Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X
Volume 3, Issue 5, 2025

BOTNET DETECTION WITH ML TECHNIQUES USING THE BOT-IOT DATASET

Ishteeaq Naeem[1], Saqlain Sajjad[2], Imtiaz Hussain[3], Aqsa Zahid[4], Muhammad Sajjad[5], Mohabbat Ali[*6]

[1,2,3,4,5,*6]*Department of Computer Science, University of Management and Technology, Sialkot Campus.*

[1]ishteeaq.naeem@skt.umt.edu.pk, [2]saqlainkhan1011@gmail.com, [3]imtiaz.hussain@skt.umt.edu.pk, [4]aqsa.zahid@skt.umt.edu.pk, [5]sajjadballoch786@gmail.com, [*6]mohabbat.ali@skt.umt.edu.pk

DOI: https://doi.org/10.5281/zenodo.15494085

Keywords
Internet of Things, Botnet, BoT-IoT dataset, Machine Learning (ML).

Article History
Received on 14 April 2025
Accepted on 14 May 2025
Published on 23 May 2025

Copyright @Author
Corresponding Author: *
Mohabbat Ali

Abstract
Internet of Things (IoT) gadgets have advanced quickly in the last few years, and their use is steadily rising daily. However, cyber-attackers can target these gadgets due to their distributed nature. Additionally, many IoT devices have significant security flaws in their implementation and design, making them vulnerable to security threats. Hence, these threats can result in significant data security and privacy breaches from a singular attack on network devices or systems. Botnets are a significant security risk that can harm the IoT network; hence, sophisticated techniques are required to mitigate the risk. This research employs machine learning techniques to detect IoT devices being controlled by BotNets. The proposed technique identifies the net attack by distinguishing between legitimate and malicious traffic. This article proposes a hyper parameter tuning model to improvise the method to improve the accuracy of existing processes. The results demonstrated an improved and more accurate indication of Botnet-based cyber-attacks.

INTRODUCTION

The Internet of Things (IoT) is defined by the international standardization sector (ITUT) as a global network of linked information and communication technology (ICT) devices (Internet of Things Global Standards Initiative). IoT technology, through IoT devices, has recently enhanced the comfort, convenience, and efficiency of our lives. It is a rapidly growing technology that is transforming traditional lifestyles into modern ones [3]. For instance, many organizations use IoT technology to enhance efficiency in industries, develop smart cities, and improve healthcare services, providing significant relief and benefits to human life.

IoT devices are rapidly becoming a part of our daily lives, enabling them to sense everything in our surroundings [10]. According to a survey [9], one trillion digital devices are connected to the IoT environment via the internet. The main advantages and uses of IoT include smart homes, intelligent transportation, energy-saving, also pollution control. Overall, IoT represents revolutionary combinations of sensors, digital devices, frameworks, and intelligent systems. Despite its widespread importance in various fields, IoT introduces security gaps to limitations in soft computing and storage resources. The IoT system is integrated into numerous energy-related areas and employs various technologies for data transfer, leading to challenges and problems. Data and information security are critical aspects of IoT technology. Since the IoT environment is linked to the internet, it opens multiple vulnerabilities for attackers to exploit, compromising information and data. However, IoT security developers are committed to safeguarding data, as IoT security has become a primary economic concern. Consequently, IoT cyber-attack identification has emerged as a significant and timely topic [17]. Although IoT technology offers numerous advantages, security remains a critical real concern. To maintain security and protection of IoT devices, networks, and applications, security developers f must continuously innovate and enhance security solutions.

The number of connected devices in the IoT is rapidly increasing, and the prevalence of cyber-attacks has also increased. Among these, Botnets represent of the most significant challenges for security organizations and IoT users. A Botnet is a set of infected devices that can a botmaster can be remotely operated by a botmaster. By carefully established Command and Control (C\&C) communication channels, the bot enables the attacker to manipulate the behavior of the infected systems remotely. Botnet topologies are divided into the three main categories, C\&C communication channel: centralized, decentralized, and unstructured models. By exploiting well-known protocols such as IRC, HTTP, and P2P protocols, Botnet C\&C channels disguise their legitimate communications, making them difficult to detect. In recent years, Botnets have posed significant threats through hazardous cyber-attacks on IoT environments. A study [6] reported an average of 5200 Botnet attacks per month on IoT devices. Furthermore, attackers have increasingly escalated the frequency of cyber-attacks on IoT devices, creating an alarming situation. The reference [19] examined an auto encoder-based botnet detection method centered exclusively on anomaly detection, proposing alternative strategies for machine learning (ML) algorithms. Likewise, [27] an ML algorithm was designed exclusively for DDoS attacks, elucidating the reasons attackers often exploit DDoS techniques. The aforementioned studies highlight the limitations of traditional Botnet detection models and datasets, which often suffer from issues such as over-fitting, lack of IoT-specific traces, and imbalanced data. To address these limitations, a new approach is proposed that integrates hyper-tuning techniques with feature engineering and ML methodologies to the BoT-IoT dataset. A pre-processing strategy was employed to balance the dataset and incorporate a cross-validation approach to provide precise insights into the model's functioning. The efficiency of this model is assessed by hyper-tuned parameters and accuracy metrics. This approach effectively addresses the challenges of overfitting, class imbalance, and lack of precision, ensuring a reliable way to verify that the model meets expectations at deployment.

In this paper, three ML classification algorithms were applied: Support Vector Machine (SVM), eXGBoost, and Logistic Regression (LR). These algorithms were chosen because they are widely used and have shown strong performance in various classification tasks. SVM is a powerful algorithm that performs well in both linear and non-linear classification tasks and has high accuracy. eXGBoost is a decision-tree-based ensemble algorithm that can handle imbalanced data, and it is recognized for its high accuracy and speed. Logistic Regression is a simple yet effective algorithm that works well in binary classification problems and is particularly useful for interpretability and understanding the importance of features. These algorithms were chosen because they performed best when applied to the particular problem and data at hand. They use a dataset of patterns to create a model that can identify Botnets. While numerous datasets for Botnet detection are available online, this study used the BoT- IoT [2018] dataset. The suggested methodology for Botnet detection includes data collection, data pre-processing, and ML techniques to balance the dataset classes. Additionally, feature engineering was applied to enhance the presentation of ML algorithms used for classification.

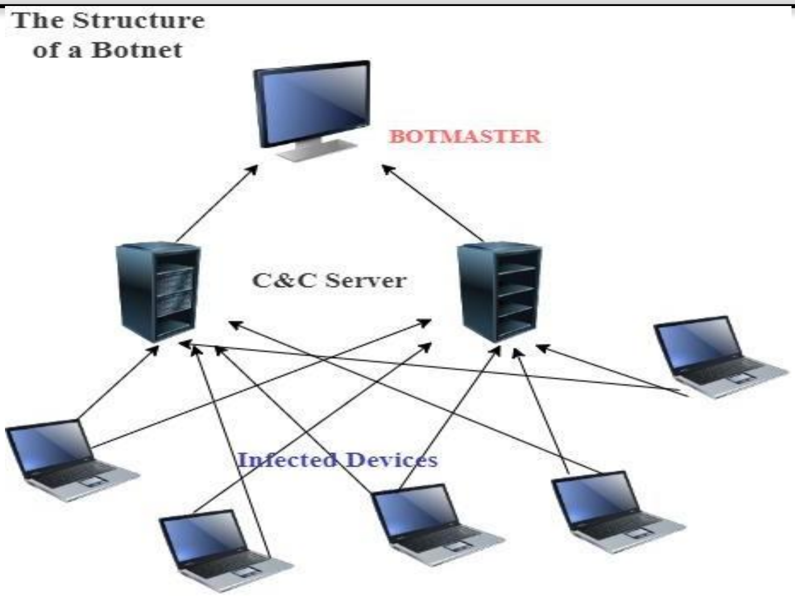


Fig. I. The Structure of a Botnet

Related work

Data imbalance significantly affects the accuracy of ML models. To address this issue, a novel method called WMI AUC was proposed by [24]. Similarly, lightweight intrusion detection models focusing on optimized feature pairs have been proposed to tackle imbalanced datasets effectively [20]. Researchers use the random forest (RFS) model for cyber- attack detection in IoT networks [18]. According to experimental findings, the model's detection accuracy is around 95%. In [13][27], the authors discuss DDoS attack detection in IoT networks using Artificial Neural Networks. The study uses the BoT-IoT dataset, which is highly imbalanced due to the disproportionate amount of regular traffic and DDoS attack traffic. To balance the data, the authors employ use the Synthetic Oversampling Technique (SMOTE), which adjusts the amount of regular attack traffic to match the DDoS attack. The Bot-IoT 2018 dataset contains 46 features; however, this study uses only 41 of them. The proposed system is designed specifically for binary classification. The data split such that 66% is used for training the proposed algorithm, while the remaining 34% is used for testing the network.

