A MODEL FOR THE DETECTION OF MULTI-CLASS MALICIOUS TRAFFIC ON NETWORK SYSTEM

M. O. Musa
Department of Cyber Security
Faculty of Computing
University of Port Harcourt
Port Harcourt
Nigeria.
martha.musa@uniport.edu.ng

E. E. Odokuma
Department of Computer Science
Captain Elechi Amadi Polytechnic
Port Harcourt
Rivers State
Nigeria
elizzyodoks@yahoo.com

Abstract- Malicious traffic poses a significant threat to network systems and the security of sensitive data. By staying vigilant, implementing robust security measures, and fostering a culture of cybersecurity awareness, organizations can better protect themselves from these threats and minimize the potential impact of malicious traffic on their networks. This paper presents a model for the detection of malicious traffic on a network system. For detecting multi-class of malicious traffics on the network system, we utilized a dataset that comprise 9 types of malicious attacks on a network system. In other to have an efficient model, we conduct exploratory data analysis on the dataset. The exploratory data analysis was used in checking missing values, correlated features, data imbalance, and also important features. The results of the exploratory data analysis show that the dataset is imbalanced which will lead to overfitting if not resolved. We resolved the data imbalance by performing random oversampling by utilizing the RandomOverSampling technique in python. After resolving the data imbalance, we used a random forest classifier to extract important features of the data. Here, we extracted ten features based on the ranking of the random forest model. The extracted features were used in training the proposed model for the detection of malicious traffic on a network system. The results of the model show a better accuracy for the detection of malicious traffic on a network system with an accuracy of 99.99%, 99.99% for precision, recall, and F1-score.

Keywords *– Malicious Traffics, network systems, cyber-security, random forest classifier*

1. Introduction

In the fast-paced realm of IT/OT refers to information technology and operational technology. systems in our modern world, safeguarding the Internet of Things (IoT) from cyber threats is of utmost importance. The IoT's vulnerability stems from two key factors: the widespread usage of IoT devices, spanning from households to smart grids, transportation systems, and other essential infrastructures, and the diverse methods of transmission employed by these devices. The amount of IoT devices has skyrocketed from 15.4 billion in 2015 to 26.7 billion in 2019, and this trend continues as people and businesses increasingly rely on their use of web-based services for daily lives [1].

The vast array of IoT devices, found in Smart homes, connected cars, smart cities, connected healthcare, and smart industrial IoT , has spurred rapid advancements in IoT technology and shortened product development cycles. Consequently, there is a need for a unified platform to enable seamless communication between these devices, considering the different communication standards they employ. Moreover, the tools and experimental platforms available for running IoT devices, and simulations at the network level are growing [2].

MQTT (Message Queuing Telemetry Transport) is a protocol for two-way communication that is simple and efficient enables remote connectivity between IoT devices and centralized

brokers. While MQTT's lightweight nature makes it suitable for mobile applications with low resource consumption, it does have vulnerabilities that potential adversaries could exploit. These threats encompass device compromise, breaches of data privacy, Denial of Service (DoS) attacks targeting MQTT services, and Man-in-The-Middle (MiTM) attacks on MQTT messages [3].

Reconnaissance attacks, also known as information gathering attacks, are a critical initial step for hackers attempting to breach a target system or network. During reconnaissance attacks, cybercriminals gather as much information as possible about the target, including its vulnerabilities, network topology, and potential points of entry. Common techniques include port scanning, network mapping, and social engineering methods like phishing. The data obtained during this phase helps attackers plan and execute subsequent stages of their attack, increasing the chances of a successful breach [4].

To counter reconnaissance attacks effectively, machine learning plays a crucial role. Machine learning-powered security systems can analyze vast amounts of data, identifying patterns and anomalies that might be overlooked by humans. These systems track and detect unusual network behaviors, recognize potential attackers' patterns, and differentiate between legitimate and suspicious activities in large-scale network traffic. Moreover, machine learning algorithms can continuously improve their detection capabilities by learning from new data, thus strengthening organizations' defenses against reconnaissance attacks and preventing them from escalating into more significant security breaches [5].

2. Review of Related Literatures

In a previous study [6], scholars have introduced a novel approach to identify and detect unauthorised access in Internet of Things (IoT) networks. A novel methodology has been adopted whereby deep learning principles are employed to effectively categorise patterns of traffic movement. The researchers utilised a recently published dataset on the Internet of Things (IoT) and retrieved pertinent informational characteristics included in the packet-level fields. In order to tackle the challenges of categorizing data into two or more groups, the researchers have devised a feed-forward neural network model that exhibits the ability to detect a range of assaults on IoT devices. These attacks encompass denial of service, distributed denial of service, reconnaissance, and information theft. The evaluation of this approach on the processed dataset reveals a notable level of classification accuracy.

In the subsequent section, the paper presents a framework model that aims to address the issue of identifying malicious traffic within Internet of Things (IoT) networks. The present methodology provides a novel metric for feature selection, denoted as CorrAUC, and proposes an algorithm that leverages this metric by utilising the area under the curve (AUC) as a performance measure. This solution effectively integrates the Entropy and the TOPSIS measure methodologies by employing soft set that is bijective to detect fraudulent data in IoT. The efficacy of this approach is demonstrated through an experiment conducted on the Bot-IoT dataset, employing four distinct machine learning algorithms. The results reveal a noteworthy average classification accuracy over 96%.

