Prediction Models to Effectively Detect Malware Patterns in the IoT Systems

Rawabi Nazal Alhamad[1]
Department of Computer Science and Technology
Jouf University, Al-Jouf, Saudi Arabia

Faeiz M. Alserhani[2]
Department of Computer Engineering and Networks
Jouf University, Al-Jouf, Saudi Arabia

***Abstract*—The Widespread use of the Internet of Things (IoT) has influenced many domains including smart cities, cameras, wearables, smart industrial equipment, and other aspects of our daily lives. On the other hand, the IoT environment deals with a massive volume of data that needs to be kept secure from tampering or theft. Detection of security attacks against IoT context requires intelligent techniques rather than relying on signature matching. Machine learning (ML) and Deep Learning (DL) approaches are efficient to detect these attacks and predicting intrusion behavior based on unknown patterns. This study proposes the application of five deep and ML techniques for identifying malware in network traffic based on the IoT-23 dataset. Random Forest, Catboost, XGBoost, Convolutional Neural Network, and Long Short-Term Memory (LSTM) models are among the classifiers utilized. These algorithms have been selected to provide lightweight security systems to be deployed in the IoT devices rather than a centralized approach. The dataset was preprocessed to remove unnecessary or missing data, and then the most significant features were extracted using a feature engineering technique. The highest overall accuracy achieved was 96% by applying all classifiers except LSTM which recorded a lower accuracy.**

Keywords—Internet of Things (IoT); malware deletion; random forest; Catboost; convolutional neural network; long short-term memory (LSTM); XGBoost

I. Introduction

The Internet of Things (IoT) refers to the billions of connected physical devices through the Internet, for global storage and data exchange [1]. Recently, the IoT has been involved in a variety of fields in our daily life, including smart cities, cameras, wearables, smart industrial equipment, household appliances, medical devices, and even nuclear reactors [2]. The infrastructure of the IoT devices has limited storage, hardware, and battery life, making the sophisticated or standard security algorithms difficult to be applied in such a domain with limited resources it has. Furthermore, the IoT environment deals with a large amount of data, making it subject to botnets, firmware hijacking, distributed denial of service attacks, eavesdropping, the man in the middle, and other attacks. The network security of IoT devices is considered more critical compared to network security due to the large number of attacks, its small size, and multiple vulnerabilities of IoT tools [3]. By 2025, it is expected that 41,600 million IoT devices will have been shipped around the world. According to the 2022 SonicWall Cyber Threat Report [4], malware decreased somewhat in 2021, indicating a third consecutive year of decline and a seven-year low. An increase in attacks during the second half of 2021 nearly wiped out the 22 percent decline in malware that researchers observed at the midpoint of the year, reducing the total decrease for 2021 to just 4 percent, where 2022 Global Cyberattack was registered approximately 60,1 million IoT Malware attacks, whose volume increased by 6 percent in 2021. This growth indicates a plateau compared to the previous two years, during which these attacks increased by 218 percent and 66 percent, respectively.

The research question behind this work is “Is it possible to enhance the accuracy of malware and benign detection in the IoT environment based on deep and ML techniques using a standard dataset that holds network traffic information? And what is the potential of deploying these techniques in end-systems?” Hence, to solve this question, five different deep and machine learning approaches are suggested to be applied to the IoT-23 dataset. The IoT-23 dataset has 14 labels for 20 malicious and 3 benign captures and many features; therefore, to reduce the computational cost of using the IoT-23 dataset, only the malware captures were explored, and only ten labels were analyzed.

The main purpose of this study is to analyze traffic traces and network behavior of IoT devices by using the 23-IoT dataset, which consists of attributes from network traffic based on different protocols to identify malware. After that, using preprocessing stage and feature engineering to extract the most significant features and thereby reduce the dataset's dimensionality. Then applies classification techniques that include Random Forest, Catboost, CNN, LSTM, and XGBoost algorithms, to detect malware and benign traffic, which helps in developing a robust intrusion system capable of detecting different types of attacks. Finally, to accomplish the evaluation and better prediction, the classifier models are subjected to cross-validation, optimization, and comparison.

The following is the structure of this research study: Section 2 depicts the related works by applying deep and machine learning techniques to detect malware activities based on the IoT-23 dataset. Section 3 outlines the suggested methodology in this research, while Section 4 discusses the various machine learning and deep learning models implemented. Moreover, Section 5 discusses the results of each model. Finally, the conclusion and future work.

II. Related Work

The evolution of the sophisticated IoT environment has necessitated the development of malware and benign system detection based on the recent deep and ML techniques, as detailed in the previous section. Many researchers have used various deep and ML models to achieve network traffic analysis to address this issue. For example, in [5], the authors implemented deep and ML algorithms, namely, RF, Naive Bayes (NB), Multi-Layer Perceptron (MLP), Support Vector Machine (SVM), and AdaBoost (ADA). The dataset used in this work was the IoT-23 dataset and the best accuracy of detection was achieved by using the RF model with a value of 99.5%.

In [6], the researchers applied four algorithms to the IoT-32 dataset, including CNN, Decision Trees, Naive Bayes, and SVM. All IoT-23 dataset's labels were analyzed in this work the highest accuracy achieved was 73% by using the decision tree model, CNN achieved 69.35%, and SVM attained 69% of accuracy. Based on the same dataset, another author was analyzing all labels of the dataset by using Decision Trees, RF, Naive Bayes, SVM, AdaBoost, XGBoost, CNN, and Multi-Layer Perceptron models [7]. The highest accuracy achieved was 74% by-using Decision Trees, RF, and MLP models.

On the other hand, the authors in [8] achieved a better detection rate near 100% of the f1-score metric; however, only four malware types (Hide and Seek, Torii, Mirai, and Trojan) were involved and Okiru malware was not analyzed. Support Vector Machine (SVM), Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), Isolation Forest (iForest), Local Outlier Factor (LOF), and Deep Reinforcement Learning (DRL) classifiers were used to do binary classification and multiclassification.