Experimental results indicate that the use of SMOTE enabled a DDoS attack detection rate of 100% In [16], the author discusses the use the new BoT-IoT dataset and presents a test environment for intrusion detection. They employed Long Short-term memory (LSTM), Recurrent Neural Network (RNN), and SVM Classifier ML methods. All classifier ML algorithms have been trained and evaluated. The results show that the training time for the SVM was high, its detection ratio outperformed the other algorithms when applied to all 46 features. However, the accuracy of the other algorithms improved significantly when only using the top 10 features. Despite this, the overall detection ratio of SVM remained greater than that of the LSTM and RNN models. Several Botnet attacks have targeted IoT networks.

In this research [25], new standards for signature-based malware detection in IoT are proposed. The author utilizes the BoT-IoT dataset for model training and testing, using 659,015 records of data with 43 input features. The J48 model is employed to generate rules for attack detection. This model develops 24 rules: one rule for detecting 'DDoS' attacks, one for 'Theft' attacks, and all other rules for 'Probing' attacks. Compared to other public signature-based IDS tools such as Snort and Suricata, the proposed model performs better in Botnet detection and rule extension. However, the study did not include accuracy metrics or evaluation results. In the paper[14], the authors proposed a hybrid intrusion detection system (HIDS) to enhance the accuracy of IoT attack detection. The anomaly-based IDS detects zero-day attacks, while the signature-based IDS identifies well-known intrusions. One-class SVMs are employed for the anomaly-based component, and C5 Decision Trees are utilized for the signature-based component. The outputs of these classifiers are integrated using a boosting method. The HIDS model uses only 13 features from the original 46 features in the BoT-IoT dataset. When used separately, the anomaly-based and signature-based models achieve accuracy levels of approximately 92% to 93%. However, combining the two models with the proposed method increases the attack detection accuracy to 99.97%. The researchers also compare their model to the alternative algorithms, including SVM, KNN, Random Forest, and Naive Bayes. Among these, the proposed model achieved the highest attack detection accuracy. Nevertheless, the experimental findings indicate that the model's training time is considerably high. In[4], the researchers tested the detection mechanism using SVM Classifier, Naive Bayes (NB), DT, and K-Nearest Neighbors (KNN).

These models were applied to 82,000 records from the USNW-NB15 2018 dataset for Botnet identification. To address the challenge of dimensionality in large datasets, the chi-squared method was used for feature selection, reducing the number of features analyzed. Among the models, the DT model demonstrated the best performance, achieving a 99.89% success rate for Botnet identification using 20 selected features. Additionally, the DT model scored a perfect 100% in F1 score, precision, and recall, highlighting its exceptional effectiveness in identifying Botnets in IoT systems accuracy was good, according to the results, but on the other hand, it received a, score. The goal of this study [23] is that the author emphasizes the selection of ML algorithms from many algorithms. They proposed a new and hybrid framework ML model for Intrusion detection. BoT- IoT 2018 dataset has many features and 73 **million records.** Due to many features, they select only forty-four features from the BoT-IoT dataset. Recent similarly focuses on enhancing Botnet detection in IoT using optimize ML models for more identification [21]. Their selection method for ML algorithms is a Objective Soft set. Before applying this method, they applied only five ML algorithms for intrusion detection to evaluate the best ML algorithm using the Objective Soft method. Their result shows that applying the method gets better accuracy results for invasion detection in IoT devices. In [12], the researchers identified a 215.7% increase in cyber- attacks on IoT devices from 2017 to 2018, highlighting a critical and harmful trend for IoT systems. To address this issue, they proposed a novel Bayesian Optimization Gaussian Process technique. The DT classification method was used to perform malware recognition in IoT environments to enhance detection effectiveness. Performance evaluation using the BoT-IoT 2018 dataset demonstrated promising outcomes, with the approach yielding high accuracy, F1 score, precision, and recall score for Botnet detection. The author of[22] focused on a feature selection method to increase the attack detection ratio within the IoT ecosystem. They utilized a featured selection approach named CAUC (Correlation Area under the Curve) and developed a new CAUC model based on this approach. This model relies upon the wrapper technique to identify the optimal features for attack detection in the proposed ML algorithm. Furthermore, they validated their approach using a soft objective set based on Shannon entropy, integrated with the TOPSIS method.

The researcher employed four ML algorithms and achieved an overall average accuracy exceeding 96%. The author of [2][2] provides a novel approach for distinguishing Botnet traffic from ordinary traffic and Categorizing the types of attacks carried out by cyber criminals on IoT devices. For this purpose, the BoT- IoT dataset, which includes various attack and sub-attack categories, was utilized. To preprocess the data, the researcher applied the Synthetic over Sampling Technique (SMOTE). The study compared the performance of three ML algorithms: J48, Random Forest, and Multilayer Perception. According to the author, the binary classifier achieved an accuracy of 0.99%, while accurateness for the main outbreak and subcategory classifications was 0.96% and 0.93%, respectively. In addition, the author analyzed the False Negative (FN) rates of J48 and Random Forest classifiers. Different ML methods can be used to identify Denial of Service (DoS) threats in IoT networks.

In [5], the author focuses on detecting DoS attacks in wireless sensor networks (WSNs) within IoT systems and proposes two new frameworks, termed voting and multi scheme. These frameworks integrate the average one-dependence estimator (A1DE) and a two-dependence estimator (A2DE). The BoT- IoT 2018 dataset was utilized for both testing and training phases of this model. In the multi-scheme classifiers, the entire dataset was used during the training phase, either by applying the A1DE or the A2DE classifier. On the other hand, the Voting Scheme combines the outputs of the selected classifiers, namely A1DE and A2DE. During the testing phase, if the results from both classifiers match, the outcome is accepted. However, if there is a discrepancy between the results, one of the outcomes is chosen randomly. From the BoT-IoT dataset, the author utilized 477 ordinary traffic samples and 3,668,045 legitimate traffic samples. The results indicate that when using only five features from the BoT-IoT dataset, the detection accuracy of the Multi-Scheme Mechanism classifier and A2DE was identical. However, the training time for the Multi-Scheme classifier was longer than for A2DE. Although the detection accuracy of the Voting Scheme was higher than that of the Multi-Scheme classifier, it also required significantly more training time. Furthermore, the author compared A1DE and A2DE with RNN.

The findings revealed that A2DE outperformed RNN. However, this research exclusively focuses on DoS attacks and does not address other types of attacks. The goal of the study [6] is to establish an effective hybrid predictive model, XGB-RF, for identifying Botnets in IoT devices. The proposed scheme combines eXtreme Gradient Boosting (XGB) with the Random Forest (RF) classifier. According to the author, Random Forest was employed for feature selection, while XGB was used for Botnet detection in IoT networks. The training and testing phases utilized the N-Bayes IoT dataset, comprising 115 features and 229,829 total records. However, the study focused on 13,113 regular instances and 216,716 attack traffic instances. The authors reported that the detection accuracy results of their system for intrusion detection outperformed other state-of-the- art ML techniques. Specifically, their methodology achieved a detection accuracy of 99.97% using only 40 features from the dataset. A notable drawback of this scheme, was its substantial time requirement for intrusion detection.

According to this study [11], the authors present a deep learning technique for intrusion detection in IoT networks, emphasizing the effects of adversarial robustness. They use Adversarial Robustness Toolbox (ART) to generate a malicious sample. The study employs two DL classifiers (FNN) and (SNN). Using the top- 10 features of the 3.6 million record BoT-IoT 2018 dataset, the models were evaluated for accuracy, precision, and recall. Results showed that FNN achieved an accuracy of 94%, out-performing SNN under normal conditions. When adversarial samples were included in the dataset, SNN demonstrated better accuracy for intrusion detection compared to FNN. Addition- ally, the authors utilized feature normalization to enhance the models' accuracy, achieving improved results even without adversarial samples.

This finding underscores the importance of feature normalization in increasing the accuracy of intrusion detection systems. The authors in [8] introduce a novel paradigm for multiple and bipolar classifications using feed forward neural networks. The study achieved a 99% intrusion detection accuracy rate for reconnaissance, DoS, and DDoS in multi-class classification, the detection accuracy for DoS and DDoS attacks was 0.99, and for reconnaissance attacks, it was 0.98. However, the detection rate for information theft attacks dropped to 88%. This discrepancy highlights the imbalance in the BoT-IoT dataset, as the detection ratio for information theft attacks was notably lower compared to other attack types. This study [7] explores the detection of attacks in IoT net- works using seven deep learning models, categorized into two groups. This first group includes conditional models such as (RNN), (DPP), and CNN. The second group comprises Deep Belief Networks (DBN), Deep Auto-encoders (DA), Restricted Boltzmann Machines (RBM), and a Deep Boltzmann Machine (DBM).