In reference [8], a fresh and stimulating strategy has been introduced for the analysis of IoT malware traffic. This approach employs deep learning and visual representation techniques,

resulting in enhanced efficiency in the detection and classification of emerging malware, commonly known as zero-day malware. This approach focuses on the identification of malicious network traffic at the packet level, resulting in a notable reduction in detection time by employing advanced deep learning technology. The utilisation of Residual Neural Network (ResNet50) in the conducted experiment has yielded highly encouraging outcomes, with a detection accuracy rate of 94.50% achieved for identifying malware traffic.

Now, let us redirect our focus to reference [9], in which the authors put forth a mechanism for identifying malicious communication by employing a combination of Support Vector Machine (SVM) and Convolutional Neural Network (CNN). Although both approaches demonstrate satisfactory outcomes with a minimal occurrence of false positives, it is evident that the SVM method surpasses the CNN method in terms of all evaluation metrics. The study also suggests potential future research directions, such as investigating the impact of varying transport layer size and direction as features, and utilising a Convolutional Neural Network (CNN) enhanced with a Long Short-term Memory (LSTM) for automated feature engineering in order to identify malicious network traffic.

The present study [10] proposes a two-layer approach for the detection of malware in Android applications. The initial layer of the system employs a static malware detection model that relies on permission, intent, and component information. This is achieved through the utilisation of a fully connected neural network. The subsequent layer presents a novel methodology known as CACNN, which effectively integrates Convolutional Neural Networks (CNN) with AutoEncoder techniques to identify malicious software by analysing network traffic characteristics of applications. The empirical findings suggest that the two-layer model has a notable level of efficacy in identifying malware, encompassing both binary classification and identification of malware by category and malicious family.

The subsequent study [11] introduces an innovative approach to detect malicious DNS tunnelling tools by employing a hierarchical classification technique and utilising machine learning algorithms on DNS over HTTPS (DoH) network traffic. The system's prototype underwent a comprehensive evaluation using the CIRA-CIC-DoHBrw-2020 dataset. The evaluation demonstrated notable levels of accuracy in filtering DoH traffic, detecting suspicious DoH traffic, and identifying malicious DNS tunnel tools.

In the subsequent section, the present study investigates the application of network profiling and machine learning techniques in order to enhance the security of Internet of Things (IoT) systems by mitigating the risks associated with cyber-attacks. The suggested anomaly-based intrusion detection solution actively conducts profiling and monitoring of all networked devices, hence efficiently identifying any attempts of tampering with IoT devices and suspicious transactions within the network. The methodology demonstrates encouraging outcomes, attaining an aggregate precision of 98.35% and an exceptionally minimal false-positive ratio of 0.98% when assessed on the Cyber-Trust testbed utilising both benign and malevolent network data.

The authors in reference [13] provide a dataset that comprises real-time SCADA test bed data, encompassing both normal and attack instances. Feature extraction techniques such as Chi-Square, ANOVA, and LASSO are employed to decrease the dimensionality of the feature set. Furthermore, the researchers employ the Support Vector Machine (SVM) variation known as SVMSMOTE to address the issue of imbalanced dataset. The results of the performance test

of four machine learning algorithms indicate that the Support Vector Machine (SVM) algorithm, when combined with filtering and Synthetic Minority Over-sampling Technique (SVMSMOTE), demonstrates superior performance compared to the other approaches. This combination achieves a remarkable Receiver Operating Characteristic (ROC) value of 99.96%.

Finally, the aforementioned research [14] presents an anomaly traffic detection mechanism known as D-PACK, which has demonstrated significant efficacy. The proposed approach integrates a Convolutional Neural Network (CNN) and an unsupervised deep learning model known as an Autoencoder to perform traffic profiling and filtering. The D-PACK method demonstrates exceptional performance in identifying hostile flows by analysing only the initial two packets. It achieves a remarkable accuracy rate of nearly 100% while maintaining an impressively low false-positive rate of 0.83%. Consequently, D-PACK proves to be an exceedingly efficient approach for the purpose of preventing malicious flows.

In their study referenced as [15], the authors address the issue of imbalanced datasets inside intrusion detection systems by the use of diverse methodologies. The authors conduct an empirical evaluation of various machine learning classifiers, including deep neural networks, random forest, voting, variational autoencoder, and stacking, using the CIDDS-001 dataset. The system proposed by the authors demonstrates notable outcomes, attaining a maximum accuracy of 99.99% in the identification of attacks. Additionally, it effectively manages imbalanced class distributions by utilising a reduced number of samples. As a result, it presents a suitable approach for addressing real-time data fusion challenges.

3. Design Methodology

The sub-section describes the architectural design of the proposed system. It also shows various components of the system and how the are interrelated. The architectural design of the proposed system can be seen in 1.