On another trend, a DL ensemble for network malware detection was proposed based on IoT-23, LITNET-2020, and NetML-2020 datasets [9]. Deep Neural Network (DNN) and Long Short-Term Memory (LSTM), and a meta-classifier (i.e., logistic regression) were applied. The method employed a two-step process for the detection of network anomalies to improve the capabilities of the suggested methodology. For the feature engineering challenge in the first stage, data pre-processing, a Deep Sparse AutoEncoder (DSAE) was used. For classification in the second phase, a stacking ensemble learning strategy was applied. The proposed method was evaluated on IoT-23, LITNET-2020, and NetML-2020 datasets, and only Benign, Mirai, Attack, PartOfAHorizontalPortScan, and C&C malware types are classified in this work. The classification type that was applied by the authors is binary classification to recognize normal from abnormal attacks and the accuracy achieved is around 100%.

In [3], the authors have built and performed IoT anomaly detection systems based on actual IoT-23 large data for identifying attacks based on artificial NN like Convolutional NN (CNN), Recurrent Neural Networks (RNN), and Multilayer Perceptron (MLP). As a consequence, Convolutional Neural Networks outperform Multilayer Perceptron and Recurrent Neural Networks in IoT anomaly detection, with a metric accuracy score of 0.998234 and a minimal loss function of 0.008842. The Mirai, DoS, NScan, Normal, and MITM_ARP attacks are classified in this work.

An anomaly-based intrusion detection model to detect malware in IoT network traffic was created and implemented on IoT-23, IoT Network Intrusion, BoT-IoT, and MQTT-IoT-IDS2020 datasets [10]. A multiclass classification model was created using CNN (Convolutional Neural Network) models and then uses to accomplish binary and multiclass classification via the transfer learning principle. Three model architectures were suggested by the authors, namely, 1D, 2D, and 3D CNN models. The utilized multiclass classifier not only can classify 15 different types of attacks but also efficiently differentiate them from normal network data [11]. The authors included seven attacks and one normal from the IoT-32 dataset: Normal, Attack, C&C, FileDownload, HeartBeat, Okiru, Port Scan, and Torii attacks. The proposed models achieved high values in the used performance metrics, where all models reached more than 99.89% for accuracy, precision, recall, and f1-score metrics.

The authors in [12] used Bidirectional Generative Adversarial Networks (BiGAN) and Adversarial Autoencoders (AAE) to detect malware in network traffic based on the full IoT-23 dataset version. The proposed models outperformed traditional ML such as RF, by getting an f1-score of 99. However, not all labels were analyzed in this work, only nine malware types were classified. Sahu, Amiya Kumar, et al. [13] proposed a DL-based classifier for detecting IoT attacks. The CNN model is used to learn the IoT features, and then an LSTM-based classifier is used to classify them. The model was applied to eight labels on the IoT-23 dataset, which included Command and Control (C&C), Distributed Denial of Server (DDoS), File Download, Heart Beat, Part of A Horizontal Port Scan, Mirai, Torii, and Okiru. A 96% accuracy rate in detecting malicious devices was achieved but the Benign label was ignored. Dartel, Bram. The author in [14] suggested a ML technique comprising Decision Tree, RF, Support Vector, and an ESP32. The sub-labels of the IoT-23 dataset are judged irrelevant because their methodology is focused on malware detection rather than malware classification. The result was running on an IoT device that really works as an IoT device, so it was easy to run the device next to the malware detection algorithm.

Using a similar size dataset, they divided it into two separate datasets. In addition, the data were randomly dispersed throughout them in order to limit the number of multiclass labels, hence the metrics showed a significant disparity in class size. Following this, they used the following ML techniques: RF, Naive Bayes, Support Vector Machine, and Decision Tree. 99.5 percent of the accuracy of the RF algorithm produced the best results [15]. They suggested using ML techniques like Support Vector Machine, Decision Tree, and RF to classify malware attacks like DDoS attacks, and also, a Principal.

Component Analysis (PCA) was used to reduce the number of dimensions. The results of PCA were measured against what would have happened if PCA hadn't been used. When PCA was used, the algorithm ran much faster with fewer features than it did when PCA wasn't used. When it comes to classifying attacks, Decision Tree and RF are better than SVM [16].

TABLE I. COMPARISON OF RELATED WORK

Author & year	Study Name	Method	Accuracy	Features selected	Notes
Ullah, Imtiaz Mahmoud, Qusay H[11] 2021	Design and development of a DL-based model for anomaly detection in IoT networks	1D, 2D, and 3D CNN	99.89%	Normal, Attack, C&C, FileDownload, HeartBeat, Okiru, Port Scan, and Torii attacks	They get high accuracy with three models by CNN but for 7 attacks classes with normal class
Abdalgawad, N Sajun, [21] 2021	Generative DL to detect Cyberattacks for the IoT-23 Dataset	(BiGAN), (AAE), RF	getting an f1-score of 99.	9 attacks	not all labels were analyzed
Sahu, Amiya Kumar Sharma, [13] 2021	IoT attack detection using hybrid DL Model	CNN, LSTM	96%	eight labels on the IoT-23 dataset, which included Command and Control (C&C), Distributed Denial of Server (DDoS), File Download, Heart Beat, Part Of A Horizontal Port Scan, Mirai, Torii, and Okiru	8 attacks and Benign label was ignored
Dr. R. Thamaraiselvi and S. Anith [14] 2021	Malware detection in IoT devices using ML	DT, RF, SVM, and an ESP32	Training model on Iot-23. Testing was running on an IoT device that works as an IoT device	Sub Labels of IoT-23	
R. Thamaraiselvi [15]- 2020	Attack and Anomaly Detection in IoT Networks using ML	RF, Naive Bayes, Support Vector Machine, and Decision Tree	99.5	Split the dataset into two parts	
D. Nanthiya [16] 2021	SVM Based DDoS An IoT Using Iot-23 Botnet Dataset	SVM, DT , RF	SVM is higher	Principal Component Analysis (PCA) was used to reduce the number of dimensions	

Authors in [17] proposed a method for Malware Detection in Fog Layer. This method has three important phases for pre-processing: feature extraction, feature selection to reduce the number of features, and classification. Convert the binary files to hexadecimal using HexDump, segmentation into 4-gram, features selection and reduction using Gain Ratio; decision tree and Component Analysis (PCA). Finally, apply decision tree classifier. The accuracy was Acc. 96.7. The authors [18] have presented a new method based on ML for detecting Mirai malware "NBaIoT" dataset, which data consist of features infected by the Mirai Malware, is used in that study. The Cross-Validation technique has been used for data splitting to overcome overfitting, and the experiment was conducted using ANN. The achieved accuracy is 92.8%. The Opcode dataset has been used in this research; it consists of 70,140 normal and 69,860 malicious malware. The IoT Device dataset's benign or malignant input is classified using a deep neural network (DNN). The obtained accuracy reached 99.7 % [19]. Table I summarize the related works and provide a comparison between different approaches.

