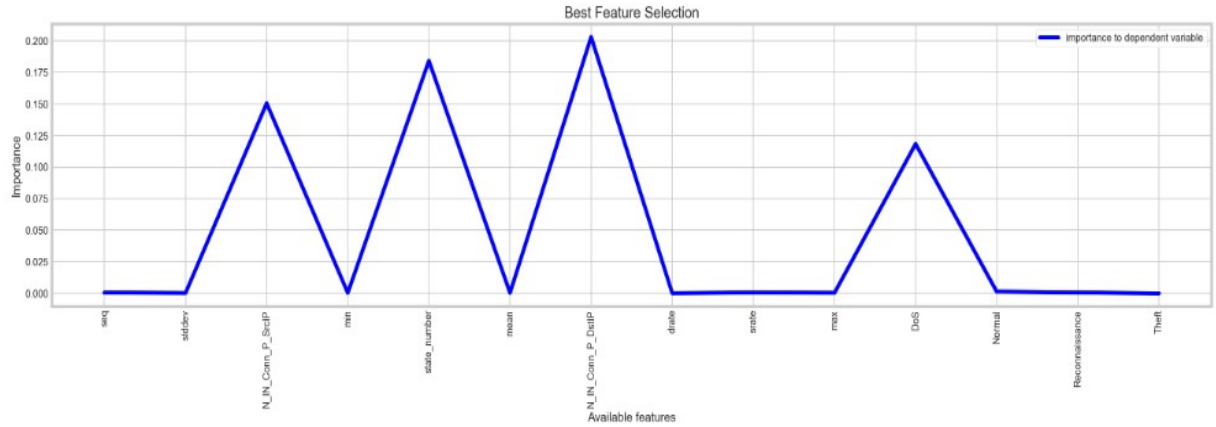

Figure 5: Finally Used features.

	N_IN_Conn_P_SrcIP	state_number	N_IN_Conn_P_DstIP	DoS	attack
0	75	1	96	1	1
1	2	2	1	1	1
2	75	1	96	1	1
3	75	1	96	1	1
4	75	1	96	1	1

Figure 6: Feature selection based on mutual information

B. Framework

Pandas: Pandas is a data loading, evaluating, and mining tool that is used to get a better understanding of the data. In addition to this, it is utilized to organize the data in a fashion that makes it suitable for both machine learning and deep learning.

NumPy: NumPy is frequently utilized with pandas in order to effectively view multi-dimensional arrays and carry out a variety of mathematical operations on them.

Scikit-learn: This helps to make it easier to construct a wide variety of different methods for regression, classification, and clustering. In addition to that, we are able to carry out measuring metrics like classification reports, roc curves, and confusion matrix with its assistance.

Seaborn: Packages for data visualization such as Seaborn make use of an intricate user interface to produce statistical images that are both visually appealing and educational.

Matplotlib: Matplotlib is a program that can plot either a 2-D or 3-D array.

Shap: Shap is a tool that is utilized to gain an understanding of the significance of features. It is not available in any other models than the ensemble ones (random forest). In addition to this, it illustrates how the model incorporates the independent variable that is subject to classification.

C. Integrated Development Environment (IDE)

IDE is a development environment for software. There are a variety of IDEs available for various reasons. Popular among data scientists, the Anaconda software is utilized in this study. Jupyter Notebook which is the developers interactive computing tools, open standards, and services in many programming languages is majorly used in the Anaconda software.

V. Experimental Results

The model's classifications results are shown in table 2.

Model	Precision (%)	Recall (%)	F1-Score (%)	Support (%)	Accuracy (%)
RF	100	100	100	917131	100
LR	100	100	100	917131	100
SVM	100	100	100	917131	100

Table 2: Classification report

- **LR** = Logistic Regression
- **SVM** = Support Vector Machine
- **NB** = Naïve Bayes
- **KNN** = K-nearest neighbor
- **MLPANN**= multi-layer perceptron artificial neural network
- **DT** = Decision Tree
- **Opt. DT** = Optimized Decision Tree
- **MM** = Measuring metrics
- **PM** = Proposed Model
- **RF** = Random Forest

Model	Test Data			
	TP	FP	TN	FN
RF	133	0	916998	0
LR	133	0	916998	0
SVM	133	0	916998	0