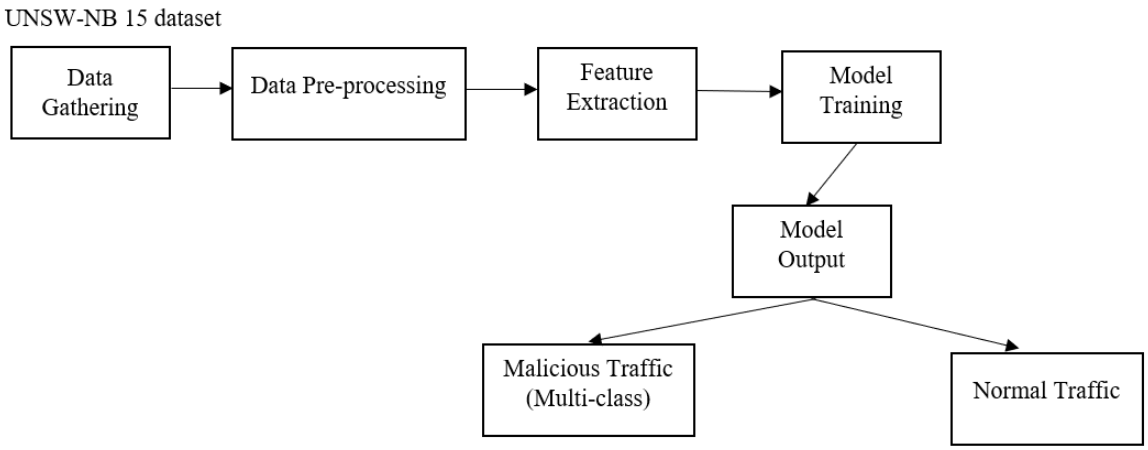

Figure 3.1: Figure 1: Architectural Design

Data Gathering: The dataset utilised in this study was obtained from Kaggle.com and comprises of unprocessed network packets that were collected using the IXIA PerfectStorm

tool at the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS). The objective was to devise a fusion of authentic contemporary routine tasks with artificially constructed contemporary aggressive behaviours. In order to achieve this objective, the Tcpdump utility was utilised to collect a total of 100 gigabytes of unprocessed network traffic, which was afterwards saved in Pcap file format.

The dataset comprises a collection of nine unique categories of attacks, namely Fuzzers, Analysis, Backdoors, Denial of Service (DoS), Exploits, Generic, Reconnaissance, Shellcode, and Worms. The data processing involved the utilisation of the Argus and Bro-IDS tools. This process resulted in the generation of a set of 49 characteristics, each accompanied by its associated class label. These features were generated through the application of twelve different algorithms.

Data Pre-processing: The initial step was examining the dataset for any instances of missing values. The dataset underwent pre-processing to standardise all values. The task was accomplished by employing the MinMaxScaler() function in the Python programming language. The MinMaxScaler is computed by the following formula:

Assumption

The computation of the scaled feature value, represented as x_scaled, involves dividing the discrepancy between the original feature value (x) and the minimum value of the feature in the dataset (x_min) by the range of the feature (x_max - x_min). The aforementioned equation successfully converts the initial feature value into a new value that falls within the interval of [0, 1]. This conversion process involves assigning the minimum value in the dataset to 0 and the maximum value to 1. The remaining values in the dataset are adjusted proportionally to their respective values within the range of [0, 1].

Feature Extraction: This module is responsible for selecting the most important features on the dataset. We used a random forest classifier in selecting the top ten important features according to their ranking. This was done so as to reduce the dimension of the dataset for efficient classification. The mathematical expression to compute feature importance using Random Forest is based on the Gini impurity that each feature reduces in the model's decision-making process. The Gini impurity-based feature importance for feature j is computed as follows:

$$\text{Feature\_Importance\_j} = \frac{\text{Sum Gini\_impurity\_decreases\_at\_all\_nodes\_splitting\_on\_feature\_j)}}{\text{Total number of trees in the Random Forest)}} \quad \text{Eqn. 2}$$

Model Training: The model was trained using Random Forest Classifier.

The mathematical expression for the Random Forest Classifier can be represented as follows:

Let:

i. X be the feature matrix with dimensions (n_samples, n_features), where n_samples is the number of data points, and n_features is the number of features.

ii. y be the target vector with dimensions (n_samples), representing the class labels for each data point.

iii. M be the number of decision trees in the Random Forest.

iv. T_m(X) be the m-th decision tree that takes the feature matrix X as input and produces predictions.

Each decision tree T_m(X) can be represented as a function that maps the feature matrix X to predicted class labels:

$$T\_m(X) = f\_m(X) \quad \text{Eqn. 3.}$$

The final prediction from the Random Forest can be obtained through majority voting:

$$RF(X) = \text{majority\_vote}(T\_1(X), T\_2(X), ..., T\_M(X)) \quad \text{Eqn. 4.}$$

where majority_vote is a function that takes the outputs of individual decision trees T_m(X) and returns the class label with the highest frequency.

4. Experimental Results

We conducted an experiment on Jupyter Notebook, the experimental results is made up of two phases. The phases of the experiments are the Exploratory Data Analysis (EDA) and the training of the Random Forest Classifier for the detection of malicious traffics on network systems.

4.1 Exploratory Data Analysis (EDA)

We conducted an EDA on the dataset in other to have a better insight into the dataset. In Figure 2, we conducted a correlation matrix on the dataset. This was done, to enable us to know features of the dataset that was correlated. Next, we conducted an EDA to know if the dataset is imbalanced. According to the findings presented in Figure 3, it is evident that the dataset exhibits an imbalance, as the number of instances in each class is not equal. In order to address this issue, a random oversampling technique was employed in the Python programming language. This technique involved increasing the number of instances in the minority classes to match that of the majority class. The counplot of the balanced data can be seen in Figure 4. The last part of the EDA was to find the ten most important features of the dataset. The achieved using Random Forest Classifier. This was done by performing ranking on the features of the dataset. From the ranking, the ten most important features can be seen in Table 1 and Figure 5.