III. METHODOLOGY

Generally, the research methodology has been designed for detecting about 10 IoT malware types which will be explained in detail in this section. To detect IoT malware attacks, five ML and DL algorithms are implemented based on the IoT-23 dataset, namely, RF, Catboost, Convolutional Neural Network, Long Short-Term Memory (LSTM), and XGBoost. The proposed methodology has five stages to be accomplished to evaluate the algorithms: data collection, pre-processing, feature selection, training and testing, and classification stages as shown in Fig. 1.

A. Data Collection

To use machine learning techniques, a dataset with a large number of samples that have been contextualized and labeled correctly is necessary. This section gives a quick overview of the chosen dataset in this work. The IoT-23 dataset is chosen since it provides a large dataset with twenty-three captures of various IoT network traffic that include three benign and twenty malware traffic captures [20]. It is a modern dataset that consists of more than one million network traffic of IoT devices and was first published by the Stratosphere Laboratory in January 2020, with captures spanning the years 2018 to 2019. The main goal of creating this dataset is to create a large dataset under real circumstances for researchers that h valid malware and benign traffic labels in order to apply machine learning algorithms to enhance intrusion detection in IoT environments.

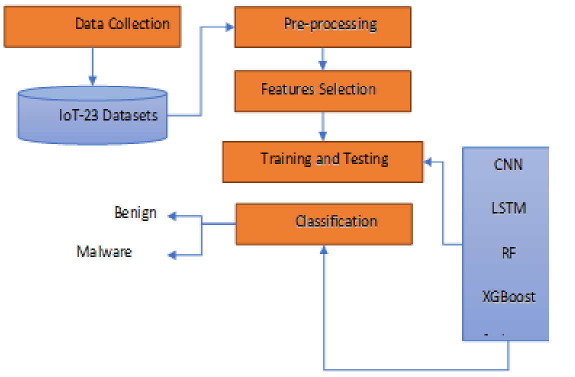


Fig. 1. The Proposed Methodology in this Work.

This dataset has 14 labels for 20 malicious and 3 benign captures that include Part-Of-A-Horizontal-PortScan, DDoS, Attack, FileDownload, Okiru, Benign, C&C-HeartBeat-FileDownload, C&C, C&C-FileDownload, C&C-HeartBeat, C&C-Mirai, C&C-FileDownload, Okiru-Attack, and C&C-Torii. Despite this, the dataset contains 21 feature properties that determine the feature of the connections, one of which is the class label, as shown in Table II. Some of the features are nominal, and some features have time-stamp values, while others are quantitative. In this work, only the malware captures were investigated to reduce the computational cost of using the full version of the IoT-23 dataset, as well as only ten labels were analyzed: Part-Of-A-Horizontal-PortScan (753565 rows), DDoS (138777), C&C-Mirai (1), C&C-HeartBeat (341), C&C (15100), Attack (3915), Benign (195270), FileDownload (13), C&C-FileDownload (43), C&C-Torii (30), and C&C-HeartBeat-FileDownload (8) labels.

TABLE II. THE DESCRIPTION OF THE IMPORTANT FEATURES OF IOT-23 DATASET [9]

Feature No.	Feature Name	Data Type	Feature Description
1	Ts	int	Timestamp of the capture.
2	Uid	str	The capture's Unique ID.
3	id.orig_h	str	Originating IP where the attack happened.
4	id.orig_p	int	Source port used by the responder.
5	id.resp_h	str	The destination IP address of the device on which the capture happened.
6	id.resp_p	int	Destination port used from the response from the device on which the capture happened
7	Proto	str	Transaction or Network protocol
8	Service	str	Application protocols such as DNS, FTP, HTTP, SMTP, SSH, etc.
9	Duration	float	The overall duration of the transmission between device and attacker
10	orig_bytes	int	The transaction bytes from source to destination.
11	resp_bytes	int	The transaction bytes from destination to source.
12	conn_state	str	Represents the current connection state
13	local_orig	bool	The connection is locally initiated.
14	local_resp	bool	The response is locally initiated.
15	missed_bytes	int	The number of missing bytes of a transaction
16	History	str	The connection state's history.
17	orig_pkts	int	The total packets being sent to a device.
18	orig_ip_bytes	int	The total bytes being sent to a device.
19	resp_pkts	int	The total packets being sent from a device.
20	resp_ip_bytes	int	The total bytes being sent from a device.
21	Label	str	Type of capture: Benign or malicious, alongside with Type of the malicious capture

B. Pre-processing Stage

Data pre-processing is the process that transforms raw data into a form that can be read, accessed, and analyzed. The pre-processing stage is of major importance, to ensure or enhance the total performance or accuracy of any system before applying the machine learning algorithms. In this work, the full IoT-23 dataset is used that has around 20GB in size. When we extract this file into the local hard disk, in windows 11, 23 folders are created; however, only 20 folders are utilized that are relevant to malware captures. To read the data from each folder, a 'read_table' function from Pandas is imported to extract all data from 'conn.log.labeled' file, this function is applied in the Jupyter platform that supports python programming language. Therefore, 20 variables are created to read all data from each folder, and then the 'concat' function from Pandas is applied to combine all these variables for creating a Dataframe that can save data to a CSV file using 'to_csv'. But before saving such data to the CSV file, we need to change the labels for some rows of the extracted data. For example, some rows have a '-Malicious PartOfAHorizontalPortSca' label, while others have '(empty) Malicious PartOfAHorizontalPortScan', so all these labels denote one attack and will be changed to the 'PartOfAHorizontalPortScan' label; therefore, we did the same step for all other labels for creating unique labels.