Table 3: Summary Of The Confusion Matrix

A. Comparison Of Developed Models With Previous Ones

MM		Acc	AUC
PM	RF	100	100
	LR	100	100
	SVM	100	100
[27]	NB	100	61
	KNM	99.6	99.2
	MLPANN	87.4	47.1
[26]	DT	99.8	-
	SVM	88.4	-
	Opt. DT	99.99	-

Table 4: Comparison with previous work

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3668522 entries, 0 to 3668521
Data columns (total 5 columns):
 #   Column              Dtype
---  ------              -----
 0   N_IN_Conn_P_SrcIP   int64
 1   state_number        int64
 2   N_IN_Conn_P_DstIP   int64
 3   DoS                 uint8
 4   attack              int64
dtypes: int64(4), uint8(1)
memory usage: 115.5 MB
```

Figure 10: Number of remaining entries

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3668522 entries, 0 to 3668521
Data columns (total 15 columns):
 #   Column              Dtype
---  ------              -----
 0   seq                 int64
 1   stddev              float64
 2   N_IN_Conn_P_SrcIP   int64
 3   min                 float64
 4   state_number        int64
 5   mean                float64
 6   N_IN_Conn_P_DstIP   int64
 7   drate               float64
 8   srate               float64
 9   max                 float64
 10  DoS                 uint8
 11  Normal              uint8
 12  Reconnaissance      uint8
 13  Theft               uint8
 14  attack              int64
```

Figure 9: Confusion matrix (Source: Author)

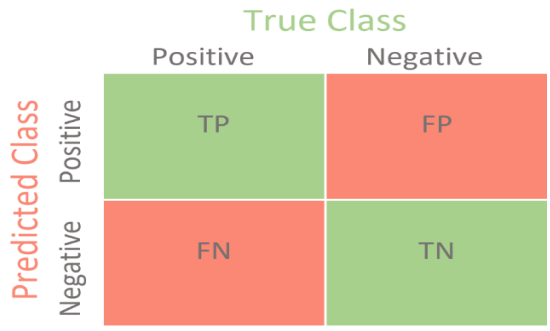

Figure 8: Confusion matrix (Source: Author)

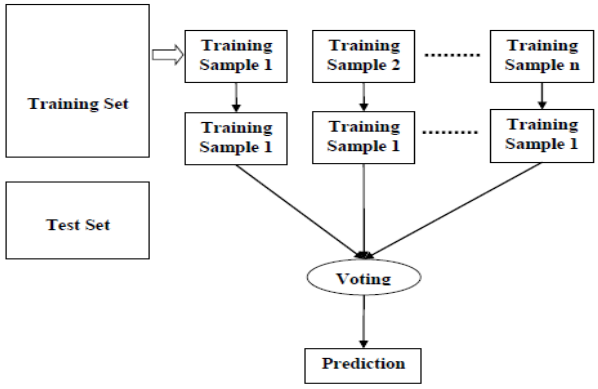

Figure 7: Random Forest Classifier (Author)

From the table 2, there is documentation of the classification report of random forest (RF), support vector machine (SVM), k-nearest neighbor (KNN) and the artificial neural network (ANN).

The precision, recall and the f1-score of the models have an accuracy of 100%. Table 3 tabularize the confusion matrix shown in figure 11. From the table 3, RF, SVM, and the LR have 0 FP and the FN, which shows good model performance [43]. All the model predictions fall under both false negative (FN) and true positive (TP) which is very good indication of the experimental accuracy.

In general, a score of 100% in any evaluation metric can be seen as an indication of overfitting. However, in the context of this study, a score of 100% is a result of the highly imbalanced nature of botnet datasets. In the vast majority of the datasets, the non-bot class is represented by the overwhelming majority of the devices [27]. When testing on some datasets, there is only one instance of the bot class, which can result in either 0% or 100% precision depending on the circumstances. However, using different techniques which attempts to fix the imbalanced dataset problems, the author confirms the performance of the model using confusion matrix (figure 11).

B. Confusion Matrix

As explained earlier, confusion matrix is a very important technique which can help to know if there is problem of overfitting in the model. The model's confusion matrix is depicted in figure 11.

C. Auc And Roc Curve

The rock curve (figure 12) and table 5 shows that all the models are performing greatly with amazing score, hence, the overlap of the result as shown in figure 4 below.