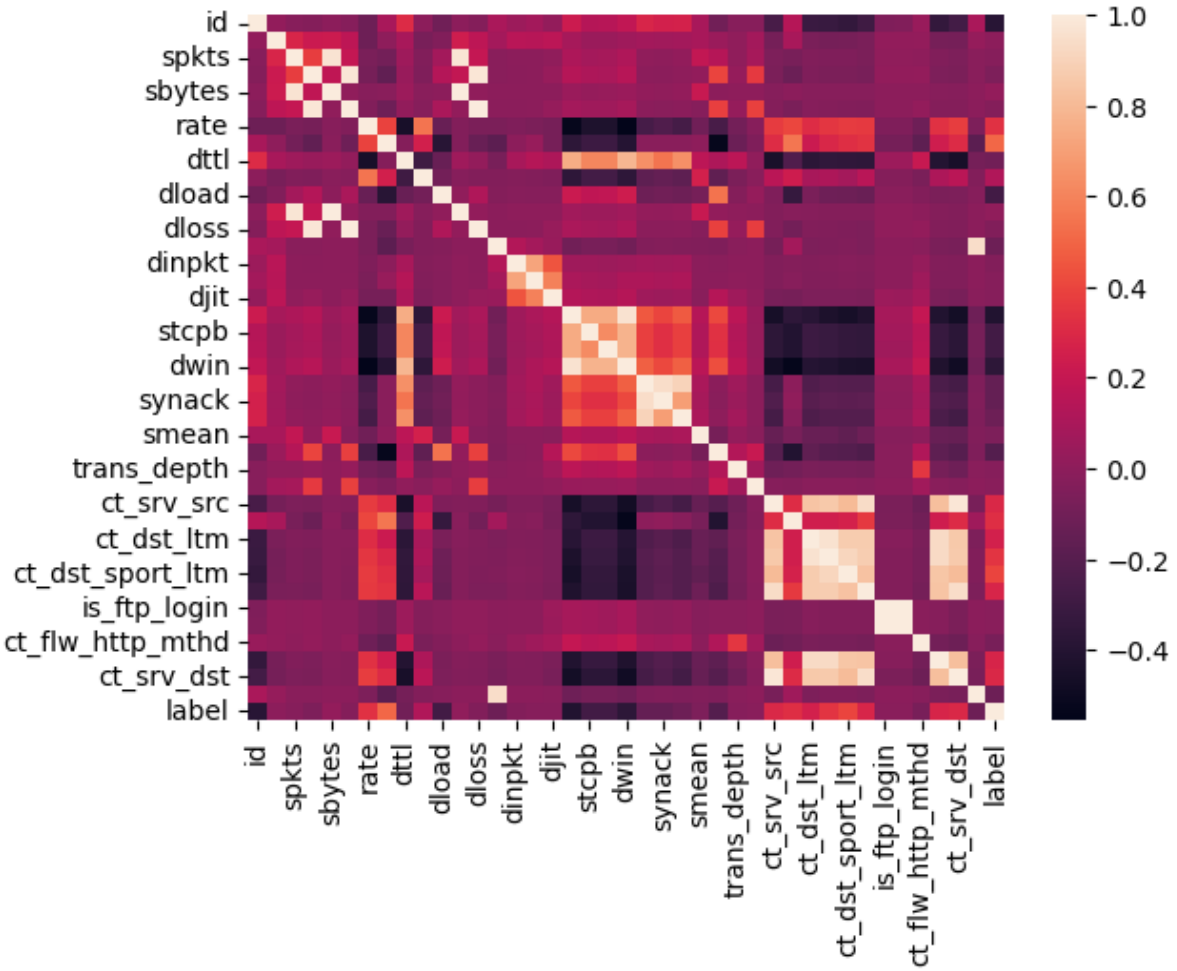

Figure 2: Correlated Features

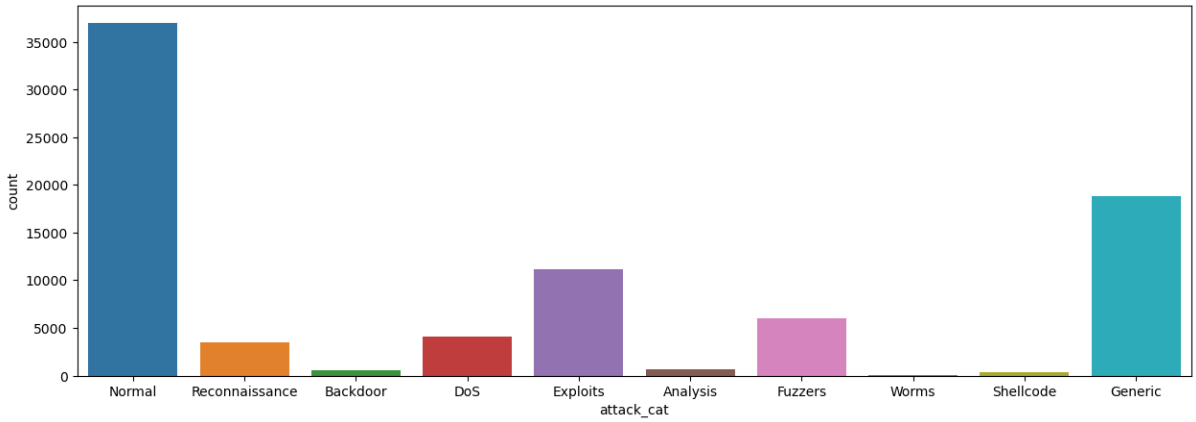

Figure 3: Countplot of the Imbalanced Data

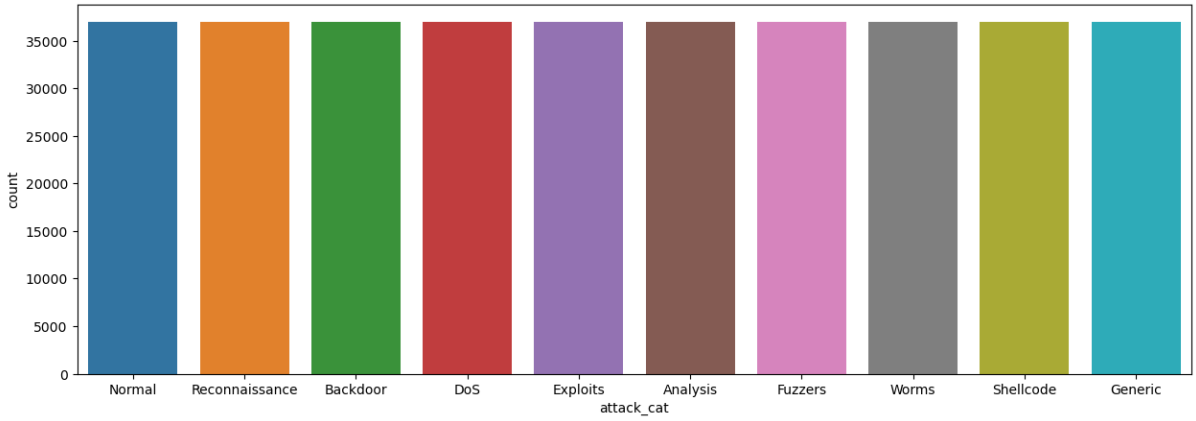

Figure 4: Countplot of the Balanced Data

Table 1: Most Ten Important Features

	Feature	Important_Features
0	id	0.235239
1	sbytes	0.098822
2	sload	0.053006
3	ct_dst_sport_ltm	0.050069
4	smean	0.045883
5	service	0.044786
6	ct_srv_dst	0.034816
7	ct_dst_src_ltm	0.030366
8	dpkts	0.026507
9	ct_src_dpoutItm	0.026507