C. Features Selection

The process of selecting the features that have the greatest impact on the prediction outcomes is of major importance to increase the overall accuracy. Therefore, in this section, effective steps will be explained in detail. After the pre-processing stage, all data is stored in a CSV file named 'iot23_combined.csv', and this file is loaded to a Dataframe by using the 'read_csv' Panda's function. After that, the first feature is removed since it represents the number of rows in the IoT-23 datasets. On the other hand, all labels will be analyzed in this work except the 'Okiru' label, because of the computational cost of the large size of the IoT-23 dataset, as well as we found that some researchers have done malware detection by using only a few labels not all of them [8], [9].

Furthermore, in order to delete data that is not related to the 'label' column in this dataset, a correlation matrix is applied to all features, as shown in Fig. 2. The correlation matrices show the strength of correlation by using a scale from dark blue to yellow. The negative correlation is represented by dark-blue, while yellow represents the positive correlation, and green denotes to the weak correlation. From the figure, the yellow color indicates that the features are strongly correlated while the dark blue indicates that the features are weakly correlated. Furthermore, the repeated features should be removed when any exist by applying the 'get_duplicate_features' function from the 'fast_ml' library, and now the data is ready for the training and testing stage.

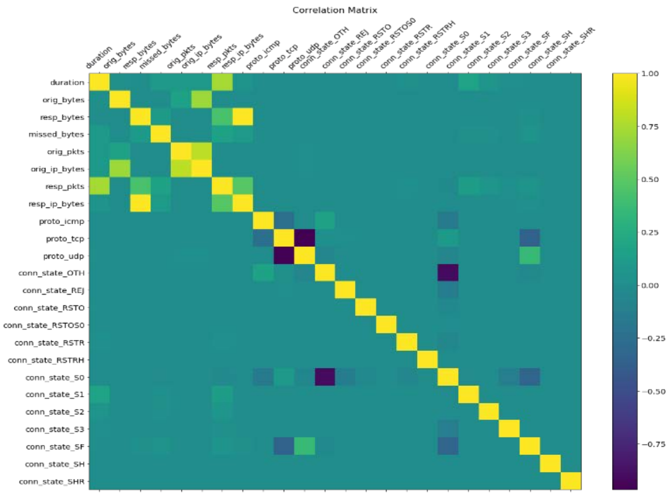


Fig. 2. The Correlation Matrix for All IoT-23 Dataset's Features.

D. Training and Testing

The process of training and testing the proposed machine learning models will be explained in this stage. This process depends on the prepared dataset in the previous stage to train and test five machine learning algorithms (CNN, LSTM, RF, XGBoost, and Catboost) until these models can distinguish the malware from the benign captures. However, there are a few steps that should be taken into consideration such as splitting the data into test and train data by applying the 'train_test_split' function from the Sklearn library, which is a function to split the matrices into random train and test subsets. Sklearn is a free python library, and being created to perform some techniques in the machine learning field such as classification, regression, and grouping. Applying the 'train_test_split' function is very important for an unbiased evaluation of all suggested machine learning models and checking the final accuracy not only in the training data but also in the test data that have not been seen or trained before in such models. In this work, the size of training data is 80%, while the test data is 20% of all the prepared data. After all these steps, all the models are ready to be trained and tested with different parameters depending on the requirements of the model being run at the time of execution.

E. Classification

Classification is a supervised learning perception that is responsible forsplittings data into separate classes, and can be applied in various classification issues such as facial detection, speech recognition, handwriting recognition, and document categorization. In this work, after the proposed models are trained on the IoT-23 dataset, and then a set of data isolated from these models are used to check the accuracy of the trained models to correctly separate all data according to their labels. Accuracy, Precision, F1-Score, and Recall are the measures used to evaluate the algorithms' efficiency, and their definitions are as follows:

- Accuracy: is the percentage of correctly labeled classes in relation to the total number of classes and is given by the next equation:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Precision: how many of the positive class classifications made by the model are correct? and is given by the next equation:

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

- Recall or Sensitivity: how many of the positive class scenarios with expected values are correct? and is given by the following equation:

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

- F1-score: is a harmonic average that combines precision and sensitivity into one measure, and the following equation shows how to calculate this measure:

$$F1 - score = \frac{2*Precision*Recall}{Precision+Recall} \quad (4)$$

Where TP (True-Positive) means the model classifies malware cases as positive (malware cases classified as malware correctly). Whereas FN (False-Negative) represents mistakenly classified malware as negative (malware classified as normal or benign but the truth is that malware has existed). Moreover, TN (True-Negative) denotes correctly current cases classified as negative or no malware that has existed in the current cases. Finally, FP (False-Positive) means mistakenly classified malware as positive (malware classified as malware but the truth is that no malware exists).

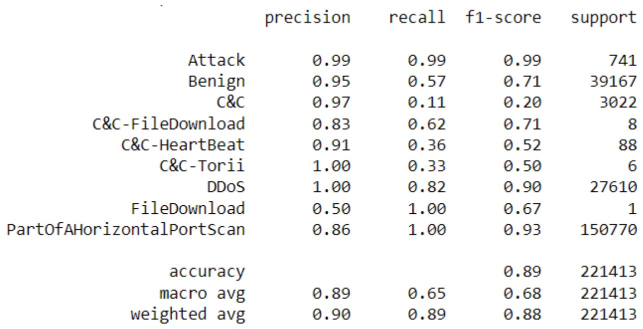
IV. Experimental Results

In this section, various experiments were carried out intending to check the accuracy of detection of the proposed models in this work. To effectively identify the IoT malware and benign captures, Random Forest (RF), CatBoost, XGBoost, LSTM, and CNN were implemented. The measure performances used in this work include confusion matrix, classification report (precision, recall, f1-score, accuracy), and ROC curves, for comparing the different performance of the proposed models.

