Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 4, 2026

IoT Botnet Detection Using Artificial Intelligence

Hazrat Shah Afridi[*1], Alfahad Bin Jan[2], Adil Ali Raja[3], Saffiullah[4], Imran Fareed Nizami[5]

[*1,2,3,4,5]*Department of Electrical Engineering, Bahria School of Engineering and Applied Sciences H-11 Islamabad, Pakistan*

[*1]afridihazratshah@gmail.com, [2]alfahadbinjan10@gmail.com, [3]adilali.buic@bahria.edu.pk, [4]saffiullah465@gmail.com,[5]imnizami.buic@bahria.edu.pk

DOI: https://doi.org/10.5281/zenodo.20266369

Keywords
Internet of Things, Botnet Detection, Machine Learning, Support Vector Machine, UNSW-NB15, Network Security.

Article History
Received: 12 February 2026
Accepted: 22 March 2026
Published: 09 April 2026

Corresponding Author: *
Adil Ali Raja

Abstract

Recent findings draw the important security concern of botnets to the IoT devices and intensified by the surge in the number of connected devices and the swift development of advanced botnet networks. This paper is dedicated to the implementation of machine learning and deep learning algorithms to detect and classify botnet attacks in the IoT setting based on data, including UNSW-NB15. The machine learning pipeline provided is a complete cycle which includes exploratory data analysis and preprocessing of data, training, testing and evaluation of the data with different Machine Learning Algorithms. It is important to note here that the SVM model showed a 99.06% accuracy when classifying network traffic data as being normal or malicious with a slightly lower F1 score of 95.52. This model is the best in striking the right balance between accuracy and recall, correctly recognizing true positives and true negatives and giving a false alarm rate of 0.93% percent to produce few false positives and the classification of benign activities as detrimental. This approach will help in detecting botnets attack proactively and provide a preemptive approach to prevent future attacks. The system, based on machine learning methods, is effective at identifying and classifying botnet attacks, and is scalable, and can scan more than one IoT object at a time, even botnet threats it hadn't known before. Its convenient design allows it to integrate with the existing IoT devices with ease. Overall, the given suggested solution is a solid way of identifying botnet attacks and mitigating them in the IoT environment. Using the strength of machine learning, the system offers scalable and efficient detection functionality and attempts to protect IoT devices against possible botnet threats and improve the overall security levels.

I. INTRODUCTION

The Internet of Things (IoT) has experienced exponential growth, enabling the interconnection of smart devices such as sensors, cameras, household appliances, and industrial controllers. While IoT technologies offer significant benefits across healthcare, transportation, energy management, and smart cities, their rapid adoption has introduced serious security challenges. Many IoT devices suffer from limited computational resources and weak security mechanisms, making them attractive targets for cyber attackers.

Botnets such as Mirai, Bashlite, and Reaper exploit these vulnerabilities to compromise IoT devices and launch large- scale attacks, including distributed denial-of-service (DDoS) attacks. Traditional rule-based intrusion detection systems often fail to detect evolving and previously unseen botnet behaviors. Consequently, intelligent data-

driven approaches based on machine learning have gained significant attention for IoT botnet detection.

This study investigates the effectiveness of machine learning techniques for identifying botnet traffic in IoT environments. By leveraging the UNSW-NB15 dataset, the proposed framework aims to improve detection accuracy while maintaining a low false alarm rate.