Performance was assessed using the datasets BoT-IoT 2018 and CSECICIDS2018, Results of experiments show that the CNN approach got the greatest binary classification accuracy on both data sets. On the CSECICIDS2018 data set, the CNN model reached 97. 37% accuracy and needed 330 seconds for training. The BoT-IoT dataset had a botnet detection accuracy of 98.37 percent and a training time of 1367 seconds. The Particle Deep Framework is an IoT intrusion detection technique introduced in this research. The source of cyber-attacks in IoT networks is meant to be tracked by this forensic network system along with Botnet assaults detection. Al-Kasssbeh et al. [1] used Fuzzy Rule Interpolation (FRI) for botnet detection in IoT networks. The usage of a fuzzy system tries to reduce the complexity of intrusion detection systems and make them easier to implement. The authors trained the proposed technique using the BoT-IoT 2018 dataset, focusing on the top five attributes and 50,000 records. Our method attained a 95% detection accuracy rate, indicating its potential for intrusion detection in IoT systems.

Problem Statement/Definition

Numerous studies have explored Botnet detection models, but few focus on utilizing feature extraction to conduct a comprehensive evaluation and avoid challenges associated with large datasets, such redundancy and multi-collinearity. While using conventional datasets without feature engineering may address some issues, it of results in over-fitting, which compromises the model's generalizability. Many studies rely on traditional datasets that lack IoT traces, making them ineffective for detecting modern Botnet problems within the IoT environment. Moreover, most studies utilize real-time datasets, which are often highly imbalanced. While they aim to achieve high correctness by various MLAs on these datasets, they often overlook the significant impact of dataset imbalance on model performance.

The precision derived from such imbalanced datasets can be misleading, as the model may perform well on the majority class while failing to identify minority class instances effectively. Furthermore, many studies evaluate the performance of MLAs based solely on accuracy after training the model. However, this approach only measures the model's ability to handle the training data, failing to assess its generalization ability and performance on unseen data.

The need for more robust and accurate detection models, capable of handling imbalanced datasets and generalizing to unseen data, remains a critical challenge in the field. Many Botnet detection algorithms suffer difficulties due to class imbalance, in which the majority class dominates the training process, resulting in biased results. As a result, accuracy becomes a less useful criterion for evaluating model performance because it may neglect the model's capacity to accurately detect the minority class. To overcome this issue, we propose combining hyper parameter tweaking, feature engineering, and machine learning approaches. For this investigation, we chose the BoT-IoT dataset, a recent and relevant dataset created in an IoT environment that includes both regular and attack traffic records. This dataset is extremely unbalanced, with attack traffic occurring substantially less frequently than routine traffic. To mitigate the effects of class imbalance, a preprocessing strategy was employed to balance the dataset before training. We utilized the cross-validation approach to access our model's performance precise accurately.

MATERIAL AND METHOD

The study aims to assess ML models using the BoT-IoT dataset. The BoT-IoT dataset was created by the Cyber Range Lab at the University of North South Wales(UNSW) in 2018 and was subsequently published in 2019. The data for BoT-IoT was assembled within a simulated environment featuring with multiple virtual machines operating on various operating systems. Tools such as the Argus network security tool, Node-RED, network taps, and network firewalls [21]. The Argus security tool generated 73 million data records in the Comma Separated Values (CSV) file format, representing the original version of the dataset.

Training and validating a model using a vast amount of data requires significant time and computational resources. To overcome this issue, a 5% subset from the original dataset was created. This dataset has 46 features, with three dependent features and 43 independent features. In this study, the 10 best features were selected from the 5% subset of the dataset using a combination of the mapping correlation coefficient and entropy method. These 10 features were selected based on their rankings Table I shows the traffic characteristics of the 5% subset for the 10 features.

TABLE I. SUBSET RECORDS

Category	Sub Category	Records
DoS	HTTP TCP UDP	1184 826349 492615
DDoS	TCP HTTP UDP	7,82,228 786 758201
Reconnaissance	OS_Fingerprinting Service Scanning	14293 58626
Normal	Normal	370
Information Theft	Keylogging Data Exfiltration	59 6

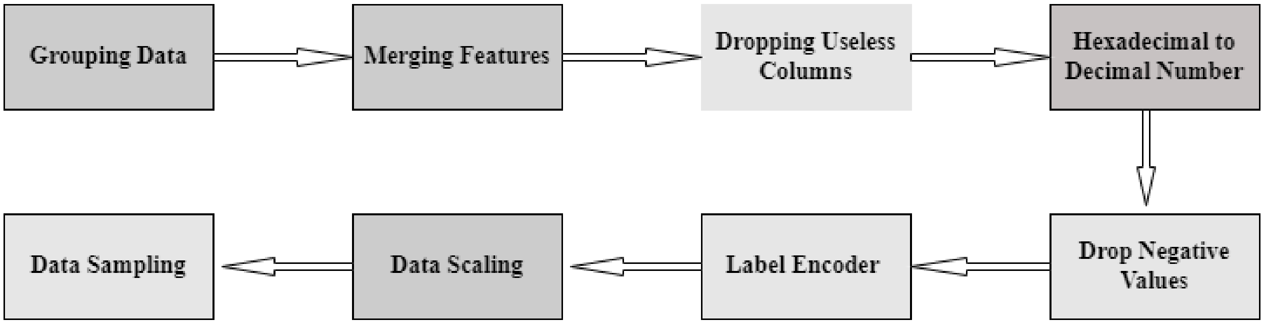


Fig. 2. Steps of preprocessing

When the data are uploaded into the Jupyter Notebook for preprocessing, it becomes ready for training and testing to measure the model's performance. Real-world data often contains null values, duplicate and missing values, or data in a format unsuitable for the ML model for performance evaluation. The ML model's ability to learn directly depends on the quality of the input data, making data preprocessing an essential step. The primary objective of data preprocessing is to convert raw data into a machine-readable format. Datasets are composed of data objects, referred to as vectors, samples, or records, and are described by features. Features provide detailed information about objects and are also known as attributes or variables. Features are classified into two main types:

Grouping Data: During the preprocessing step, the dataset's category and subcategory features are grouped together. After grouping, the occurrences of each category are counted along with their corresponding subcategories. Fig. 3 illustrates the number of attacks, categorized by their respective subcategories, along with their counts. Merging Data: In the next step, the Category column and 'Subcategory' column are combined into a single column named 'target'. This new name is 'target' col- umn serves as the primary feature for identify and classifying the data.