A. Random Forest Classifier's Experiments

As mentioned in the implementation section, a random forest classifier is used because of its accuracy is higher, due to this classifier takes the final prediction based on the forecast from each tree, not from a single decision tree. The random forest achieved the highest detection accuracy for differentiating malware and benign captures with a value of 89% as portrayed in Fig. 3. Besides, in this figure, three measure metrics ((precision, recall, f1-score) are appeared for each label, while the overall accuracy is presented to show the total detection accuracy for all labels included in this work. The highest precision attained is for C&C-Torii and DDoS with a value of 100% for both of them, the highest recall was 100% for FileDownload and PartOfAHorizontalPortScan, and the highest f1-score was 99% and 93% for Attack and PartOfAHorizontalPortScan, respectively. The f1-score is more important than precision and recall because it can assist balancing the metric between positive and negative samples. Additionally, two malwares (C&C-HeartBeat-FileDownload, C&C-Mirai) have not appeared in the classification report because the small number of samples they have: 1 for C&C-

Mirai and 8 samples for C&C-HeartBeat-FileDownload. Furthermore, this the figure has micro-averaging and macro-averaging curves, where the micro-averaging is used to score each prediction equally and the macro- averaging is used to examine the overall performance of the classifier based on the most common class labels.



```
                          precision    recall  f1-score   support

                  Attack       0.99      0.99      0.99       741
                  Benign       0.95      0.57      0.71     39167
                     C&C       0.97      0.11      0.20      3022
         C&C-FileDownload       0.83      0.62      0.71         8
            C&C-HeartBeat       0.91      0.36      0.52        88
               C&C-Torii       1.00      0.33      0.50         6
                    DDoS       1.00      0.82      0.90     27610
             FileDownload       0.50      1.00      0.67         1
PartOfAHorizontalPortScan       0.86      1.00      0.93    150770

                accuracy                           0.89    221413
               macro avg       0.89      0.65      0.68    221413
            weighted avg       0.90      0.89      0.88    221413
```

Fig. 3. The Classification Report of Random Forest Model.

To draw a ROC (receiver operating characteristic curve) curve for these labels, we need to encoded them to integers ranging from 0 to 8: Attack=0, Benign=1, C&C=2, C&C-FileDownload=3, C&C-HeartBeat=4, C&C-Torii=5, DDoS=6, FileDownload=7, and labels PartOfAHorizontalPortScan=8. These labels were numbers according to their appearance in the classification report in Fig. 3. The ROC curve shows the performance of the classification of the model, as shown in Fig. 4. Additionally, as we see this graph has classes that range from 0 to 8, where class 0 = Attack, class 1= Benign, and so on. The best ROC curves area values were 100% for the Attack and FileDownload.

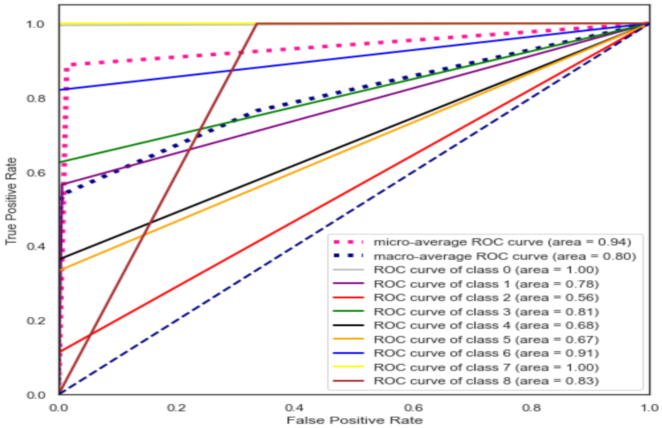
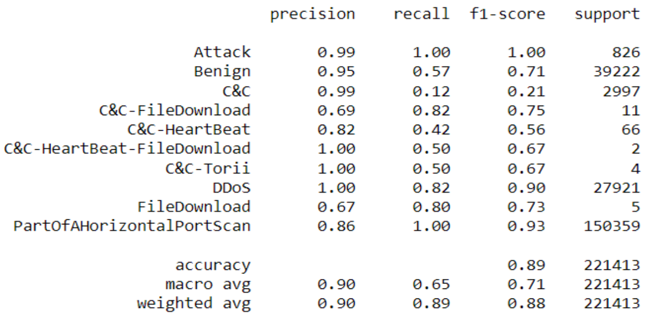


Fig. 4. The ROC Curve of Random Forest Model.

B. XGBoost Classifier's Experiments

The XGBoost classifier achieved the same total accuracy as the random forest with a value of 89% as shown in figure 5. The highest precision attained is for C&C-Torii, C&C-HeartBeat-FileDownload, and DDoS with a value of 100% for all of them, the highest recall was 100% for Attack and PartOfAHorizontalPortScan, and the highest f1-score was 100% and 93%, for Attack and The ROC curve of Random Forest model, PartOfAHorizontalPortScan, respectively.

We notice also that 'C&C-HeartBeat-FileDownload' is presented in this model, so the encoding process is changed starting from 0 to 9 according to the order that these labels appear in the classification report as shown in Fig. 5, Attack=0, C&C-HeartBeat-FileDownload=5, so on.



```
                           precision    recall  f1-score   support

                    Attack       0.99      1.00      1.00       826
                    Benign       0.95      0.57      0.71     39222
                       C&C       0.99      0.12      0.21      2997
          C&C-FileDownload       0.69      0.82      0.75        11
             C&C-HeartBeat       0.82      0.42      0.56        66
C&C-HeartBeat-FileDownload       1.00      0.50      0.67         2
                 C&C-Torii       1.00      0.50      0.67         4
                      DDoS       1.00      0.82      0.90     27921
              FileDownload       0.67      0.80      0.73         5
 PartOfAHorizontalPortScan       0.86      1.00      0.93    150359

                  accuracy                           0.89    221413
                 macro avg       0.90      0.65      0.71    221413
              weighted avg       0.90      0.89      0.88    221413
```

Fig. 5. The Classification Report of XGBoost Model.

The highest ROC curves area values were 100% for the Attack and 91% for both C&C-FileDownload and DDos labels as shown in Fig. 6.