A. LITERATURE REVIEW

The technologies behind IoT are too wide and have numerous sensors and communication protocols as well as data processing algorithms. The sensor technologies range between the simplest temperature sensors and to more sophisticated industrial monitoring systems, all of which have their own list of operation requirements as well as security issues [1]. Over the last years, multiple research works have been carried out to prove the efficiency of using the methods of Machine Learning and Deep Learning to detect botnet attacks that have been increasing. Some research is interested in determining the key aspects or features of a botnet that can help distinguish a botnet attack and normal traffic. As an example, Dong et al. investigated the application and effectiveness of machine learning in the botnet detection. They investigated the botnet architecture to determine some pertinent attributes that may distinguish botnet traffic and normal traffic [2]. The feature can then be used to filter out the most useful features to the machine learning model. The knowledge of the underlying properties of botnets will enable researchers to come up with superior and precise detection mechanisms. These peculiarities should be discovered as one of the key steps of developing efficient machine learning based solutions to fight the increasing threat of botnet attacks in the IoT environment. Findings of researchers such as the one made by Dong et al. could be used to develop superior machine learning algorithms capable of detecting and classifying botnet operations with a high degree of accuracy with the ultimate aim of improving the security of internet of things, devices and networks [3]. As the use of Internet of Things, devices are becoming more popular, efforts to mitigate the security threats they present are urgently required. Vishwakarma et al. used honeypot mechanism to attract attackers and collect information in an IoT network. This new information was then processed to examine different attack areas including IP addresses, MAC addresses, packet sizes among others. Also, Guerra-Manzanar and colleagues proposed to use mixed feature selection model to shrink the feature set and enhance precision. Having 115 features in the dataset, which is a considerable sum, the filter, wrapper and hybrid models of feature selection techniques were used to help reduce the dataset. These characteristics were fed to K-Nearest Neighbor (K-NN) and Random Forest, with the resulting successful accuracies of 99%. Interestingly, Decision Tree (DT) Classifier has demonstrated potential in identifying the P2P botnets. Haqu and Singh examined the different classifiers and clustering algorithms used to identify botnets with the help of a dataset that consists of approximately 38,000 network traffic records (both normal and attack traffic). The DT classifier in their study had the highest level of accuracy at 90.2723, and the Decision Tree classifier came next with 87.7853. Moreover, the problems of detecting P2P botnets because of their centralized and dispersed properties were pointed at by Khan et al.

Khan et al. (2013) suggested a P2P botnet detection method that included two phases in 2013. The second step entailed the port assessment, DNS query, and data flow analysis to filter non-P2P traffic [5]. The second step used session features to simplify the process of packet analysis. Traffic identification and classification was done using machine learning algorithms. The experiment has used the CTU dataset, which has 13 unique botnet samples. There were three main ML algorithms that paid attention to session-related features in order to detect P2P botnet traffic: Naïve Bayes (NB), Decision Tree (DT) classification, and Artificial Neural Network (ANN). The findings showed that NB had a 75.5% detection rate, ANN had 93.8% accuracy and DT algorithm had 94.4% accuracy. This highlights the efficiency of the two-stage method which combines the use of P2P traffic filtering and

the use of DT classifier basing on the characteristics of the session in effectively identifying P2P botnet traffic [6].

B. DATASET DESCRIPTION:

UNSW-NB15 dataset is a popular dataset in the network security and intrusion detection arena. It has network traffic information which is used to develop and test intrusion detection systems. The description of the UNSW-NB15 dataset with its characteristics and samples is presented below in details:

Features: The dataset consists of a variety of features that capture different aspects of network traffic behavior. These features include:

- Simple flow attributes such as duration, type of protocol, service, source and destination IP addresses, source and destination port.
- Content features such as the number of bytes and packets sent/received, flags and payload data. Statistical values such as mean, standard deviation, minimum, maximum, and total number of attributes of the network traffic.
- Statistical features like mean, standard deviation, minimum, maximum, and sum of various network traffic attributes.
- Time-based features that store time-related data on network traffic.

Samples: The data set has numerous samples which depict the occurrence of network traffic data. The dataset is associated with a network communication session or event through each sample. To evaluate and train the intrusion detection models, the samples are categorized as normal and malicious [7].

Labeling: UNSW-NB15 dataset is labeled to show whether it is normal network traffic or malicious activity. Such labeling plays an important role in training machine learning algorithms with the goal of identifying normal and harmful actions.

Usage: UNSW-NB15 data are utilized to create, test, and develop intrusion detection systems by researchers and practitioners. Machine learning models can be trained using the characteristics and tags of the dataset to identify and categorize the various types of cyber-attacks in the network traffic data accurately 8].

Evaluation Metrics: The common metrics that are used to assess the performance of intrusion detection models are accuracy, precision, recall, F1 score, and false alarm rate, which are commonly assessed on the dataset. These measures assist in testing the efficacy of the models in identifying and treating the security threats. UNSW-NB15 is a valuable resource to researchers and practitioners in the context of cybersecurity because the dataset is realistic and diverse with respect to network traffic data on network to develop and test intrusion detection systems [9].

C. PROPOSED SYSTEM

The proposed system would apply methods of artificial intelligence (AI) and specifically machine learning algorithms (MLAs) to fill the gaps of the current systems. The following are the techniques used:

Feature Selection: The identification of useful features in the data is a very important part in intrusion detection system performance. Finding the most informative features and eliminating noise and irrelevant information is a rather complicated task that should be thoroughly analyzed and has to be well-informed about the domain [10].