- **RF** = Random Forest
- **LR** = Logistic Regression
- **SVM** = Support vector machine

MODEL	**AUC (%)**
RF	100
LR	100
SVM	100

Table 5: Auc Score Based Of The Models

The report of table 5 shows that the AUC result of RF, LR, and SVM, are all 100%. The next chapter critically discuss these findings.

VI. Discussion

Because of the quick expansion and development of self-sufficient, energy-aware sensing devices, the proliferation of the Internet of Things (IoT) has impacted practically all of our day-to-day applications. This is a direct effect of the rapid growth and development of intelligent systems. The inherent limitations of IoT devices in terms of computational power, storage capacity, and network access have contributed significantly to the rise in IoT-based botnet assaults, despite the fact that these kinds of attacks have become more common.

In order to protect against vulnerabilities in the Internet of Things, this research proposes using three different models. These models include logistic regression, support vector machines, and random forests. As shown in section 3.2, there are a few different attacks that can be carried out on an Internet of Things device; however, the Denial of Service (DoS) attack is the most important one to consider followed by feature selection. DoS attacks can be defended against by incorporating the model that performs the best into the consumer application. This allows the application to identify whether or not an attack is actually taking place (Figure 13).

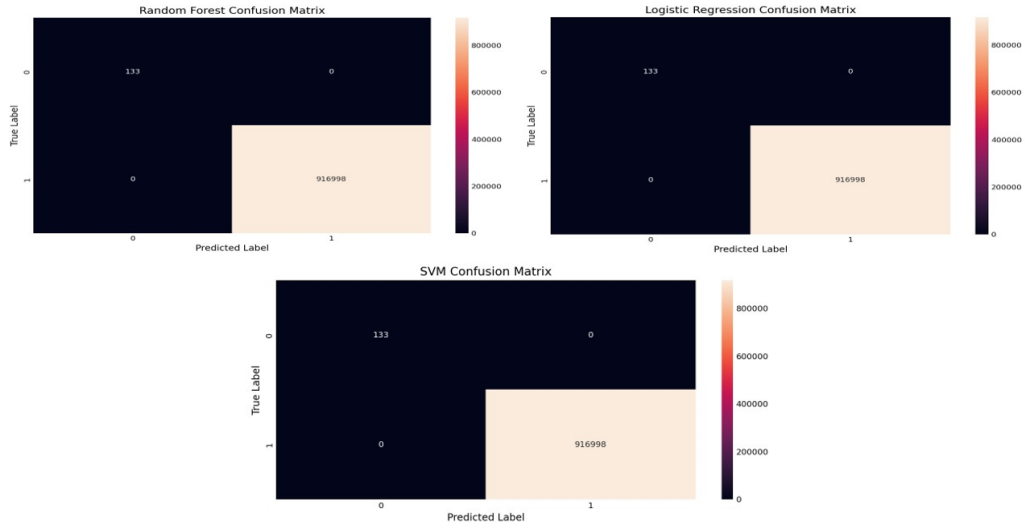

Figure 12: The confusion matrixes of logistic regression model

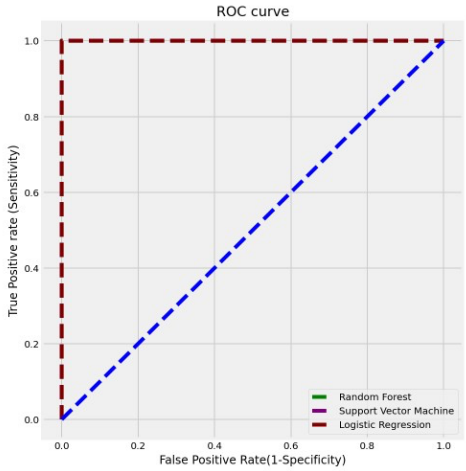

Figure 13: ROC curve on test data

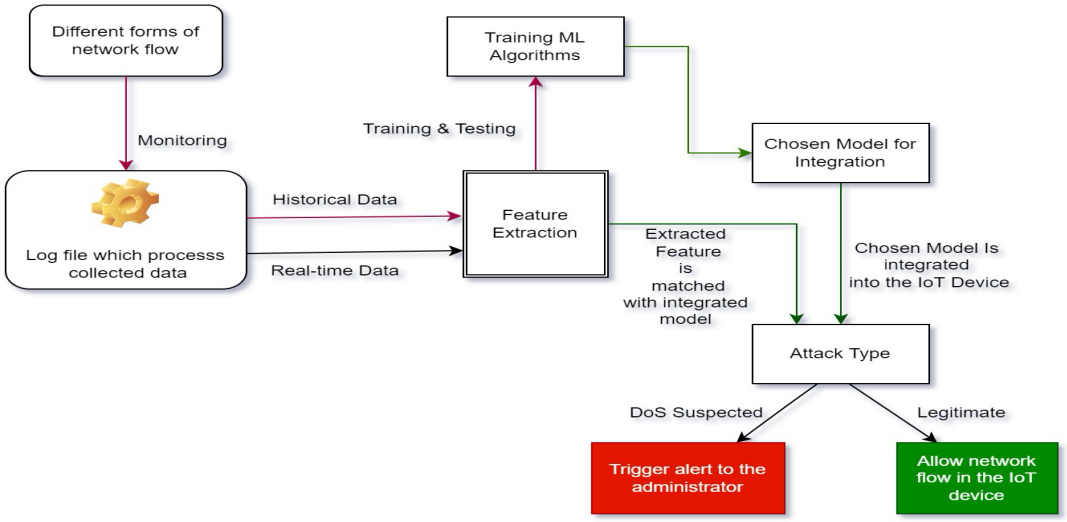

Figure 11: How the ML model will be integrated for Network Threat Detection

Different types of networks are supported by IoT devices and are recorded in the device's log file, as shown in Figure 13. In addition to serving as a place to store information, log files are also used as a means of collecting data. The log file not only keeps track of the actual data being collected, but also of the datasets that were used. To make sure the data fits the extracted features of Figure 6, the author employs those characteristics to make a feature selection (extraction). All machine learning models have been trained and tested as planned. The confusion matrix shown in Section 4.2 of this study demonstrate that all three methods (RF, SVM, and LR) achieved 100% accuracy. As a result, if simply accuracy is a concern, any of the models can be selected. The user's browser, phone, or computer all work as entry points for the ML model end-user integration. Integrations can be used in a variety of contexts. Figure 13 shows how the integrated model uses the dependent variable to determine which types of network attacks (DoS vs. legitimate) should be banned or allowed.