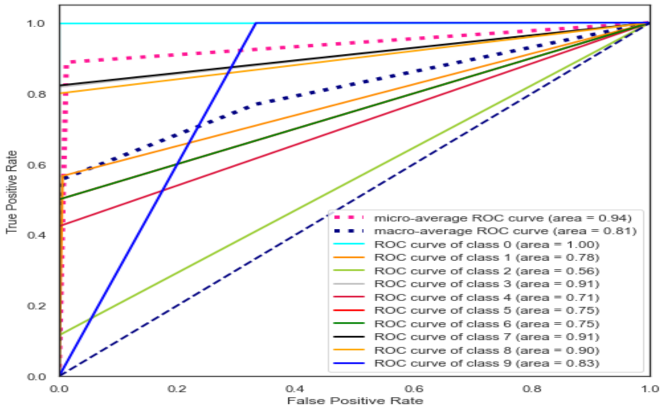
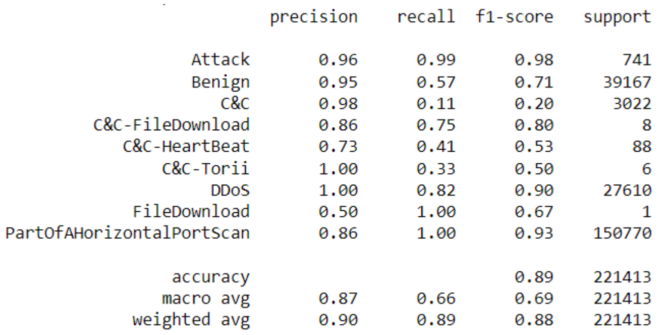


Fig. 6. The ROC Curve of XGBoost Model.

C. CatBoost Classifier's Experiments

CatBoost classifier reached the same overall accuracy as the Random Forest and XGBoost with a value of 89% as depicted in Fig. 7, but it takes more time than both of them. The 'C&C-HeartBeat-FileDownload' label does not appear in the classification report of this model since the very small number it has. The highest precision got is 100% for both C&C-Torii and DDoS, and also 96% for the Attack label. Furthermore, the highest recall was 100% for FileDownload and PartOfAHorizontalPortScan, 99% for the Attack, while the highest f1-score was 98%, 93%, 90%, for Attack, PartOfAHorizontalPortScan, and DDos labels, respectively.

On the other hand, the ROC curve was drawn for this model, where the highest area values were 100% for both Attack and FileDownload and 91% for the DDos label as shown in Fig. 8.



```
                          precision    recall  f1-score   support

                   Attack       0.96      0.99      0.98       741
                   Benign       0.95      0.57      0.71     39167
                      C&C       0.98      0.11      0.20      3022
          C&C-FileDownload       0.86      0.75      0.80         8
            C&C-HeartBeat       0.73      0.41      0.53        88
                 C&C-Torii       1.00      0.33      0.50         6
                     DDoS       1.00      0.82      0.90     27610
             FileDownload       0.50      1.00      0.67         1
  PartOfAHorizontalPortScan       0.86      1.00      0.93    150770

                 accuracy                           0.89    221413
                macro avg       0.87      0.66      0.69    221413
             weighted avg       0.90      0.89      0.88    221413
```

Fig. 7. The Classification Report of CatBoost Model.

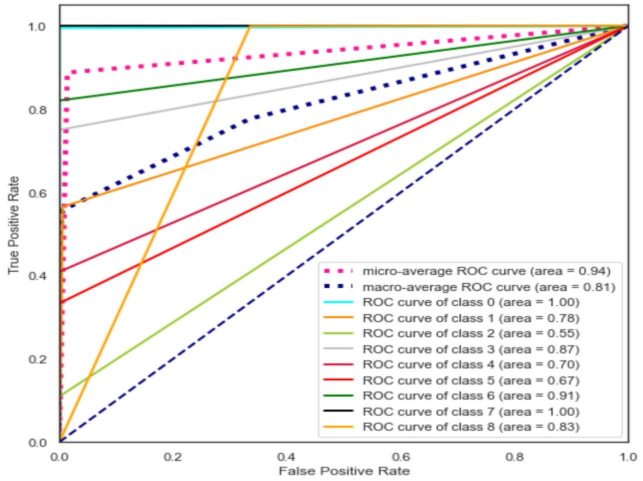
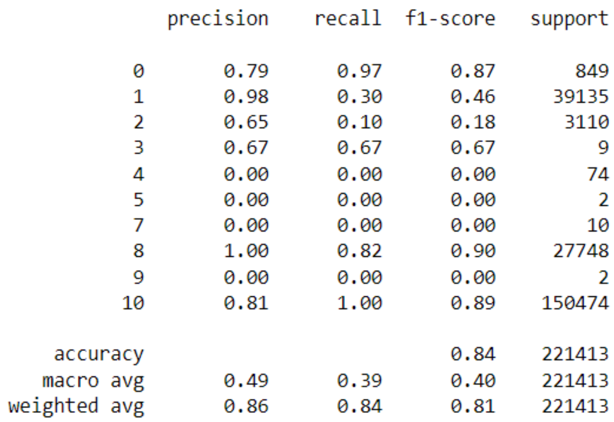


Fig. 8. The ROC Curve of CatBoost Model.