Logistic Regression: The UNSW-NB15 dataset uses the Logistic Regression due to its capability of providing the probability of a binary outcome, which is appropriate to discriminate normal and malicious network traffic. Logistic Regression is useful in intrusion detection systems because it is simple, interpretable, and efficient when dealing with huge datasets such as the UNSW-NB15 dataset. Logistic Regression can give insights into the possibility of a specific network behavior being malicious, assisting in the detection of prospective cyber-attacks. Furthermore, Logistic Regression is widely used in cybersecurity research due to its efficacy in modelling the link between input variables and the likelihood of a given event occurring, which is consistent with the goal of identifying intrusions in network traffic data. Logistic Regression's ability to provide probabilistic results and analyze the influence of multiple characteristics on the outcome makes it

an important tool for developing intrusion detection systems that can properly identify network traffic as normal or malicious [11].

Linear Support Vector Machine: Support Vector Machine (SVM) is applicable to intrusion detection in the UNSW-NB15 dataset due to its capability to handle high dimensional data, capability to deal with both linear and nonlinear relationships within the data and also capability to identify difficult decision boundaries. Because SVM is rather effective to classify network traffic data into many categories, it is especially appropriate in this data as it can be utilized to detect abnormalities and likely cyber-attacks [12]. Network traffic data can have legitimate and malicious activities, and in this case, the imbalance data is well controlled by SVM in the area of intrusion detection systems. SVM has the ability to process massive data and extrapolate well to new, unknown data and hence is a beneficial tool to be used by researchers dealing with the UNSW-NB15 dataset. SVM would be an excellent choice of anomaly detection and proper classification of network traffic due to the flexibility to the range of kernel functions, which also enables it to identify more complicated patterns in data.

Decision Tree: UNSW-NB15 dataset utilizes decision trees to detect intrusion because it is readable, easy to apply and effective in handling categorical and numerical types of data.

They particularly fit this dataset because decision trees can be used with a wide range of types of features, and thus can be used with network traffic data. Moreover, it is known that decision trees are able to process large quantities of data successfully and are less prone to overfitting compared to more advanced models [13].

Random Forest Classifier: Because it can handle high- dimensional data, imbalanced datasets, and attain high accuracy rates, the Random Forest Classifier is used in the UNSW-NB15 dataset for intrusion detection and classification. The Random Forest algorithm is particularly good at digesting complex network traffic data and seeing patterns linked to harmful behavior in the context of network security and intrusion detection. Because Random Forest is ensemble in nature, that is, it combines the outputs of several decision trees, it may generate forecasts that are both reliable and accurate.

Additionally, because the Random Forest Classifier can swiftly handle both binary and multi-class classification jobs, it is highly beneficial for the UNSW-NB15 dataset. Researchers may use Random Forest in a distributed Big Data environment like Apache Spark to address the problems caused by excessive dimensionality and uneven data in the dataset. Moreover, the Random Forest model's capacity to recognize and classify network assaults effectively may be enhanced by adjusting hyper parameters as the number of estimators, maximum features, and tree depth [14].

D. DATABASE DESIGN

The 49-feature dataset was reduced to 17 features using feature engineering approaches. In order to construct a dataset, 257673 rows and 18 columns were needed for the modelling stage. Two variables made up this dataset: X held the target variable and only network traffic information, whereas Y included both. Subsequently, training and testing sets comprising 70% and 30% of X and Y were divided.

A range of categorization methods were employed to train the model using the training data. The number of dimensions was decreased by applying Spearman's correlation method. The monotonic relationship between two data points may be statistically measured using the Spearman's correlation coefficient [15].

Spearman's correlation is sometimes categorized as a non- parametric statistic since it does not need normality, in contrast to Pearson's correlation. With Python, the dataset with 44 characteristics was reduced to 17 critical features, which was sufficient for IoT botnet traffic prediction based on the correlation coefficients from the correlation matrix. More insight into the network traffic statistics utilized in this study is provided by the graph.