After model development, it is important to compare the result of proposed model with other existing works in the similar domain.

An improved ML-based framework was proposed by [26] to identify botnet assaults on Internet of Things items; it uses a combination of a Bayesian optimization Gaussian Process (BO-GP) and a decision tree (DT) classification model. The team's mission was to create a system capable of detecting Internet of Things (IoT) attacks in real time. Experimental results showed that the new version of their DT-based architecture improved upon the previous version in areas such as accuracy, precision, recall, and F-score. For these four indicators, their best results were, respectively, 99.99%, 0.99%, 1.00, and 1.00. The researchers concluded that the results supported the efficacy of their proposed strategy for detecting botnet assaults in IoT devices.

This research has a major contribution towards the improvement of security of IoT device via the development of three different machine learning models which are logistic regression (LR), random forest (RF) and the support vector machine (SVM). The result presented in this research has shown amazing accuracy when compared with the works of other researchers that utilize the same dataset over the year. As pointed out earlier, the major success of this can be attributed to great commitment in the necessary steps for data preparation such as exploratory data analysis (EDA), feature selection techniques and the feature engineering techniques as documented in this research.

VII. Conclusion And Recommendation

ML has garnered a great deal of attention from researchers working in a wide variety of application fields. ML can process complicated data and automatically extract raw features without requiring any prior expertise.

As a consequence of these findings, model development in the field of machine learning was possible. This research presented the model development based on the random forest (RF), logistic regression (LR) and the support vector machine (SVM). Before the model development, there was relevant data-preprocessing which identifies any form of missing or null value, in this stage the structure of the dataset was also identified. Exploratory data analysis (EDA) was the next step, which enables the author to be able to identify the relationship between the dependent and the independent variables. The balanced and the imbalanced dataset was also identified in this stage. This enables the need for feature engineering techniques to fix the issue of the imbalanced dataset and the normalization techniques to fix the challenges of the bias choice based on the big entry size differences.