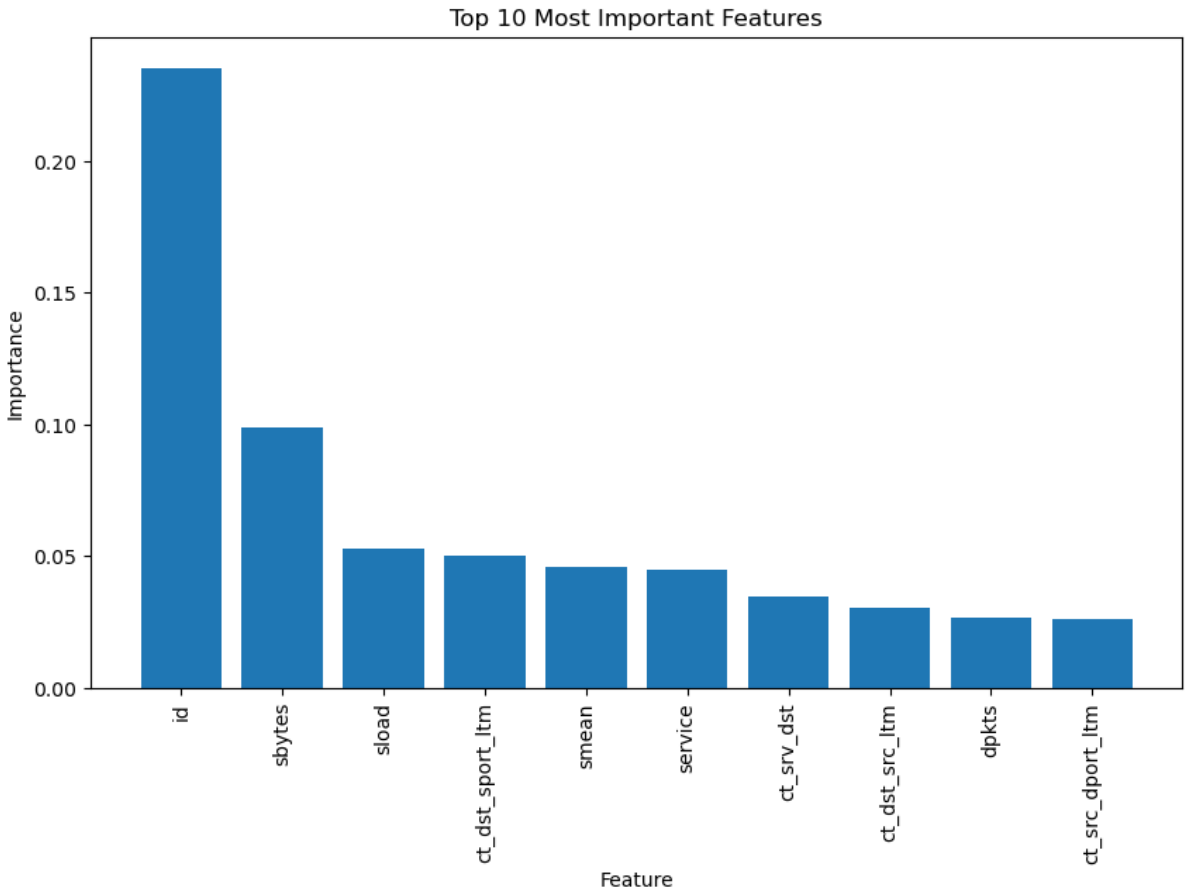

Figure 5: Top ten important features

4.2 Model Training With Random Forest Classifier

In other to have an efficient model for the purpose of identifying potentially harmful network activity system, we trained the random forest model using just the ten most important features. The random forest model was evaluated in terms of accuracy, precision, and recall. The results of the random forest model can be seen in Figure, 6, 7, and 8. The results shows that the proposed model shows better performance in detecting malicious traffics on network systems.

	precision	recall	f1-score	support
Normal	1.00	1.00	1.00	677
Backdoor	1.00	1.00	1.00	583
Analysis	1.00	1.00	1.00	4089
Fuzzers	1.00	1.00	1.00	11132
Shellcode	1.00	1.00	1.00	6062
Reconnaissance	1.00	1.00	1.00	18871
Exploits	1.00	1.00	1.00	37000
DoS	1.00	1.00	1.00	3496
Worms	1.00	1.00	1.00	378
Generic	1.00	1.00	1.00	44
accuracy			1.00	82332
macro avg	1.00	1.00	1.00	82332
weighted avg	1.00	1.00	1.00	82332