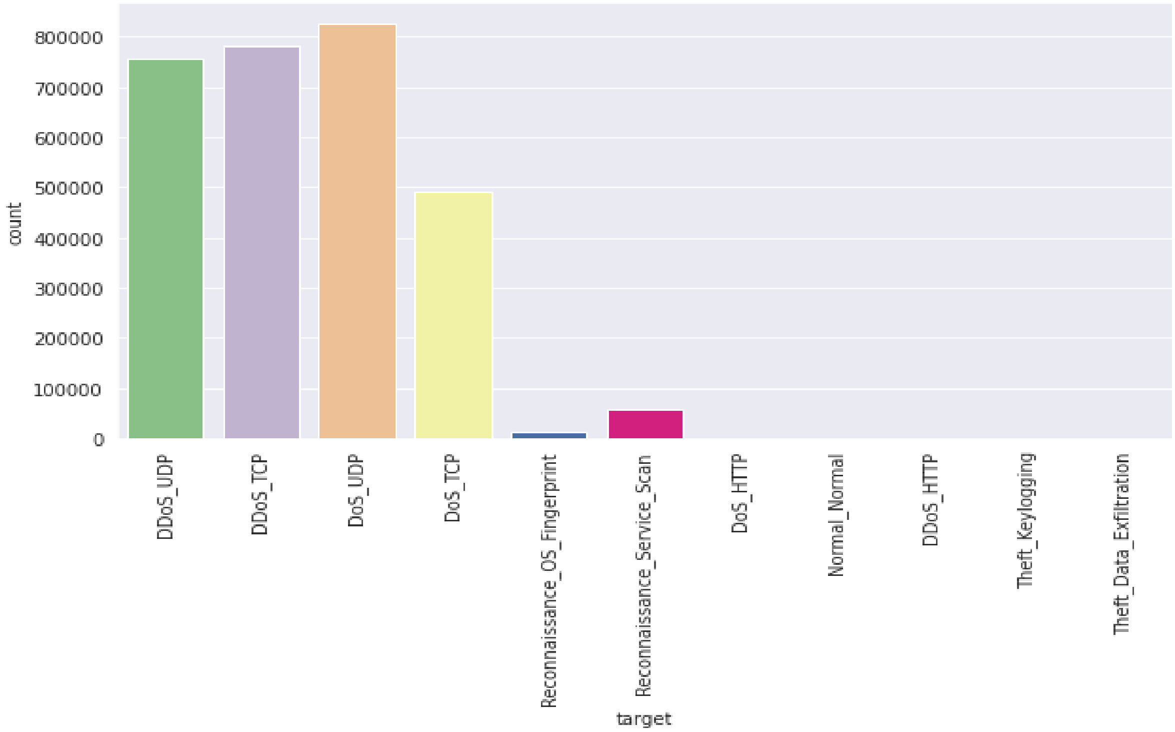


Fig. 3. Group by data in graphic form

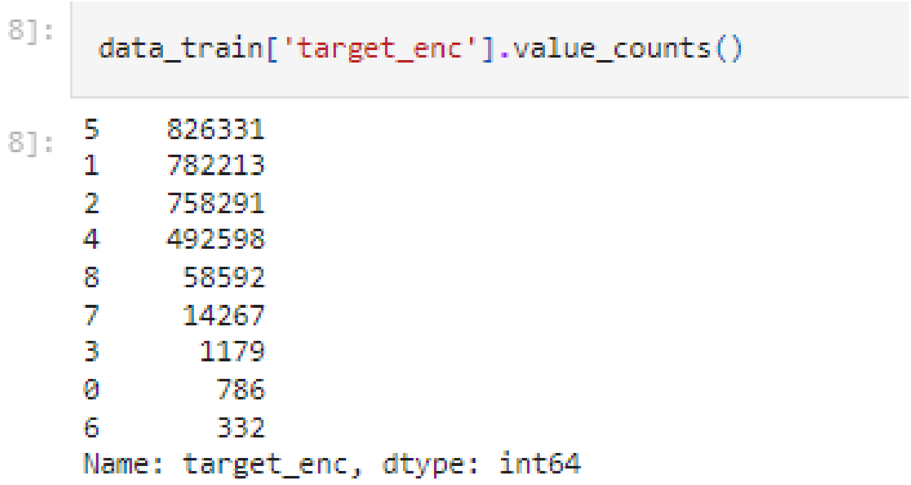
Merging Data: In the next step, the 'Category' column and 'Subcategory' column are combined into a single column named 'target'. This new name is 'target' column serves as the primary feature for identify and classifying the data.

Dropping Column: During the analysis of attack counts, it was observed that the 'theft' attack category had very low values, which could negatively affect the ML models' accuracy. To address this issue, the 'theft' column was removed from the dataset, ensuring it does not skew the results. Additionally, some features were deemed unnecessary for the ML model's performance during detection. To optimize time complexity and conserve storage resources, the 'pkSeqID' and 'seq' features were also removed from the data frame. This helps streamline the dataset by retaining only the relevant features, thereby improving the efficiency and accuracy of the model.

Hexadecimal to Decimal: During the preprocessing step, it was observed that the 'sport' feature contained hexadecimal values. A condition was applied to identify values starting with '0x', indicating hexadecimal numbers. These values were evaluated and converted to their corresponding decimal equivalents, replacing the original hexademical numbers in the data fram. The same procedure was applied to the 'dport' column, which also contained some hexadecimal numbers. These values were similarly replaced with their corresponding decimal equivalents. Drop Negative Values: Next, the 'dport' column was examined to identify values less than one. Since the 'dport' feature should not contain negative or zero values, any rows with such values were removed from the dataset. After applying these preprocessing steps, the values in the 'target' feature were counted to verify the dataset's integrity and ensure the distribution of data was appropriate for ML model training and evaluation.

Drop Negative Values: Next, the 'dport' column was examined to identify values less than one. Since the 'dport' feature should not contain negative or zero values, any rows with such values were removed from the dataset. After applying these preprocessing steps, the values in the 'target' feature were counted to verify the dataset's integrity and ensure the distribution of data was appropriate for ML model training and evaluation.

Label Encoder: It can improve model prediction by converting categorical data into integer form, making it machine-readable. The label encoder is a well-known method for converting labels to numeric form, which facilitates machine-reading. A label encoder is used to normalize the labels. The sklearn library provides an efficient method for labeling data through the Label Encoder function. Four features, 'target,' 'proto,' 'saddr', and 'daddr,' were converted into numeric form by applying the label encoder process. It was then observed that the 'target', 'proto', 'saddr', and 'daddr' features were unnecessary, so they were dropped them from the training and testing data. The 'target-enc' feature now has the following numeric values: 5 for DoS UDP, 1 for DDoS TCP, 2 for DDoS UDP, 4 for DoS TCP, 8 for Reconnaissance Service Scan, 7 for Reconnaissance OS Fingerprint, 3 for DoS HTTP, and 6 for Normal. Fig. 4 shows the values of all attacks in numeric format after applying the label encoder, which converts the categorical labels to a numeric format.



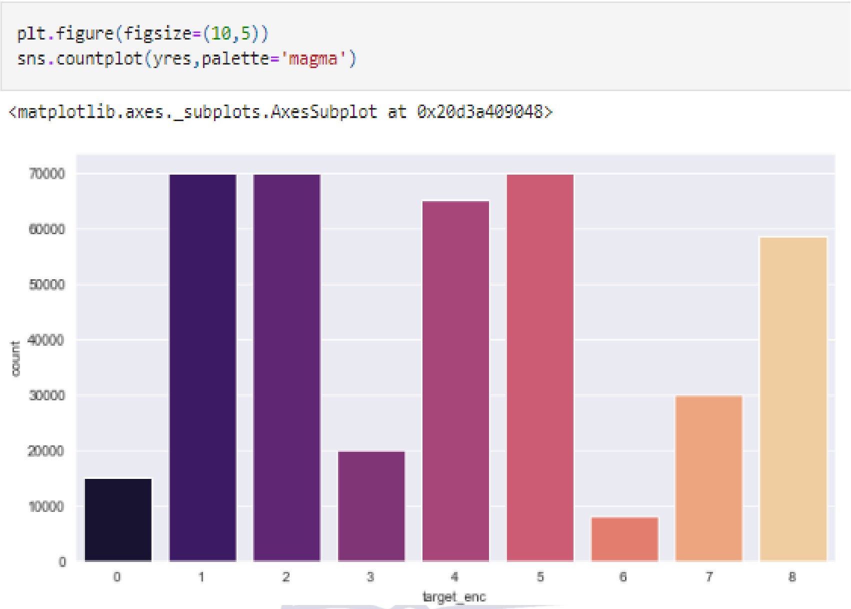
```
8]:  data_train['target_enc'].value_counts()

8]:  5    826331
     1    782213
     2    758291
     4    492598
     8     58592
     7     14267
     3      1179
     0       786
     6       332
     Name: target_enc, dtype: int64
```

Fig. 4. Categorical to Numeric Form

Data Scaling: Scaling is used to adjust the input data range so that the values become closer to each other. In this case, we used the Standard Scaler to standardize the data by subtracting the mean and dividing it by its standard deviation. Unit variance refers to the distance of each value from the mean. A key function of the Standard Scaler is to remove the mean and data, adjusting the distance of each value from the mean

Data Sampling: Sampling in preprocessing is a useful step when the data is too large and difficult to analyze. There are several methods for selecting sample data. In this research, the data Random Over-Sampler method has been used for sampling. Fig. 5 illustrates the method for selecting the random values of attacks for evaluation metric measurements.

Graph Representation: Graphs can present data in an easily understandable form, making it easier to interpret individual behaviors. Sample data in graphic form clearly displays the values of all the attacks. Fig. 5 Graphs are used to illustrate the data of the sample data obtained using the Random Over-Sampler method.



```
plt.figure(figsize=(10,5))
sns.countplot(yres,palette='magma')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x20d3a409048>
```

Fig. 5. Traffic in Graphic Format

In this graph, 0 represents DDoS HTTP, 1 represents DDoS TCP, 2 represents DDoS UDP, 3 represents DoS HTTP, 4 represents DoS UDP, 5 represents DoS TCP, 6 represents Normal, 7 represents Reconnaissance_OS_Fingerprint, and 8 represents Reconnaissance Service Scan. This thesis focuses on supervised ML techniques for Botnet detection using the BoT-IoT dataset. There are numerous supervised algorithms available for Botnet detection. The next question is which ML algorithm is best suited to improve our accuracy prediction results. In this paper, three types of ML algorithms were selected based on classification:

1) Logistic Regression (LR)
2) eXtreme Gradient Boost (eXGBoost) model
3) Support Vector Machine (SVM)

Logistic Regression (LR)

The LR approach is employed when the independent variable is quantitative or continuous, but the dependent variable is discrete. This model is very significant relative to other ML algorithms because of its capacity to identify the most pertinent features and classify various data types.