From the proposed models, this experiment was able to get the accuracy of 100% for SVM, RF and the LR models. It is true that overfitting or error due to approximation can lead to the poor accuracy of the model, the author confirmed the accuracy by checking the consistency with the confusion matrix which shows that the false negative (FN) and the false positive (FP) are both zero.

VIII. Recommendation

For the recommendation on this domain, big data analytics is becoming the important technology as everyday life has been documented on one data or the other. However, machine learning approach has been major concern due to their failure on very large dataset. For example, the dataset used in this research is only 5% of the total size of the actual dataset. The author of the dataset provided the 5% size for the academic purpose which is suitable for the machine learning model. However, the full dataset contains over 75 million entries which can be an example of big data. However, using this kind of data for ML is not feasible. Hence, there is a need for the deep learning (DL) technology is there is an attempt to use the full dataset for model development.

Although there is tremendous improvement in the proposed model development, this research can still be improved further by comparing the computational time with other ML technologies. This will enable an organization to choose the best model when time performance is of important over accuracy.

These findings also suggested that merging DL with big data or cloud-based technologies can be important to the IoT network threat detection. This is in addition to optimization techniques, ensemble methods, and other machine learning algorithms which are already in use. In light of the findings, it is expected that this study will prove to be an invaluable reference for those working in the field of cybersecurity research and development.

As pointed out earlier, the availability of datasets is the key to the success of ML models, and the larger the dataset, the better. However, the hybrid ML which combines different ML models in stack for better model performance can be used to improve the fastness and other hidden features of the model.

Lastly, after the model development, integration into the end user consuming applications can be the next step. Future work recommends integration of this model as the plugin into the IoT-related device so that it can be used for real-time IoT device related threat detection.

REFERENCES

[1] Harb, H.; Mansour, A.; Nasser, A.; Cruz, E.M.; Diez, I.D.L.T. A Sensor-Based Data Analytics for Patient Monitoring in Connected Healthcare Applications. *IEEE Sens. J.* 2021, *21*, 974–984. [CrossRef]

[2] Haider, I.; Khan, K.B.; Haider, M.A.; Saeed, A.; Nisar, K. Automated Robotic System for Assistance of Isolated Patients of Coronavirus (COVID-19). In Proceedings of the 2020 IEEE 23rd International Multitopic Conference (INMIC), Bahawalpur, Pakistan, 5–7 November 2020; pp. 1–6.

[3] Sarkar, N.I.; Kuang, A.X.-M.; Nisar, K.; Amphawan, A.; Sarkar, N.I. Performance Studies of Integrated Network Scenarios in a Hospital Environment. *Int. J. Inf. Commun. Technol. Hum. Dev.* 2014, *6*, 35–68. [CrossRef]

[4] Chowdhry, B.; Shah, A.A.; Harris, N.; Hussain, T.; Nisar, K. Development of a Smart Instrumentation for Analyzing Railway Track Health Monitoring Using Forced Vibration. In Proceedings of the 2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT), Tashkent, Uzbekistan, 7–9 October 2020; pp. 1–5.

[5] Haque, M.R.; Tan, S.C.; Yusoff, Z.; Nisar, K.; Lee, C.K.; Chowdhry, B.; Ali, S.; Memona, S.K.; Kaspin, R. SDN Architecture for UAVs and EVs using Satellite: A Hypothetical Model and New Challenges for Future. In Proceedings of the 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2021; pp. 1–6.

[6] Ahmad, F.; Ahmad, Z.; Kerrache, C.A.; Kurugollu, F.; Adnane, A.; Barka, E. Blockchain in Internet-of-Things: Architecture, Applications and Research Directions. In Proceedings of the 2019 International Conference on Computer and Information Sciences (ICCIS), Aljouf, Saudi Arabia, 3–4 April 2019; pp. 1–6.