D. CNN Classifier's Experiments

CNN classifier is a deep learning algorithm is built based on many dense layers, as we explained in the methodology section. The proposed CNN model has ten layers and before train this model, the data is transferred into integers by applying 'MinMaxScaler' from the Sklearn library, and also the 'get_dummies' function from the Panda's library is applied to all used labels in this model, to change categorical variable into dummy/indicator variables. Unlike XGBoost, Random Forest, and CatBoost models, the classification report of the CNN model shows all used labels even if some labels have 0% for precision, recall, and f1-score metrics. However, the overall accuracy got by this model is lower than these models with a value of 84%, as shown in Fig. 9. The encoding numbers for this model is as follow: Attack=0, Benign=1, C&C=2, C&C-FileDownload=3, C&C-HeartBeat=4, C&C-HeartBeat-FileDownload=5, C&C-Mirai=6, C&C-Torii=7, DDoS=8, FileDownload=9, and PartOfAHorizontalPortScan=10. The C&C-Mirai label has not appeared in the classification report since it has only one case, while C&C-Torii, C&C-HeartBeat, C&C-HeartBeat-FileDownload, and FileDownload labels have appeared, but the measure matrices have 0%. The highest precision achieved is 100% for DDoS and 98% for Benign. Furthermore, the highest recall was 100% for PartOfAHorizontalPortScan, 97% for the Attack, while the highest f1-score was 90%, 89%, 87%, for DDos, PartOfAHorizontalPortScan, Attack labels, respectively.



```
              precision    recall  f1-score   support

           0       0.79      0.97      0.87       849
           1       0.98      0.30      0.46     39135
           2       0.65      0.10      0.18      3110
           3       0.67      0.67      0.67         9
           4       0.00      0.00      0.00        74
           5       0.00      0.00      0.00         2
           7       0.00      0.00      0.00        10
           8       1.00      0.82      0.90     27748
           9       0.00      0.00      0.00         2
          10       0.81      1.00      0.89    150474

    accuracy                           0.84    221413
   macro avg       0.49      0.39      0.40    221413
weighted avg       0.86      0.84      0.81    221413
```

Fig. 9. The Classification Report of CNN Model.

On the other hand, the ROC curve was drawn for this model, where the highest area values were 100% for FileDownload, C&C-FileDownload, Attack, and C&C-HeartBeat-FileDownload classes, 93% for DDos class as shown in Fig. 10. Moreover, the micro-average ROC curve achieved the highest area value with 98% as compared to the previous models, they only attained 94%. Furthermore, the macro-average ROC curve has a 'nan' value because of including the C&C-Mirai label data, when we remove this label from being trained, this value change to 82%, which is the highest value compared to the mentioned models.

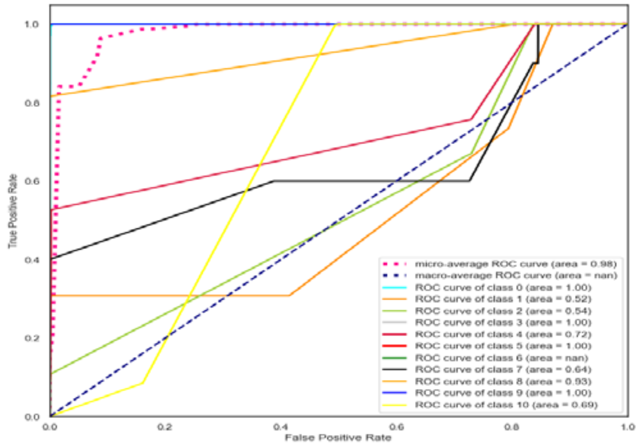
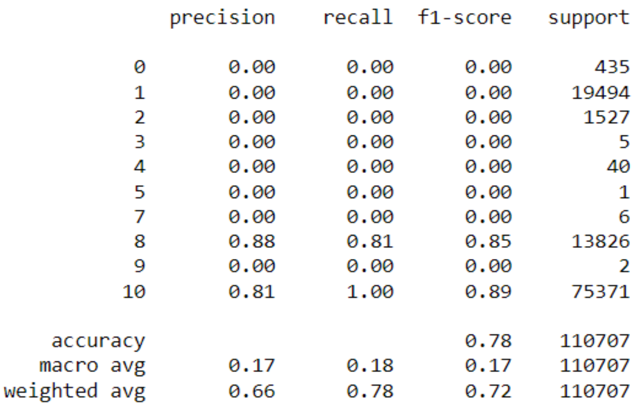


Fig. 10. The ROC Curve of CNN Model.

E. LSTM Classifier's Experiments

LSTM model is applied using various LSTM cells starting from 50 to 2000 cells, but the results were not good as the previous models. The highest overall accuracy is 78%; however, only DDoS and PartOfAHorizontalPortScan showed better results in the classification report of this model, while the rest had 0% for the used measure matrices, as depicted in Fig. 11. The encoding numbers for this model is exactly as in the CNN model.

On the other hand, the ROC curve was drawn for this model, where the highest area values were 99% for C&C-HeartBeat-FileDownload and C&C-FileDownload, 89% for C&C-Torii, 82% for DDoS, and 88% for PartOfAHorizontalPortScan label, as depicted in Fig. 12.



```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       435
           1       0.00      0.00      0.00     19494
           2       0.00      0.00      0.00      1527
           3       0.00      0.00      0.00         5
           4       0.00      0.00      0.00        40
           5       0.00      0.00      0.00         1
           7       0.00      0.00      0.00         6
           8       0.88      0.81      0.85     13826
           9       0.00      0.00      0.00         2
          10       0.81      1.00      0.89     75371

    accuracy                           0.78    110707
   macro avg       0.17      0.18      0.17    110707
weighted avg       0.66      0.78      0.72    110707
```

Fig. 11. The Classification Report of LSTM Model.

These values are the lowest among all other used models. Furthermore, the micro-average ROC curve achieved an area value of 88% and the macro-average ROC area reached 57% when we removed class number 6 from being calculated.

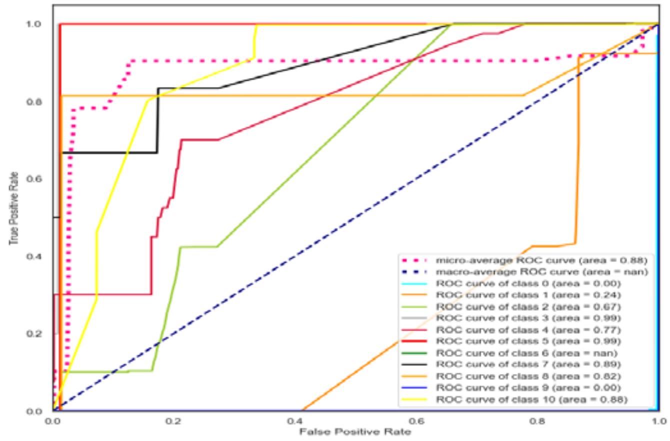
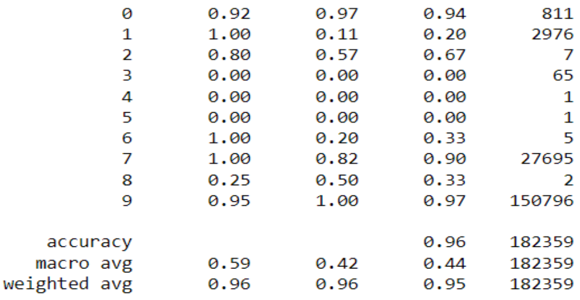


Fig. 12. The ROC Curve of LSTM Model.