II. SYSTEM TESTING

The classification model's state of confusion during a prediction is illustrated by the confusion

matrix, which may be used to infer the following information:

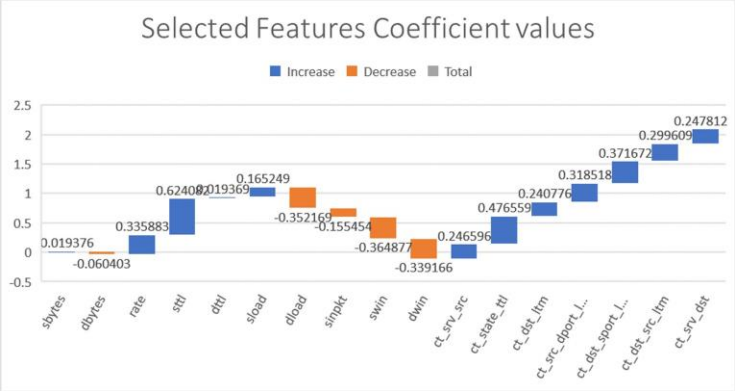

FIGURE 1. Plot showing the extracted features

	Predicted Normal Traffic	Predicted IoT Botnet Traffic
Actual Normal Traffic	TN	FP
Actual IoT Botnet Traffic	FN	TP

FIGURE 2. Confusion Matrix

The extracted attributes and their correlation coefficient values are listed in the table below. Additionally, tree-based feature selection was utilized to identify the pertinent network properties in the dataset. The results, however, showed that the Spearman's correlation feature selection method provided greater details on the characteristics of network traffic. Consequently, the IoT botnet detection model was given all 17 recovered features, enabling faster and more accurate detection.

III. EVALUATION

The numerous tests that have been applied to determine the accuracy and performance of the categorization models that were employed in this research study are discussed in this section. The SVM model adequately identified network traffic data as normal or malicious at the rate of 99.06, which enabled the identification of the IoT botnet to be successful. The model showed a good balance between the accuracy and recall despite the much lower F1 score of 95.52%. This implies that the model can be able to reliably determine both false negatives and true positives. The SVM model is also able to restrict false positive detection and reduce the rate of misclassifying non-dangerous activities as dangerous activities as its false alarm rate is low at 0.93%.

sbytes	sloss	0.9515464133660682
dbytes	dloss	0.9912941785427634
dbytes	dpkts	0.9705744719061106
sttl	ct_state_ttl	0.9058027624589845
sttl	label	0.9043459910087561
dloss	dpkts	0.9921743916746946
swin	dwin	0.9971933359586712
stime	ltime	0.9999999998073185
tcprtt	synack	0.9332414134584021
tcprtt	ackdat	0.9202047490089745
ct_srv_src	ct_srv_dst	0.956721026945663
ct_srv_src	ct_dst_src_ltm	0.942148711061324
ct_srv_dst	ct_dst_src_ltm	0.9510250540010916
ct_dst_ltm	ct_src_ltm	0.9384612565049639
ct_dst_ltm	ct_src_dport_ltm	0.9601144948755526
ct_src_ltm	ct_src_dport_ltm	0.9453045008443021
ct_src_dport_ltm	ct_dst_sport_ltm	0.9214458874161181
ct_src_dport_ltm	ct_dst_src_ltm	0.9109191341192471

FIGURE 3. Selected Features

Selected Features	Feature Description
sbytes	Bytes transmitted from source to destination.
dbytes	Bytes transmitted from destination to source
rate	Transmission rate
sttl	source time to live value
dttl	Destination time to live value
sload	Source bits per second
dload	Destination bits per second
sinpkt	Source interpacket arrival time (msec)
swin	Source TCP window value
dwin	Destination TCP window value
ct_srv_src	No. of connections that contain the same service and source
ct_state_ttl	No. of connections of the same state and the time to live value
ct_dst_ltm	No. of connections of the same destination
ct_src_dport_ltm	No. of connection of the same source address and the destination port.
ct_dst_sport_ltm	No. of connections of the same destination address and the source port
ct_dt_src_ltm	No. of connections of the same source and the destination address
ct_srv_dst	No. of connection that contain the same service and destination address

FIGURE 4. Description of Selected Features

Dataset	Model	AUC	F1-score	False Alarm Rate
Train	LR	0.9878791093434393	0.9585322685085258	0.012120890656560756
Test	LR	0.9875316737082718	0.9581082448153768	0.012468326291728213

FIGURE 5. Code snippet of Logistic Regression

Dataset	Model	AUC	F1-score	False Alarm Rate
Train	SVM	0.9907030504693602	0.955294589640227	0.009296949530639703
Test	SVM	0.9906682443423142	0.9552035837133029	0.009331755657685866