Support Vector Machine (SVM)

SVM is used to locate a partition called a hyper plane that separates the distinct classes. The margin is the gap between the hyper plane and the support vectors that SVM produces for separating the data in the two classes. SVM attempts to find the hyper plane that provides the largest separation between the two classes. To select the optimal hyper plane within the data, the SVM kernel transforms the input data in practice. SVM kernel maps data points from a smaller to a larger number of dimensions. This method effectively converts non-separable issues into separable ones by adding dimensions to the feature space.

Exgboost Model

Extreme Gradient Boosting, or eXGBoost, is a scalable and effective application of the Gradient Boosting Decision Tree (GBDT) machine learning toolset. The decision tree method, a highly successful machine learning technique for tree boosting, is the foundation of this strategy. Gradient boosting, decision trees, ensemble learning and supervised machine learning are all leveraged by eXGBoost. Because of its ability to handle large datasets and generate high-performance models, it is frequently employed for classification and regression issues.

Confusion Matrix

The confusion matrix is a useful tool for calculating the accuracy of ML models. It is a performance evaluation method is for classification algorithms. The confusion matrix assesses the performance of a classification algorithm by identifying mistakes, such as false positives and false negatives.

TABLE II. CONFUSION MATRIX

Confusion Matrix		Actual Values	
		Positive	Negative
Predicted Values	Positive	TP	FP
	Negative	FN	TN

In the field of cyber security, extensive research and understanding are required to define cyber-attacks as positive. events. The identification of these events, based on outcomes, shows whether cyber-attacks are present. Adverse event present regular traffic in the cyber security, and the true negatives correspond to normal traffic. If misclassification occurs, where regular traffic is classified as attack traffic, this classification is called False Positive. Similarly, when attack traffic is misclassified as normal traffic, it is to as a False Negative. Building an ML model and evaluating its effectiveness is a significant undertaking. Questions arise about how the effectiveness of the model can be measured by using a variety of performance metrics, and when to stop training and evaluation to achieve the best results. Evaluation metrics are essential for judging the performance of an ML model. Model performance can be measured usage of numerous evaluation metrics, such as precision and recall. Relying solely on accuracy for performance evaluation can be problematic, and it is not a comprehensive assessment. Performance metrics depend on the four values (TP, TN, FP, FN) derived from the confusion metrics. While there are numerous methods to measure an ML model's effectiveness, this focuses on four primary performance metrics.

Accuracy

These metrics help determine the error rate of the classifier model then, count the number of correct predictions the model has made. Accuracy is determined using the formula below.

$$\frac{accuracy = TP + TN\ (Positive\ Prediction)}{TP + TN + FP + FN\ (Total\ Prediction}$$

Precision

Precision is the percentage of accurate predictions among all those that the model correctly identifies. It plays a vital role when false positive errors are expensive, allowing us to quantify the consequences. Precision is calculated using the following equation.

Recall

$$\text{precision} = \frac{TP(\text{Positive Prediction})}{TP + TN\ (\text{Total Positive Prediction})}$$

The number of genuine positive predictions divided by the total number of real samples in the dataset is the recall.

F1 Score

$$\text{recall} = \frac{TP(\text{Positive Prediction})}{TP + FN\ (\text{sum of sample belongs to Positive Prediction})}$$

The F1 score is obtained by applying a harmonic average to recall and precision. It gives an objective measure of how well a model predicts by considering both the recall and precision metrics. A high F1 score means that the both precision and recall are good. The F1 score ranges from 0 to 1; Models usually struggle when their F1 scores are low and high F1 scores normally signify good performance.

$$\text{F1 Score} = \frac{2(\text{precision x recall})}{\text{precsion} + \text{recal}}$$

RESULTS

The baseline accuracy serves as a benchmark for evaluating the performance of each model. The best models are trained on the entire training set before being tested on a new set of data. Most current research uses datasets that contain only a small number of BotNet traces. Although these methods often report good detection rates, it is unclear how they would perform on larger datasets. To address this, a larger botnet dataset produced by the University of North South Wales was used. This block diagram below shows the results of the baseline model accuracy on both training and testing data.

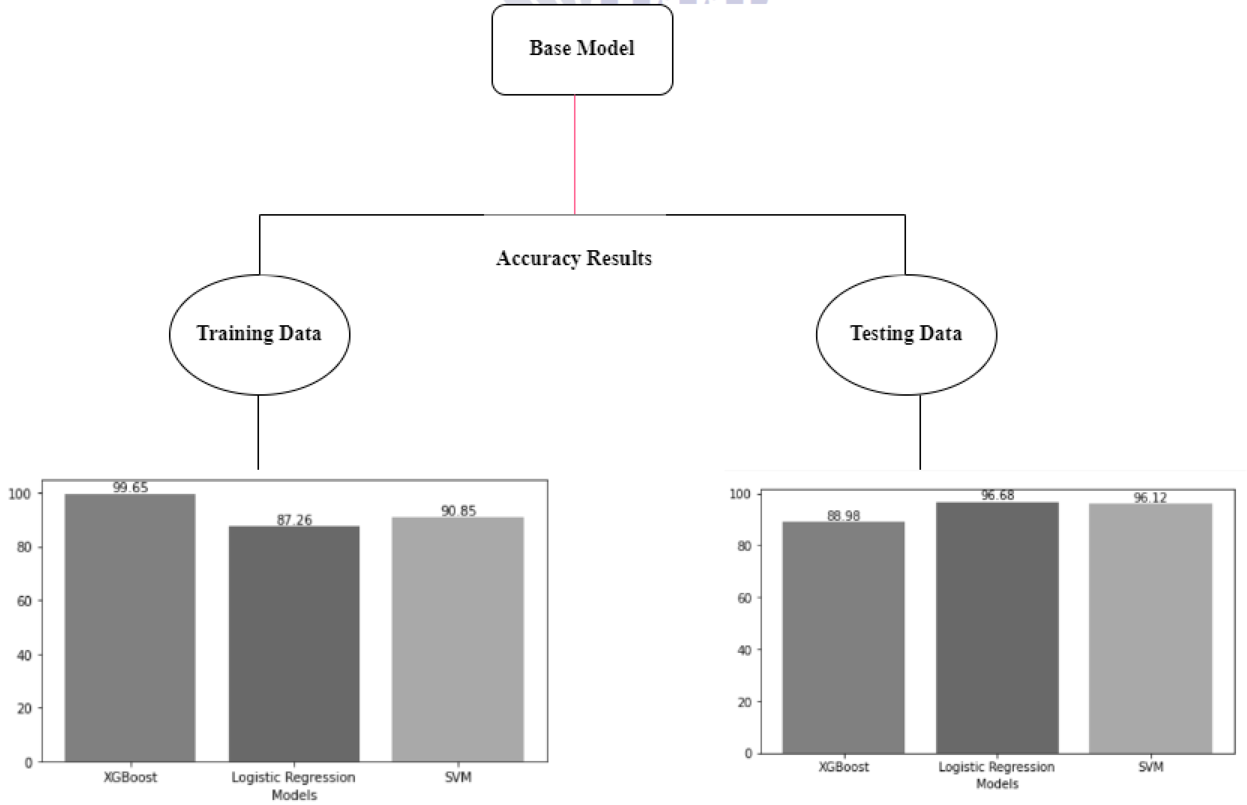
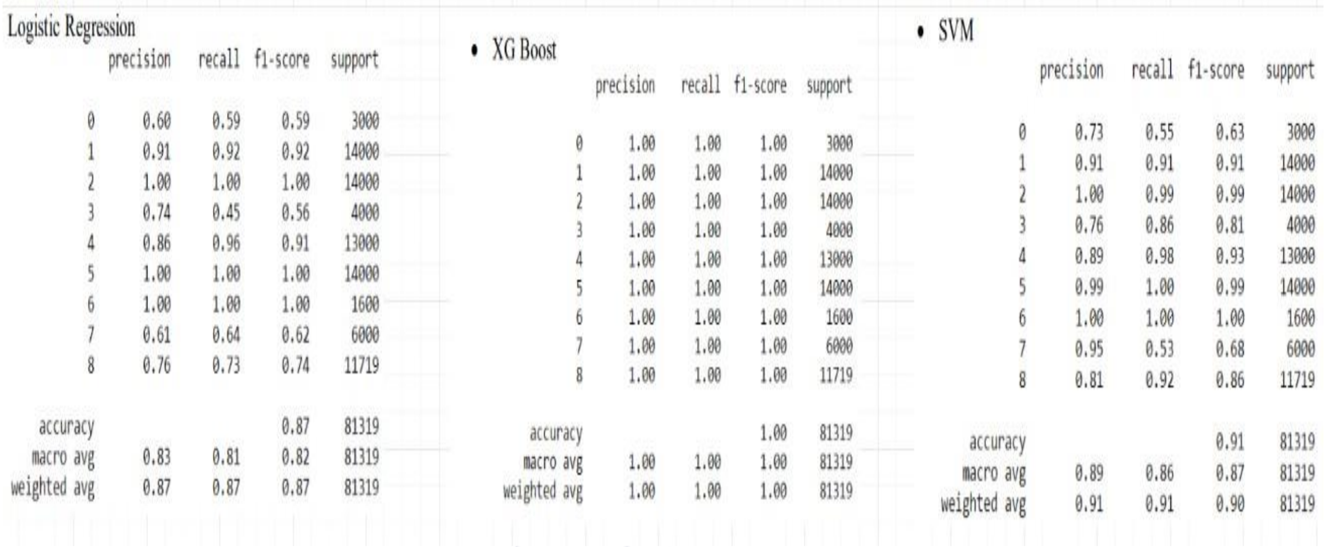