[7] Mehmood, Y.; Ahmad, F.; Yaqoob, I.; Adnane, A.; Imran, M.; Guizani, S. Internet-of-Things-Based Smart Cities: Recent Advances and Challenges. *IEEE Commun. Mag.* 2017, *55*, 16–24. [CrossRef]

[8] Ahmad, Z.; Khan, A.S.; Shiang, C.W.; Abdullah, J.; Ahmad, F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Trans. Emerg. Telecommun. Technol.* 2021, *32*, 4150. [CrossRef]

[9] Apruzzese, G.; Andreolini, M.; Marchetti, M.; Colacino, V.G.; Russo, G. AppCon: Mitigating Evasion Attacks to ML Cyber Detectors. *Symmetry* 2020, *12*, 653. [CrossRef]

[10] Xiaolong, H.; Huiqi, Z.; Lunchao, Z.; Nazir, S.; Jun, D.; Shahid Khan, A. Soft Computing and Decision Support System for Software Process Improvement: A Systematic Literature Review. *Sci. Program.* 2021, *2021*, 7295627.

[11] Maikol, S.O.; Khan, A.S.; Javed, Y.; Bunsu, A.L.; Petrus, C.; George, H.; Jau, S. A novel authentication and key agreement scheme for countering MITM and impersonation attack in medical facilities. *Int. J. Integr. Eng.* 2020, *13*, 127–135.

[12] Haque, M.R.; Tan, S.C.; Yusoff, Z.; Lee, C.K.; Kaspin, R. DDoS Attack Monitoring using Smart Controller Placement in Software Defined Networking Architecture. In *Lecture Notes in Electrical Engineering*; Springer Science and Business Media LLC: Singapore, 2018; Volume 481, pp. 195–203.

[13] Nisar, K.; Jimson, E.R.; Hijazi, M.H.A.; Memon, S.K. A survey: Architecture, security threats and application of SDN. *J. Ind. Electron. Technol. Appl.* 2019, *2*, 64–69.

[14] Bovenzi, G.; Aceto, G.; Ciuonzo, D.; Persico, V.; Pescape, A. A Hierarchical Hybrid Intrusion Detection Approach in IoT Scenarios. In Proceedings of the GLOBECOM 2020—2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–7.

[15] Khan, A.S.; Javed, Y.; Abdullah, J.; Zen, K. Trust-based lightweight security protocol for device to device multihop cellular communication (TLwS). *J. Ambient. Intell. Humaniz. Comput.* 2021, *12*, 1–18. [CrossRef]

[16] Harada, S.; Yan, Z.; Park, Y.-J.; Nisar, K.; Ibrahim, A.A.A. Data aggregation in named data networking. In Proceedings of the TENCON 2017—2017 IEEE Region 10 Conference, Penang, Malaysia, 5–8 November 2017; pp. 1839–1842.

[17] Nisar, K.; Amphawan, A.; Hassan, S.; Sarkar, N.I. A comprehensive survey on scheduler for VoIP over WLAN. *J. Netw. Comput. Appl.* 2013, *36*, 933–948. [CrossRef]

[18] Chaudhary, S.; Amphawan, A.; Nisar, K. Realization of free space optics with OFDM under atmospheric turbulence. *Optik* 2014,*125*, 5196–5198. [CrossRef]

[19] Abbasi, I.A.; Khan, A.S.; Ali, S. Dynamic Multiple Junction Selection Based Routing protocol for VANETs in city environment. *Appl. Sci.* 2018, *8*, 687. [CrossRef]

[20] Li, J.; Qu, Y.; Chao, F.; Shum, H.P.H.; Ho, E.S.L.; Yang, L. Machine Learning Algorithms for Network Intrusion Detection. In