FIGURE 6. Code snippet of Linear Support Vector Machine

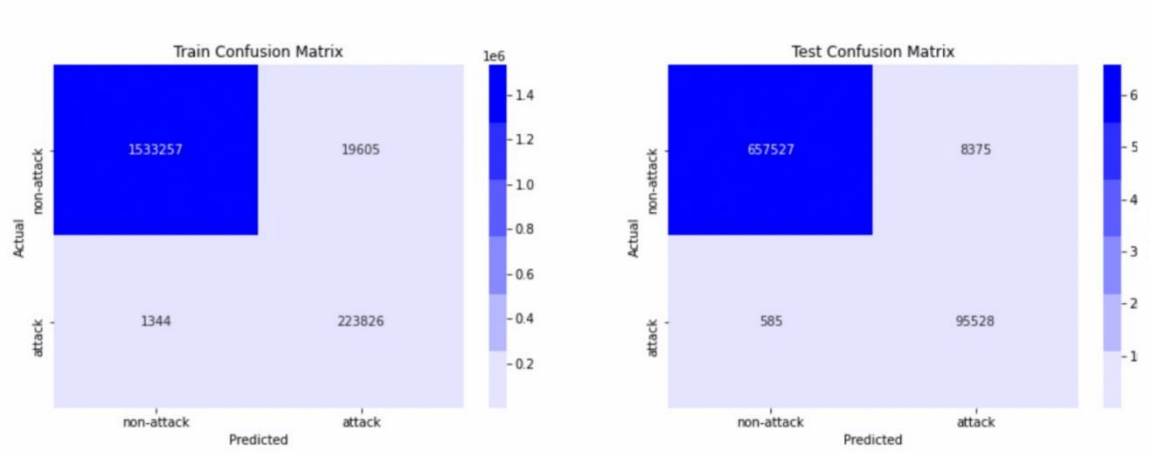

FIGURE 7. Confusion matrix plot of Logistic Regression,

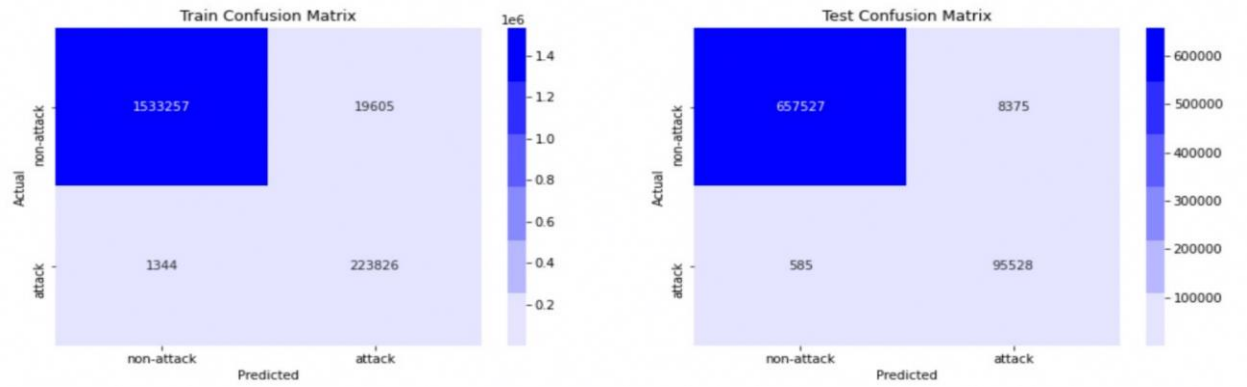

FIGURE 8. Confusion matrix plot of Linear Support Vector Machine

A. LOGISTIC REGRESSION:

The Logistic Regression model showed excellent predictive accuracy of 98.75, F1 accuracy of 95.81 and false alarm rate of 1.24. The system was correct at 98.75% of the cases. This aspect of the model where it can identify the true positives and true negatives and recall both at an equal measure demonstrates its high F1 score of 95.81%. Besides, the model has lower error rates in positive predictions, or the number of occasions where true transactions are false positively detected, as shown by the low false alarm rate of 1.24. This impressive result shows the efficiency and accuracy of the Logistic Regression algorithm that recognizes and classifies events of interest, and it is worth using in such activities as credit card fraud detection.