Fig. 6. Base Model Results

Three ML models were trained on the preprocessed training and testing data using a technique known as train-test split. Data for training and testing was divided using a 80/20 break down. It is revealed that the eXGBoost achieved the highest accuracy rate when applied to the training data, with an accuracy of 99.65%. When tested on unseen data, the SVM model performed better than the other models with an accuracy of 96.12%. Classification reports allow evaluation of ML model accuracy and various other metrics such as precision, recall, F1-score and support vector. A classification report shows you how well your trained model performs on a classification task by indicating its recall, F1-score, accuracy and support score. It shows the number of correct predictions as well as the number of incorrect ones made by the model. A breakdown of classification statistics for the eXGBoost model is presented in Fig. 7.



Logistic Regression

```
              precision    recall  f1-score   support

           0       0.60      0.59      0.59      3000
           1       0.91      0.92      0.92     14000
           2       1.00      1.00      1.00     14000
           3       0.74      0.45      0.56      4000
           4       0.86      0.96      0.91     13000
           5       1.00      1.00      1.00     14000
           6       1.00      1.00      1.00      1600
           7       0.61      0.64      0.62      6000
           8       0.76      0.73      0.74     11719

    accuracy                           0.87     81319
   macro avg       0.83      0.81      0.82     81319
weighted avg       0.87      0.87      0.87     81319
```

- XG Boost

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      3000
           1       1.00      1.00      1.00     14000
           2       1.00      1.00      1.00     14000
           3       1.00      1.00      1.00      4000
           4       1.00      1.00      1.00     13000
           5       1.00      1.00      1.00     14000
           6       1.00      1.00      1.00      1600
           7       1.00      1.00      1.00      6000
           8       1.00      1.00      1.00     11719

    accuracy                           1.00     81319
   macro avg       1.00      1.00      1.00     81319
weighted avg       1.00      1.00      1.00     81319
```

- SVM

```
              precision    recall  f1-score   support

           0       0.73      0.55      0.63      3000
           1       0.91      0.91      0.91     14000
           2       1.00      0.99      0.99     14000
           3       0.76      0.86      0.81      4000
           4       0.89      0.98      0.93     13000
           5       0.99      1.00      0.99     14000
           6       1.00      1.00      1.00      1600
           7       0.95      0.53      0.68      6000
           8       0.81      0.92      0.86     11719

    accuracy                           0.91     81319
   macro avg       0.89      0.86      0.87     81319
weighted avg       0.91      0.91      0.90     81319
```

Fig. 7. Classification Reports

Next, we perform 12-fold cross validation. The model's generalization is validated by the validation accuracy, which helps to determine which model would perform well on hypothetical test data. This process indicates that 50 performance scores are collected for each metric by each classifier. In k-fold cross-validation, each instance is used both for training and validation a total of k+1 times. In five-fold cross-validation, every instance is assigned to the validation set exactly once and to the training set four times. Cross-stratified validation ensures that the different classes are equally distributed across the folds. Fig. 8 shows the model accuracy percentage using the cross-validation techniques.

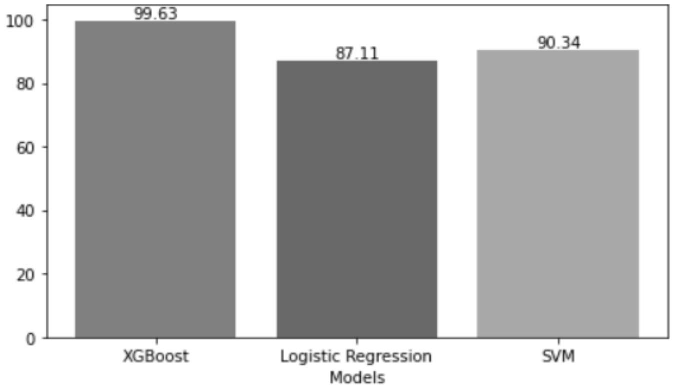


Fig. 8. Results of cross folder validation techniques

Fig. 8 shows that the eXGBoost model achieved the highest accuracy percentage, with a value of 99.63%. The effectiveness of ML algorithms depends on various factors, including dataset quality, relevant features, and properly tuned hyperparameters. However, many previous studies exhibit feature selection bias. They often rely on datasets with limited botnet diversity, undervaluing data quality. Additionally, these studies frequently neglect hyperparameter tuning, which is essential for enhancing the performance of ML models.

In this study, we utilize a recent dataset with a wide range of botnet traces and flow features. Several preprocessing methods are employed to select the most relevant features. Moreover, the hyperparameters of ML algorithms are tuned to identify the best-performing models. Machine learning involves two types of parameters: those determined by the training data, such as weights in LR, and those that specific to the learning algorithm itself, referred to as hyperparameters. Hyperparameters influence other model parameters and play a crucial role in model selection. Selecting the optimal hyperparameters involves adjusting and comparing various parameter settings to improve performance on unseen data.

Cross-validation was employed to the training data to evaluate generalization performance, as using test datasets for model selection is no considered a robust ML practice. The detection accuracy percentage of the model for the training and test datasets are presented in the Fig. 9.

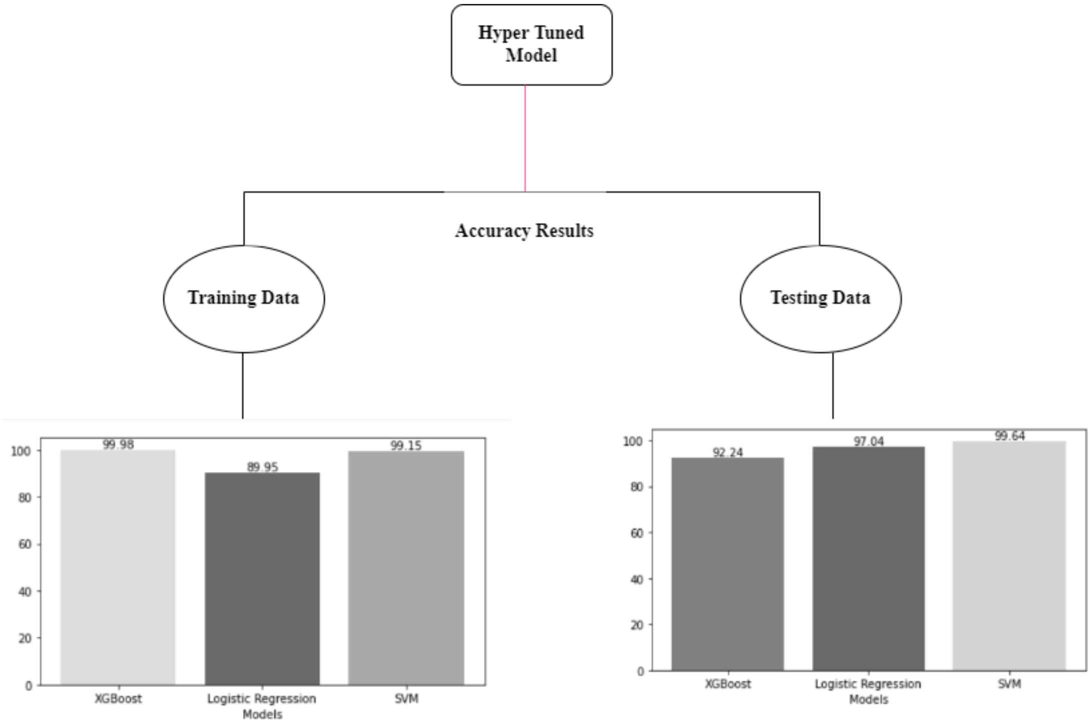
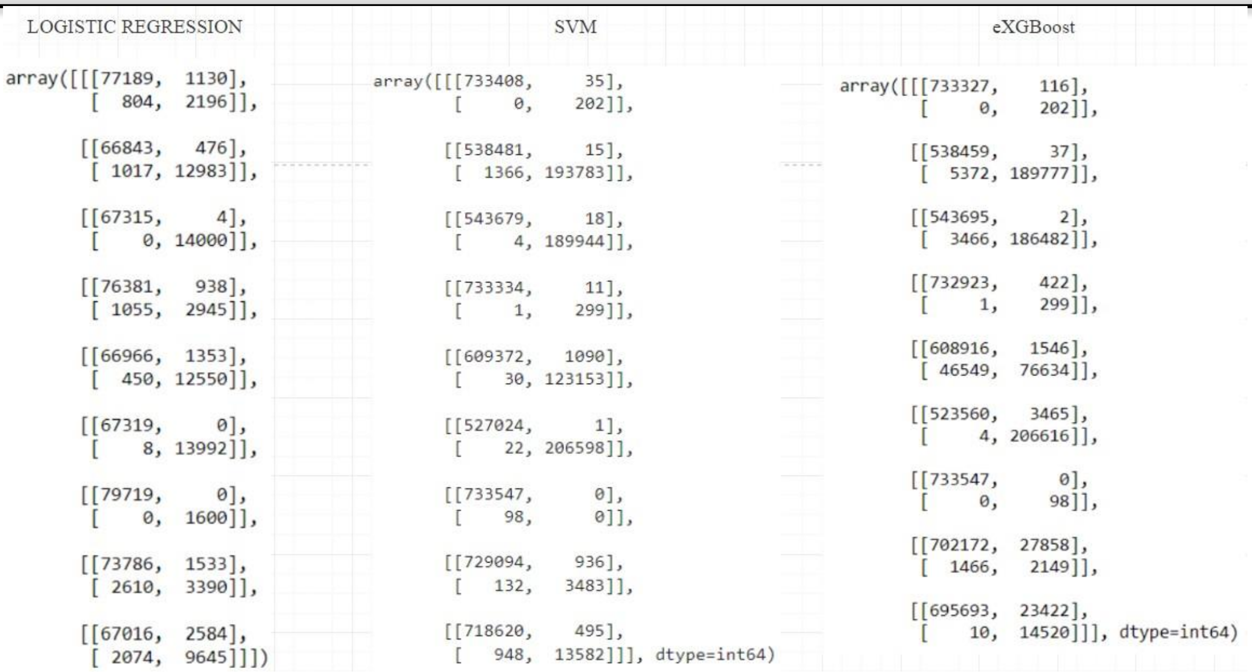


Fig. 9. Hyper Tuned Model Results

The result shows that on the training dataset, the eXGBoost model achieved 99.98% accuracy. However, on the testing dataset, the SVM model demonstrated the highest detection accuracy among all models, making it the best performing model in this study. In our analysis, the SVM model achieved the best detection accuracy percentage. Fig. 10 shows the confusion matrix for the evaluated models.



LOGISTIC REGRESSION