Figure 6: Classification Report

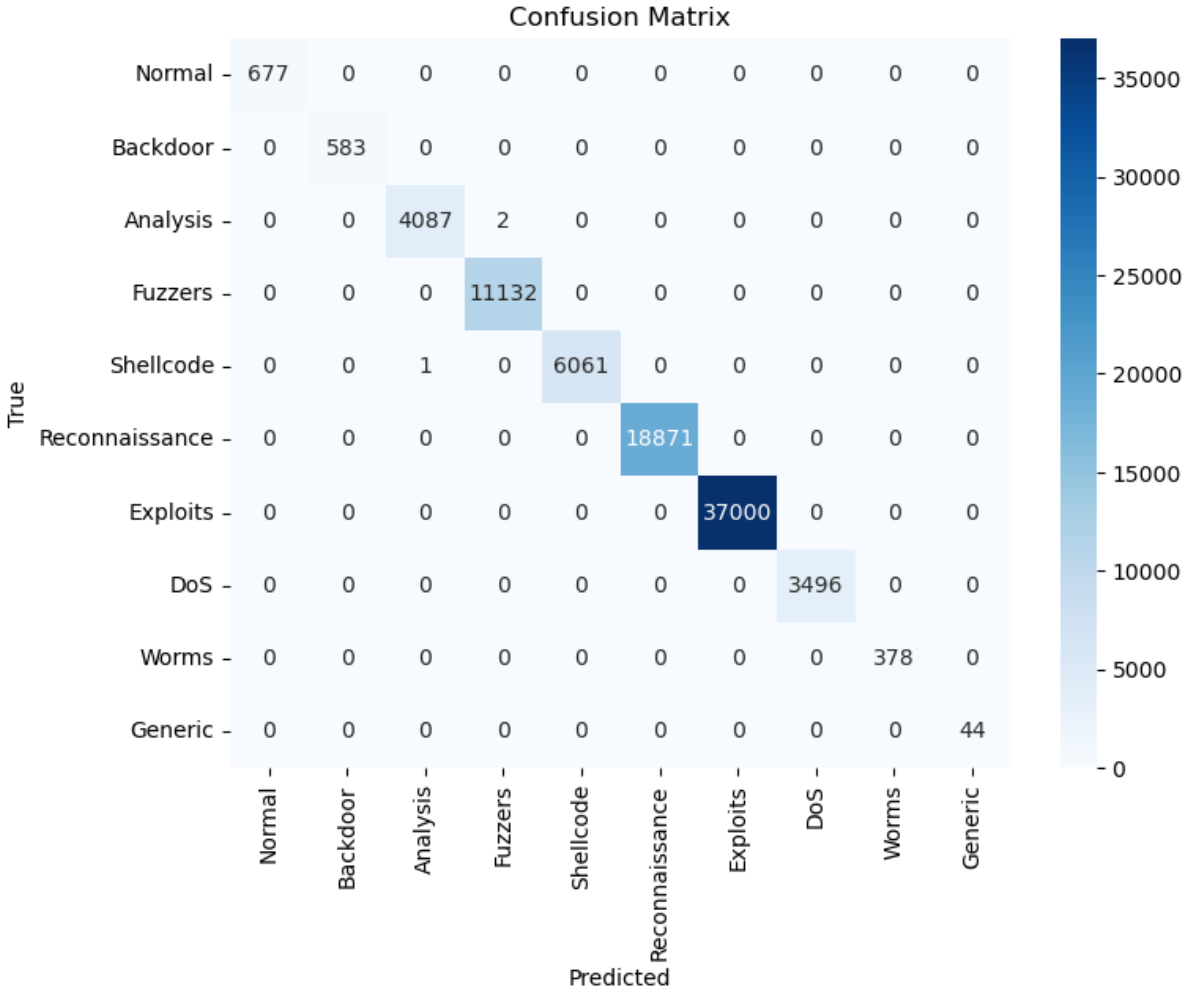

Figure 7: Confusion Matrix

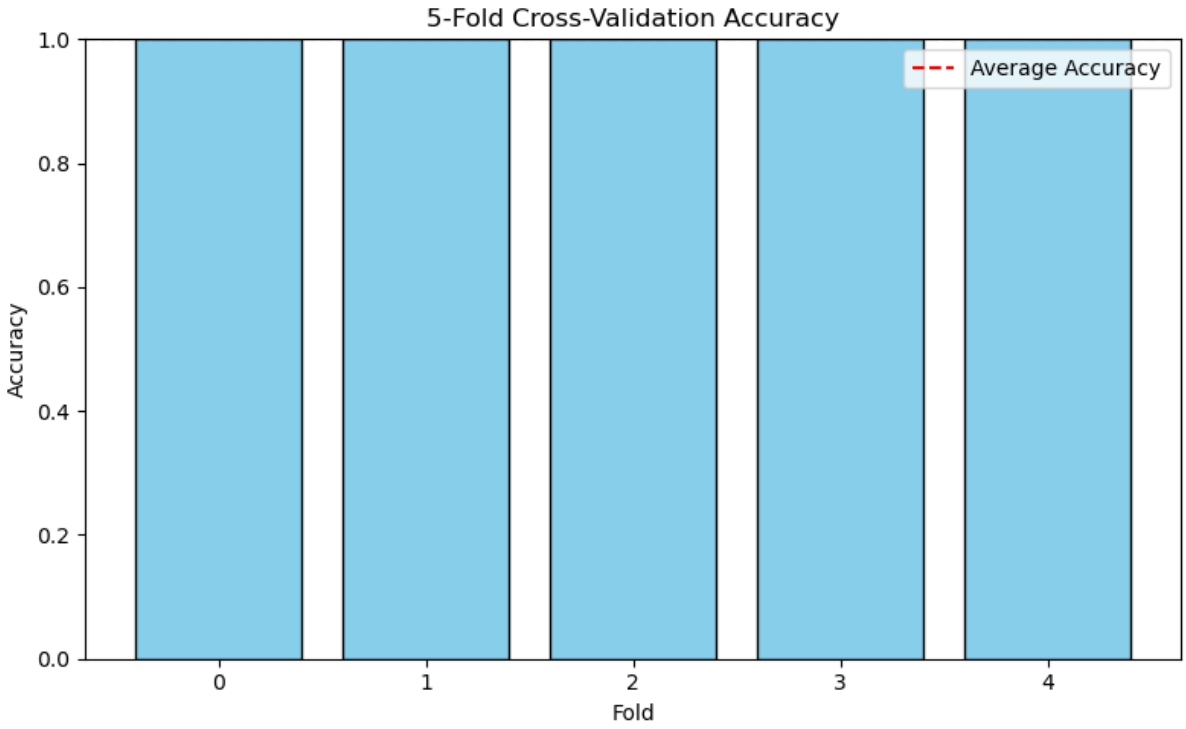

Figure 8: Five Fold Training Validation

Table 2: Comparison With Other Existing System

Authors	Title	Accuracy (%)
[16]	"Malicious traffic detection in iot and local networks using stacked ensemble classifier"	98.5
[17]	"Apply machine learning techniques to detect malicious network traffic in cloud computing"	94
Our Method		99.9

5. Discussion of Results

The correlation matrix of associated features is depicted in Figure 2, as observed in the conducted experiment. The correlation matrix is employed to assess the presence of a relationship between attributes within a given dataset. The correlation matrix reveals that some features within the dataset exhibit correlation. Figure 3 displays the countplot representing the imbalanced data. According to the data presented in Figure 3, it is evident that the normal class exhibits the greatest bar, while the generic class follows closely behind. Failure to address this

issue may result in the occurrence of overfitting during the model training process. Figure 4 depicts the countplot of the balanced dataset, whereby each class is represented by an equal number of observations. Table 1 presents the initial 10 columns of significance within the dataset, as determined by the random forest model's ranking for the purpose of selecting the most crucial attributes. The table illustrates the characteristics that exert a greater impact on the dataset. Figure 5 presents a visual depiction of the key features. It is evident from Figure 5 that the id feature holds the highest influence on the dataset, followed by the sbytes feature. Figure 6 presents the classification report of the suggested model in relation to the identification of malicious traffic within a network system. The precision score, F1-score, recall score, and accuracy of each class in the data may be observed from the classification report. These scores indicate a high level of performance, with a value of 99.99% that can be estimated as 100%. Figure 7 depicts the confusion matrix, which serves the purpose of illustrating the frequency of accurate predictions made by the model in identifying malicious network traffic within the system. The confusion matrix reveals that the suggested model exhibits minimum occurrences of false positive and false negative values. What is the prevalence of false positive and false negative rates at 0.001%? Figure 8 illustrates the mean accuracy of the model obtained during a five-fold cross-validation procedure. The five-fold cross-validation method was employed to evaluate the model's performance across five distinct iterations. It is evident that the model consistently attained a 99.99% accuracy rate at each training stage. The average accuracy is 99.99%.