B. LINEAR SUPPORT VECTOR MACHINE:

It is seen in the UNSW-NB15 data that the Support Vector Machine (SVM) approach to intrusion detection works extremely well. The SVM model with an accuracy of 99.06 showed the ability to properly classify network traffic data as either normal or harmful. Although the F1 score was much lower at 95.52, the model still presented a good balance of precision and recall and proves that the model is able to accurately detect both the real positives and real negatives. Moreover, the lack of false positive prediction as well as the low false alarm rate of 0.93 shows the ability of the SVM model to reduce the frequency with which non-malicious actions are incorrectly classified as malicious.

C. DECISION TREE:

The model of the Decision Tree of the dataset had 98.77% accuracy, 96.28% F1 score, and a false alarm rate of 1.22%. Numerous indications of patients with heart failure were categorized and predicted using the Decision Tree model therefore enhancing their care and prognosis. The Decision Tree model was also found to predict the occurrence of heart failure by optimizing the parameters of the model and its preprocessing of the data proved the model to have good accuracy and recall rates. The results indicate the ability of the Decision Tree to predict and provide valuable sources of information to enhance patient care and outcomes in the management of heart failure.

D. RANDOM FOREST CLASSIFIER MODEL:

The network traffic data of the UNSW-NB15 dataset was correctly recognized by the Random Forest model with a phenomenal precision. Its 98.54 percent rate of identities of most of the events indicates that the model has the ability of differentiating the benign and the malevolent network activity. The model had a high F1 score of 97.67 indicating a good balance between the accuracy and recall. This demonstrates that the Random Forest model could accurately identify true positives as well as false negatives even though rate of false positives and false negatives was minimized. Besides, the false alarm rate by the model of 1.45% indicates that the model was capable of minimizing the number of instances when regular transmission over a network was unaccounted as unfriendly. This is imperative to consider when exploiting intrusion detection systems as a large rate of false alerts may consume resources and cause mistrust in the system.

Dataset	Model	AUC	F1-score	False Alarm Rate
Train	DT	0.9879649532321753	0.9634806494164075	0.012035046767824754
Test	DT	0.9877445111630792	0.9628513698525557	0.012255488836920861

FIGURE 9. Code snippet of Decision Tree

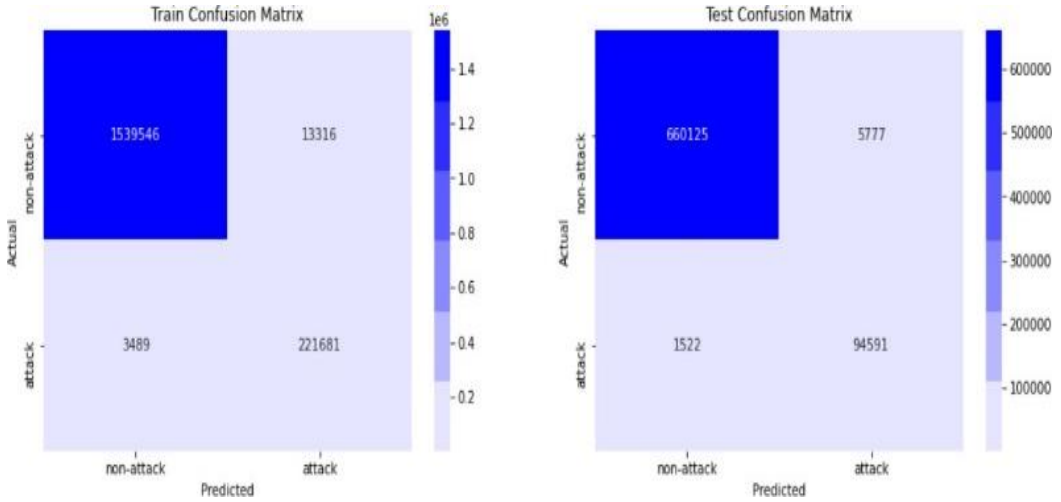

FIGURE 10. Confusion matrix plot of Decision Tree

```
# Getting result on train and test data
evaluate_result(rf_bst_clf, x_train_new_csr, y_train, x_test_csr, y_test, "RF")
```

```
+---------+-------+--------------------+--------------------+---------------------+
| Dataset | Model |        AUC         |      F1-score      |   False Alarm Rate  |
+---------+-------+--------------------+--------------------+---------------------+
|  Train  |   RF  | 0.9927925575846711 | 0.9897532424861752 | 0.007207442415328972 |
|   Test  |   RF  | 0.9854768258366028 | 0.9767504956694146 | 0.01452317416339732 |
+---------+-------+--------------------+--------------------+---------------------+
```