Intelligent Systems Reference Library; Springer: Berlin/Heidelberg, Germany, 2018; pp. 151–179. [CrossRef]
[21] Prasad, R.; Rohokale, V. Artificial Intelligence and Machine Learning in Cyber Security. In *Industrial Internet of Things*; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2019; pp. 231–247.
[22] Chan, K.Y.; Abdullah, J.; Khan, A.S. A framework for traceable and transparent supply chain management for agri-food sector in malaysia using blockchain technology. *Int. J. Adv. Comput. Sci. Appl.* 2019, *10*, 149–156. [CrossRef]
[23] Zhipeng Liu, Niraj Thapa, Addison Shaver, Kaushik Roy, Xiaohong Yuan & Sajad Khorsandroo (2020). Anomaly Detection on IoT Network Intrusion Using Machine Learning. [CrossRef]
[24] Zhiyuan Chen and Bing Liu. 2018. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 12,3 (2018), 1–207.
[25] Zhiyuan Chen, Nianzu Ma, and Bing Liu. 2018. Lifelong learning for sentiment classification. *arXivpreprint arXiv:1801.02808* (2018).
[26] MohammadNoor Injadat, Abdallah Moubayed & Abdallah Shami (2019). Detecting Botnet Attacks in IoT Environments: An Optimized Machine Learning Approach. [CrossRef]
[27] Satish Pokhrel, Robert Abbas, Bhulok Aryal (2021). IoT Security: Botnet detection in IoT using Machine learning. [CrossRef]
[28] Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. *Future Gener. Comput. Syst.* 2019, *100*, 779–796. [CrossRef]
[29] Abebe Diro, Naveen Chilamkurti, Van-Doan Nguyen and Will Heyne (2021). A Comprehensive Study of Anomaly Detection Schemes in IoT Networks Using Machine Learning Algorithms. [CrossRef]
[30] Khan, A.S.; Ahmad, Z.; Abdullah, J.; Ahmad, F. A Spectrogram Image-Based Network Anomaly Detection System Using Deep Convolutional Neural Network. *IEEE Access* 2021, *9*, 87079–87093. [CrossRef]
[31] Maryam Anwer, Muhammad Umer Farooq, Shariq Mahmood Khan & Waseemullah (2021). Attack Detection in IoT using Machine Learning [CrossRef]
[32] Panigrahi, R. and Borah, S. (2018). A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems. *International Journal of Engineering & Technology*, [online] 7(3.24), pp.479–482. [CrossRef]
[33] Maciá-Fernández, G., Camacho, J., Magán-Carrión, R., García-Teodoro, P. and Therón, R. (2018). UGR '16: A new dataset for the evaluation of cyclostationarity-based network IDSs. *Computers & Security*, 73, pp.411–424. [CrossRef]
[35] Ullah, I.; Mahmoud, Q.H. A Technique for Generating a Botnet Dataset for Anomalous Activity Detection in IoT Networks. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics SMC, Toronto, ON, Canada, 11–14 October 2020; pp. 134–140. [CrossRef]
[34] Saleem, M.A.; Alyas, T.; Asfandayar; Ahmad, R.; Farooq, A.; Ali, K.; Idrees, M.; Khan, A.S. Systematic literature review of identifying issues in software cost estimation techniques. *Int. J. Adv. Comput. Sci. Appl.* 2019, *10*, 341–346. [CrossRef]
[36] Kirill Eremenko (2022), Deep Learning A-Z™: Hands-On Artificial Neural Networks. [CrossRef]
[37] Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J. and Liu, H. (2018). Feature Selection. *ACM Computing Surveys*, 50(6), pp.1–45. [CrossRef]
[38] Venkatesh, B. and Anuradha, J. (2019). A Review of Feature Selection and Its Methods. *Cybernetics and Information Technologies*, [online] 19(1), pp.3–26. [CrossRef]
[39] Neda Abdelhamid, Fadi Thabtah, Hussein Abdel-jaber (2017). Phishing detection: A recent intelligent machine learning comparison based on models content and features. [CrossRef]
[40] Kurtis Pykes (2020). Oversampling and Undersampling A technique for Imbalanced Classification. [CrossRef]
[41] Lazy Programmer (2022). Tensorflow 2.0: Deep Learning and Artificial Intelligence. [CrossRef]
[42] Bisong, E. Google Colaboratory. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*; Apress: Berkeley, CA, USA, 2019; pp. 59–64.
[43] Aniruddha Bhandari. 2020. AUC-ROC Curve in Machine Learning Clearly Explained. [CrossRef]
[44] Brereton, R.G. and Lloyd, G.R. (2010). Support Vector Machines for classification and regression. *The Analyst*, 135(2), pp.230–267. [CrossRef]
[45] Vanajakshi, L. and Rilett, L.R. (2004). A comparison of the performance of artificial. neural networks and support vector machines for the prediction of traffic speed. *IEEE Intelligent Vehicles Symposium, 2004*. [CrossRef]
[46] Zhou, J., Gandomi, A.H., Chen, F. and Holzinger, A. (2021). Evaluating the Quality of Machine Learning Explanations: A Survey on Methods and Metrics. *Electronics*, [online] 10(5), p.593. [CrossRef]
[47] Jindal, M., Gupta, J. and Bhushan, B. (2019). *Machine learning methods for IoT and their Future Applications*. [online] IEEE Xplore. [CrossRef]