6. Conclusion

This paper presents a model for the detection of malicious traffic on a network system. For detecting multi-class of malicious traffics on the network system, we utilized a dataset that comprise 9 types of malicious attacks on a network system. In other to have an efficient model, we conduct exploratory data analysis on the dataset. The exploratory data analysis was used in checking missing values, correlated features, data imbalance, and also important features. The results of the exploratory data analysis show that the dataset is imbalanced, which will lead to overfitting if not resolved. We resolved the data imbalance by performing random oversampling by utilizing the RandomOverSampling technique in python. After resolving the data imbalance, we used a random forest classifier for the extraction of important features of the data. Here, we extracted ten features based on the ranking of the random forest model. The extracted features were used in training the proposed model for the detection of malicious traffic on a network system. The results of the model show a better accuracy for the detection of malicious traffic on the network system with an accuracy of 99.99%, 99.99% for precision, recall and F1-score.

REFERENCES

[1]. Gao, M., Ma, L., Liu, H., Zhang, Z., Ning, Z., & Xu, J. (2020). Malicious network traffic detection based on deep neural networks and association analysis. *Sensors*, *20*(5), 1452.

[2]. Zheng, J., Zeng, Z., & Feng, T. (2022). GCN-ETA: high-efficiency encrypted malicious traffic detection. *Security and Communication Networks*, *2022*, 1-11.

[3]. Xin, L., Ziang, L., Yingli, Z., Wenqiang, Z., Dong, L., & Qingguo, Z. (2022). TCN enhanced novel malicious traffic detection for IoT devices. *Connection Science*, *34*(1), 1322-1341.

[4]. Feng, J., Shen, L., Chen, Z., Wang, Y., & Li, H. (2020). A two-layer deep learning method for android malware detection using network traffic. *IEEE Access*, *8*, 125786-125796.

[5]. Wang, W., Zhu, M., Zeng, X., Ye, X., & Sheng, Y. (2017, January). Malware traffic classification using convolutional neural network for representation learning. In *2017 International conference on information networking (ICOIN)* (pp. 712-717). IEEE.

[6] Ge, M., Fu, X., Syed, N., Baig, Z., Teo, G., & Robles-Kelly, A. (2019, December). Deep learning-based intrusion detection for IoT networks. In *2019 IEEE 24th pacific rim international symposium on dependable computing (PRDC)* (pp. 256-25609). IEEE.

[7] Shafiq, M., Tian, Z., Bashir, A. K., Du, X., & Guizani, M. (2020). CorrAUC: A malicious bot-IoT traffic detection method in IoT network using machine-learning techniques. *IEEE Internet of Things Journal*, *8*(5), 3242-3254.

[8] Bendiab, G., Shiaeles, S., Alruban, A., & Kolokotronis, N. (2020, June). IoT malware network traffic classification using visual representation and deep learning. In *2020 6th IEEE Conference on Network Softwarization (NetSoft)* (pp. 444-449). IEEE.

[9] De Lucia, M. J., & Cotton, C. (2019, November). Detection of encrypted malicious network traffic using machine learning. In *MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM)* (pp. 1-6). IEEE.

[10] Feng, J., Shen, L., Chen, Z., Wang, Y., & Li, H. (2020). A two-layer deep learning method for android malware detection using network traffic. *IEEE Access*, *8*, 125786-125796.

[11] Mitsuhashi, R., Satoh, A., Jin, Y., Iida, K., Shinagawa, T., & Takai, Y. (2021). Identifying malicious dns tunnel tools from doh traffic using hierarchical machine learning classification. In *Information Security: 24th International Conference, ISC 2021, Virtual Event, November 10–12, 2021, Proceedings 24* (pp. 238-256). Springer International Publishing.

[12] Rose, J. R., Swann, M., Bendiab, G., Shiaeles, S., & Kolokotronis, N. (2021, June). Intrusion detection using network traffic profiling and machine learning for IoT. In *2021 IEEE 7th International Conference on Network Softwarization (NetSoft)* (pp. 409-415). IEEE.

[13] Rajesh, L., & Satyanarayana, P. (2021). Evaluation of machine learning algorithms for detection of malicious traffic in scada network. *Journal of Electrical Engineering & Technology*, 1-16.

[14] Hwang, R. H., Peng, M. C., Huang, C. W., Lin, P. C., & Nguyen, V. L. (2020). An unsupervised deep learning model for early network traffic anomaly detection. *IEEE Access*, *8*, 30387-30399.

[15] Abdulhammed, R., Faezipour, M., Abuzneid, A., & AbuMallouh, A. (2018). Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic. *IEEE sensors letters*, *3*(1), 1-4.

[16] Indrasiri, P. L., Lee, E., Rupapara, V., Rustam, F., & Ashraf, I. (2022). Malicious traffic detection in iot and local networks using stacked ensemble classifier. *Computers, Materials and Continua*, *71*(1), 489-515.

[17]. Alshammari, A., & Aldribi, A. (2021). Apply machine learning techniques to detect malicious network traffic in cloud computing. *Journal of Big Data*, *8*(1), 1-24.