FIGURE 11. Code snippet of Random Forest model

```
# Using entire data
auc, f1, far = final_fun_2(x_test, y_test.values)
```

```
+--------------------+--------------------+---------------------+
|        AUC         |      F1-score      |   False Alarm Rate  |
+--------------------+--------------------+---------------------+
| 0.9862268589810488 | 0.978034935409145 | 0.013773141018951229 |
+--------------------+--------------------+---------------------+
```

FIGURE 12. Code snippet of Prediction of Raw Data

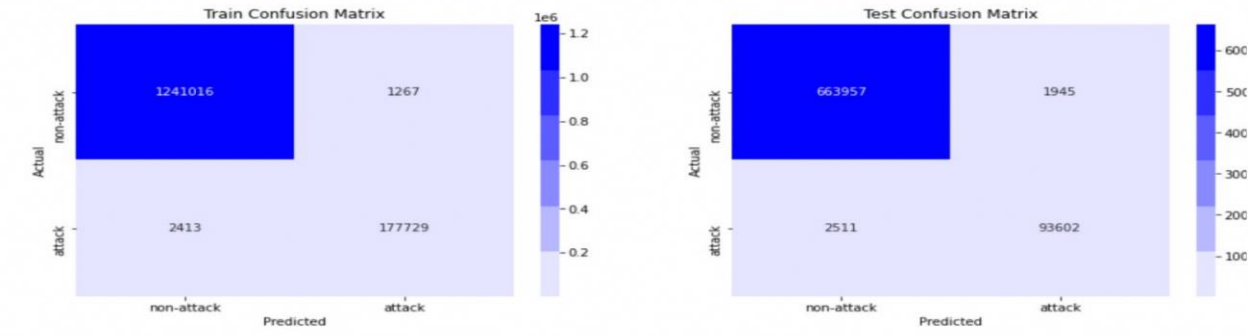

FIGURE 13. Plot of the Random Forest model's confusion matrix,

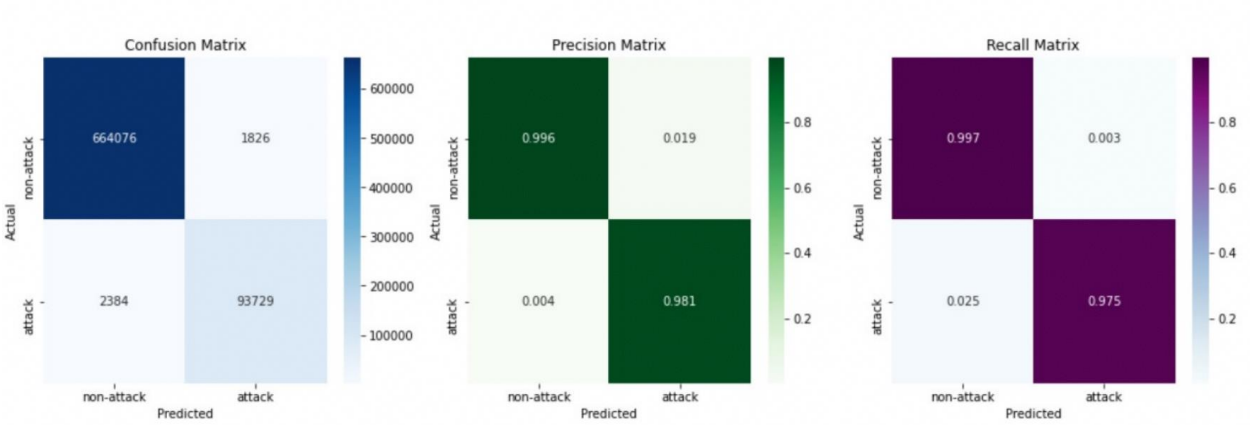

FIGURE 14. Confusion matrix plot of Prediction of Raw Data

IV. PREDICTION OF RAW DATA

The model showed a high level of performance in predicting raw data at an accuracy of 98.62, F1 score of 97.80, and a false alarm rate of 1.37. The model predicted the results in 98.62 percent of the instances. This shows the extent to which the model is discriminatory between favorable and unfavorable conditions. The model is capable of identifying actual positives and minimizing the false negatives and false positives as indicated by the F1 score of 97.80. The false positive rate of the model or the number of times that it actually identifies a valid situation as a fraud is also low at 1.37, which shows the reduction of false positive prediction. The model is useful in a variety of categorization activities because such performance measures demonstrate the model efficiency and credibility in correct predicting of the outcomes.