```
array([[[77189,  1130],
        [  804,  2196]],

       [[66843,   476],
        [ 1017, 12983]],

       [[67315,     4],
        [    0, 14000]],

       [[76381,   938],
        [ 1055,  2945]],

       [[66966,  1353],
        [  450, 12550]],

       [[67319,     0],
        [    8, 13992]],

       [[79719,     0],
        [    0,  1600]],

       [[73786,  1533],
        [ 2610,  3390]],

       [[67016,  2584],
        [ 2074,  9645]]])
```

SVM

```
array([[[733408,     35],
        [     0,    202]],

       [[538481,     15],
        [  1366, 193783]],

       [[543679,     18],
        [     4, 189944]],

       [[733334,     11],
        [     1,    299]],

       [[609372,   1090],
        [    30, 123153]],

       [[527024,      1],
        [    22, 206598]],

       [[733547,      0],
        [    98,      0]],

       [[729094,    936],
        [   132,   3483]],

       [[718620,    495],
        [   948,  13582]]], dtype=int64)
```

eXGBoost

```
array([[[733327,    116],
        [     0,    202]],

       [[538459,     37],
        [  5372, 189777]],

       [[543695,      2],
        [  3466, 186482]],

       [[732923,    422],
        [     1,    299]],

       [[608916,   1546],
        [ 46549,  76634]],

       [[523560,   3465],
        [     4, 206616]],

       [[733547,      0],
        [     0,     98]],

       [[702172,  27858],
        [  1466,   2149]],

       [[695693,  23422],
        [    10,  14520]]], dtype=int64)
```

Fig. 10. Confusion Matrix for Evaluated Models

CONCLUSION

Researchers and developers worldwide are increasingly interested in recent developments in IoT. Scholars and IoT programmers are collaborating to significantly develop the technology and maximize its positive impact on society. The future of IoT will have a significant impact on our social and economic lives. Therefore, maintaining their security is crucial. Systems detecting intrusions are effective at identifying potential security concerns and breaches. However, progress required addressing the limitations and shortcomings of current technologies, as IoT generates vast amounts of data to enhance customer service.

The role of big data analytics is also being explored for its ability to provide accurate estimates that can improve IoT systems. Beyond monitoring and halting undesirable traffic patterns, cyber IoT security must be able to detect attacks on the network. Numerous researchers have proposed various IoT network infrastructure solutions for monitoring attack traffic flow using ML techniques. Today, machine learning is increasingly being utilized across industries due to is capability to analyze data and make predictions.. Machine learning algorithms are organized into three broad categories depending on how they process data. Supervised learning, unsupervised learning and reinforcement learning are the main categories of machine learning. The fundamental challenge of developing an ML network-based DDoS assault detection system using recorded data lies in addressing data inconsistency. Despite collecting substantial volumes of DDoS traffic information from IoT systems infected with honeypots, the amount of clean data obtained from IoT devices is often limited.

This data imbalance problem must be resolved to achieve reliable detection performance. To address this issue, the unique dataset Bot-IoT dataset is presented in this thesis. The dataset comprises both conventional IoT traffic and various types of attack traffic typically used by Botnets. For future classification usage, this dataset was constructed on a real-world test based and labeled with attributes that show an assault flow, as well as the category and subcategory of intrusions. Additional features were created to improve the classifiers trained on this model’s prediction skill. Through statistical analysis, the top 10 features from the original dataset were identified and used to create a focused subset. This study applies (SVM), (LR), and eXGBoost classifiers to enhance detection accuracy. To improve model reliability, K-fold cross-validation and hyper parameter tuning were implemented. Model performance and dataset validation were assessed using four essential metrics: Accuracy, Precision, Recall, and F1 score.

The eXGBoost model achieved an impressive 88% accuracy on the test dataset. K-fold cross-validation increased accuracy to 99%, whereas hyper parameter optimization resulted in a 92% accuracy rate. The LR model achieved 96% accuracy on the test set, but had a lower k-fold cross-validation score of 87. After hyper parameter adjustment, it improved to 97 percent. The model based on SVM matched LR with 96% accuracy on the test set, but achieved 90% accuracy using k-fold cross-validation. Hyper parameter adjustment improved accuracy to 99.99%, making it the most successful approach for SVM. These findings demonstrate the potential for even improved performance with more improvements. Looking ahead, a deep learning-based network forensic model will be developed using the BoT-IoT dataset, with a focus on evaluating its reliability. In the near future, the BoT-IoT dataset will be utilized to develop a network forensic model using deep learning, and the model's reliability will be evaluated.

REFERENCES

Al-Kasassbeh, M., Almseidin, M., Alrfou, K., & Kovacs, S. (2020). Detection of IoT-botnet attacks using fuzzy rule interpolation. Journal of Intelligent and Fuzzy Systems, 39(1), 421– 431. https://doi.org/10.3233/JIFS-191432.

Alothman, Z., Alkasassbeh, M., & Al-Haj Baddar, S. (2020). An efficient approach to detect IoT botnet attacks using machine learning. Journal of High Speed Networks, 26(3), 241–254. https://doi.org/10.3233/JHS-200641.

Alrashdi, I., Alqazzaz, A., Aloufi, E., Alharthi, R., Zohdy, M., & Ming, H. (2019). AD-IoT: Anomaly detection of IoT cyberattacks in smart city using machine learning. 2019 IEEE 9th Annual Computing and Communication Workshop and Conference, CCWC 2019, 305–310. https://doi.org/10.1109/CCWC.2019.8666450

Alshamkhany, M., Alshamkhany, W., Mansour, M., Khan, M., Dhou, S., & Aloul, F. (2020). Botnet Attack Detection using Machine Learning. Proceedings of the 2020 14th International Conference on Innovations in Information Technology, IIT 2020, February, 203–208. https://doi.org/10.1109/IIT50501.2020.9299061

Baig, Z. A., Sanguanpong, S., Naeem, S., & Vo, V. N. (2020). Averaged dependence estimators for DoS attack detection in IoT networks. Future Generation Computer Systems, 102, 198– 209. https://doi.org/10.1016/j.future.2019.08.007

Faysal, J. Al, Mostafa, S. T., Tamanna, J. S., Mumenin, K. M., Arifin, M. M., Awal, M. A., Shome, A., & Mostafa, S. S. (2022). XGB-RF: A Hybrid Machine Learning Approach for IoT Intrusion Detection. Telecom, 3(1), 52–69. https://doi.org/10.3390/telecom3010003

Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50, 1–21. https://doi.org/10.1016/j.jisa.2019.102419

Ge, M., Fu, X., Syed, N., Baig, Z., Teo, G., & Robles-Kelly, A. (2019). Deep learning-based intrusion detection for IoT networks. Proceedings of IEEE Pacific Rim International Symposium on Dependable Computing, PRDC, 2019-Decem, 256–265. https://doi.org/10.1109/PRDC47002.2019.00056

Gendreau, A. A., & Moorman, M. (2016). Survey of intrusion detection systems towards an end to end secure Internet of things. Proceedings - 2016 IEEE 4th International Conference on Future Internet of Things and Cloud, FiCloud 2016, 84–90. https://doi.org/10.1109/FiCloud.2016.20

Haripriya, L., & Jabbar, M. A. (2018). Role of Machine Learning in Intrusion Detection System: Review. Proceedings of the 2nd International Conference on Electronics, Communication and Aerospace Technology, ICECA 2018, Iceca, 925–929. https://doi.org/10.1109/ICECA.2018.8474576

Ibitoye, O., Shafiq, O., & Matrawy, A. (2019). Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks. 2019 IEEE Global Communications Conference, GLOBECOM 2019 - Proceedings, May. https://doi.org/10.1109/GLOBE COM38437.2019.9014337

Injadat, M. N., Moubayed, A., & Shami, A. (2020). Detecting Botnet Attacks in IoT Environments: An Optimized Machine Learning Approach. Proceedings of the International Conference on Microelectronics, ICM, 2020-Decem(Icm), 2020–2023. https://doi.org/10.1109/ICM50269.2020.9331794

"Internet of Things Global Standards Initiative."

Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., & Alazab, A. (2019). A novel ensemble of hybrid intrusion detection system for detecting Internet of things attacks. Electronics (Switzerland), 8(11). https://doi.org/10.3390/electronics8111210s

Koroniotis, N., Moustafa, N., & Sitnikova, E. (2020). A new network forensic framework based on deep learning for Internet of Things networks: A particle deep framework. Future Generation Computer Systems, 110, 91–106. https://doi.org/10.1016/j.future.2020.03.042

Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. Future Generation Computer Systems, 100, 779–796. https://doi.org/10.1016/j.future.2019.05.041

Kumar, S., Tiwari, P., & Zymbler, M. (2019). Internet of Things is a revolutionary approach for future technology enhancement: a review. Journal of Big Data, 6(1). https://doi.org/10.1186/s40537-019-0268-2

K. Chopra, K. Gupta, and A. Lambora, "Future Internet: The Internet of Things-A Literature Review," in Proc. Int. Conf. Mach. Learn. Big Data, Cloud Parallel Comput. Trends, Prespectives Prospect. Com. 2019, pp. 135–139, Institute of Electrical and Electronics Engineers Inc., feb 2019.

Luo, T., & Nagarajany, S. G. (2018). Distributed anomaly detection using autoencoder neural networks in WSN for IoT. IEEE International Conference on Communications, 2018-May(May). https://doi.org/10.1109/ICC.2018.8422402

Özer, E., İskefiyeli, M., & Azimjonov, J. (2021). Toward lightweight intrusion detection systems using the optimal and efficient feature pairs of the Bot-IoT 2018 dataset. International Journal of Distributed Sensor Networks, 17(10). https://doi.org/10.1177/15501477211052202

Peterson, J. M., Leevy, J. L., & Khoshgoftaar, T. M. (2021). A Review and Analysis of the Bot-IoT Dataset. Proceedings - 15th IEEE International Conference on Service-Oriented System Engineering, SOSE 2021, 20–27. https://doi.org/10.1109/SOSE52839.2021.00007

Pokhrel, S., Abbas, R., & Aryal, B. (2021). IoT Security: Botnet detection in IoT using Machine learning. 1–11. http://arxiv.org/abs/2104.02231

Shafiq, M., Tian, Z., Bashir, A. K., Du, X., & Guizani, M. (2021). CorrAUC: A Malicious Bot- IoT Traffic Detection Method in IoT Network Using Machine-Learning Techniques. IEEE Internet of Things Journal, 8(5), 3242–3254. https://doi.org/10.1109/JIOT.2020.3002255

Shafiq, M., Tian, Z., Sun, Y., Du, X., & Guizani, M. (2020). Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for Internet of things in smart city. Future Generation Computer Systems, 107,433–442. https://doi.org/10.1016/j.future.2020.02.017

Shafiq, M., Yu, X., Bashir, A. K., Chaudhry, H. N., & Wang, D. (2018). A machine learning approach for feature selection traffic classification using security analysis. Journal of Supercomputing, 74(10), 4867–4892. https://doi.org/10.1007/s11227-018-2263-3

Soe, Y. N., Mada, U. G., Mada, U. G., Hartanto, R., & Mada, U. G. (2019). Rule Generation for Signature Based Detection Systems of Cyber Attacks in IoT Environments | Soe | Bulletin of Networking, Computing, Systems, and Software. 8(2), 93–97. http://ww.bncss.org/index.php/bncss/article/view/113/117

Soe, Y. N., Santosa, P. I., & Hartanto, R. (2019). DDoS Attack Detection Based on Simple ANN with SMOTE for IoT Environment. Proceedings of 2019 4th International Conference on Informatics and Computing, ICIC 2019, 0–4. https://doi.org/10.1109/ICIC47613.

Galal, Ahmed & Mabrouk, Abdelkader & M.Ameen, Hawzhen & Ameen, M & Abbas, Munawar & Ling, Dennis & Ching, Chuan & Sajjad, Mohammad Saqlain & Faqihi, Abdullah & Kolsi, Lioua & Ali, Abid & Khan, Ilyas. (2025). Optimizing flow and heat transfer in industrial processes: The potential of trihybrid nanofluid and thermal-radiation using Hamilton-Crosser and Xue models. Journal of Radiation Research and Applied Sciences. 18. 11. 10.1016/j.jrras.2025.101322.

Galal, Ahmed & Abbas, Munawar & Okasha, Mostafa & Alazman, Ibtehal & Bayz, Dyana & Alqahtani, Abdulrahman & Khan, Ilyas & Sajjad, Mohammad Saqlain. (2025). Optimizing thermal and solutal dynamics trihybrid nanofluid fluid flow in industrial processes: The potential of thermal radiation and chemical reaction. 10.1016/j.jrras.2025.101413.

Sajjad, Mohammad Saqlain & Ghazi, Hafiz Muhammad & Nadeem, Muhammad & Habib, Muhammad Irfan & Saeed, Muhammad & Saeed, & Ali, Syed & Naqvi, Hasnain & Arfeen, Zeeshan & Naeem, Isheeaq & Irfan, Muhammad. (2024). Identification of Fake Contents Using Text-mining Techniques. International Journal of Innovations in Science and Technology. 6. 2084-2103.

Habib, Muhammad Irfan & Ghazi, Hafiz Muhammad & Sajjad, Mohammad Saqlain & Khan, Muneeb & Aslam, Farooq & Mushtaq, Muhammad & Hussain, Sayyid & Bhatti, Hassnain & Naeem, Ishteaq & Nadeem, Muhammad & Asgher, Muhammad. (2024). Design, Development, and Deployment of an IoT-Enabled Solar Energy Meter. Journal of Computing & Biomedical Informatics. 07. 606-617.

Ghazi, Hafiz Muhammad & Sajjad, Mohammad Saqlain & Ali, Taha & Muhammad, Durr & Mushtaq, Muhammad & Nadeem, Muhammad & Hussain, Sayyid & Asgher, Muhammad & Qadri, Salman. (2024). Logistic Boosted Algorithms for Securing Smart Homes Against Anomalies and Security Attacks.