V. CONCLUSION

To conclude, UNSW-NB15 dataset was a good selection to use in the project that used Support Vector Machine (SVM) to detect intrusions. It was indicated that the SVM model had a high accuracy rate of 99.06% in assigning the network traffic data as malicious or benign. The model was highly able to separate real negatives and true positives with its great accuracy-to-recall ratio, although its F1 score is very low at 95.52%. The SVM model has the capability to restrict false positive predictions and reduce misclassification of innocent activities as dangerous as demonstrated by low false alarm of 0.93. The SVM model has been a powerful and credible tool in terms of intrusion detection in network security applications due to its great accuracy, fair F1 score and low false alarm rate.

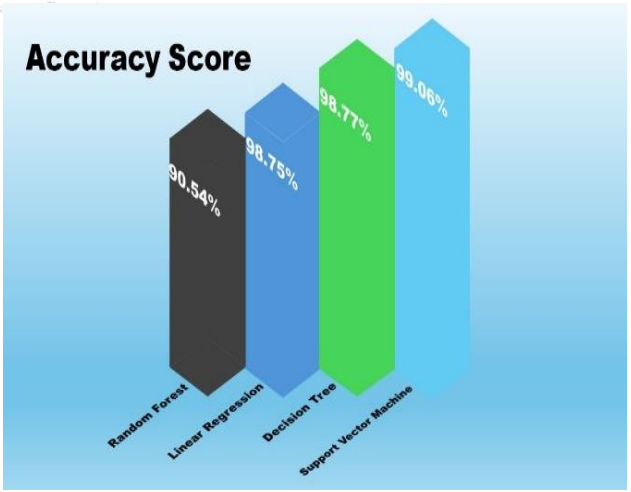

FIGURE 15. Accuracy Score of Selected Features

VI. REFERENCES

[1]Bedi, Guneet, Ganesh Kumar Venayagamoorthy, and Rajendra Singh. "Navigating the challenges of Internet of Things (IoT) for power and energy systems." 2016 Clemson University Power Systems Conference (PSC). IEEE, 2016.

[2]Azab, Ahmad, Mamoun Alazab, and Mahdi Aiash. "Machine learning based botnet identification traffic." 2016 IEEE Trustcom/BigDataSE/ISPA. IEEE, 2016.

[3] K. Alissa et al., "Botnet Attack Detection in IoT Using Machine Learning," Computational Intelligence and Neuroscience, 2022.

[4] Dong, Xiaxin, Jianwei Hu, and Yanpeng Cui. "Overview of botnet detection based on machine learning." 2018 3rd International Conference on Mechanical, Control and Computer Engineering (ICMCCE). IEEE, 2018.

[5] Rostami, Mohammad Reza, Bharanidharan Shanmugam, and Norbik Bashah Idris. "Analysis and detection of P2P botnet connections based on node behaviour." 2011 World Congress on Information and Communication Technologies. IEEE, 2011.

[6] Joshi, Harshvardhan P., and Rudra Dutta. "Identifying p2p communities in network traffic using measures of community connections." 2020 IEEE Conference on Communications and Network Security (CNS). IEEE, 2020.

[7] Z. Alothman, M. Alkasassbeh, and S. Al-Haj Baddar, "An Efficient Approach to Detect IoT Botnet Attacks Using Machine Learning," Journal of Information and Health Sciences, 2020.

[8] J. Kim et al., "Intelligent Detection of IoT Botnets Using Machine Learning and Deep Learning," Applied Sciences, vol. 10, no. 19, 2020.

[9] M. S. B. Judyflavia et al., "IoT Botnet Detection Using Machine Learning," International Journal of Health Sciences, 2022.

[10] Y. Meidan et al., "N-BaIoT: Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders," 2018.

[11] N. Elsayed et al., "IoT Botnet Detection Using an Economic Deep Learning Model," 2023.

[12] A. Kumar et al., "Machine Learning-Based Early Detection of IoT Botnets Using Network-Edge Traffic," 2020.

[13] AI-Driven Botnet Detection in IoT Networks: A Comprehensive Research Review," Computer Science Review, 2026.

[14] Advancing IoT Security: A Systematic Review of Machine Learning Approaches for the Detection of IoT Botnets," Journal of King Saud University, 2023.

[15] X. Dong, J. Hu, and Y. Cui, "Overview of Botnet Detection Based on Machine Learning," ICMCCE, 2018.