However, an experiment was conducted for all the classifiers used in this paper. First, by using the CNN model, the accuracy of this model reached 96% when the Okiru and benign labels were excluded from being trained and tested, as shown in Fig. 13.

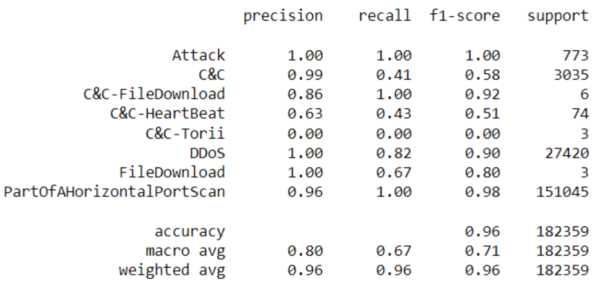
Furthermore, the RF model when Okiru and benign labels were removed and achieved the same accuracy results as CNN, as depicted in Fig. 14.



```
           0       0.92      0.97      0.94       811
           1       1.00      0.11      0.20      2976
           2       0.80      0.57      0.67         7
           3       0.00      0.00      0.00        65
           4       0.00      0.00      0.00         1
           5       0.00      0.00      0.00         1
           6       1.00      0.20      0.33         5
           7       1.00      0.82      0.90     27695
           8       0.25      0.50      0.33         2
           9       0.95      1.00      0.97    150796

    accuracy                           0.96    182359
   macro avg       0.59      0.42      0.44    182359
weighted avg       0.96      0.96      0.95    182359
```

Fig. 13. The Classification Report of the CNN Model without Okiru and benign Labels.



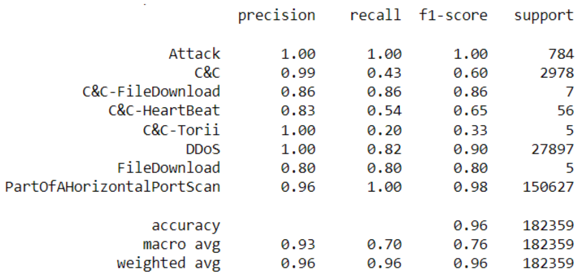
```
                          precision    recall  f1-score   support

                  Attack       1.00      1.00      1.00       773
                     C&C       0.99      0.41      0.58      3035
        C&C-FileDownload       0.86      1.00      0.92         6
           C&C-HeartBeat       0.63      0.43      0.51        74
               C&C-Torii       0.00      0.00      0.00         3
                    DDoS       1.00      0.82      0.90     27420
            FileDownload       1.00      0.67      0.80         3
PartOfAHorizontalPortScan       0.96      1.00      0.98    151045

                accuracy                           0.96    182359
               macro avg       0.80      0.67      0.71    182359
            weighted avg       0.96      0.96      0.96    182359
```

Fig. 14. The Classification Report of the RF Model without Okiru and benign Labels.

Additionally, by applying the same experiment as in CNN and RF model, the XGBoost model got the same accuracy as presented in Fig. 15.



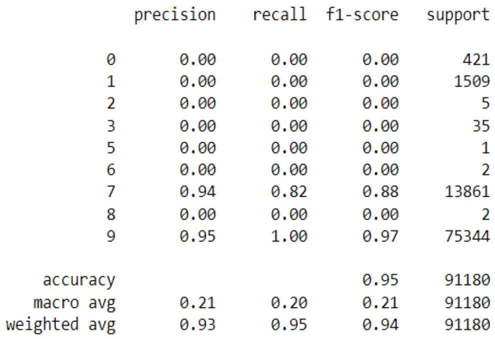
```
                          precision    recall  f1-score   support

                  Attack       1.00      1.00      1.00       784
                     C&C       0.99      0.43      0.60      2978
        C&C-FileDownload       0.86      0.86      0.86         7
           C&C-HeartBeat       0.83      0.54      0.65        56
               C&C-Torii       1.00      0.20      0.33         5
                    DDoS       1.00      0.82      0.90     27897
            FileDownload       0.80      0.80      0.80         5
PartOfAHorizontalPortScan       0.96      1.00      0.98    150627

                accuracy                           0.96    182359
               macro avg       0.93      0.70      0.76    182359
            weighted avg       0.96      0.96      0.96    182359
```

Fig. 15. The Classification Report of the XGBoost Model without Okiru and benign Labels.

Finally, the same accuracy was achieved too by using the Catboost classifier, as shown in Fig. 16, while the LSTM classifier got the lowest accuracy among all others, with a value of 95%, as depicted in Fig. 17.

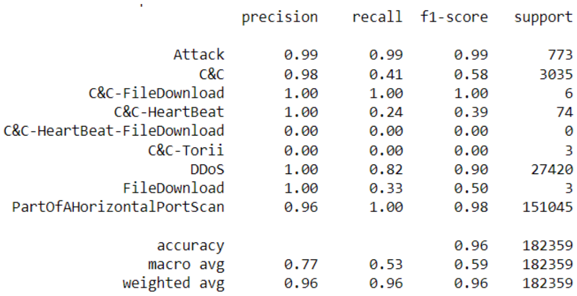


```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       421
           1       0.00      0.00      0.00      1509
           2       0.00      0.00      0.00         5
           3       0.00      0.00      0.00        35
           5       0.00      0.00      0.00         1
           6       0.00      0.00      0.00         2
           7       0.94      0.82      0.88     13861
           8       0.00      0.00      0.00         2
           9       0.95      1.00      0.97     75344

    accuracy                           0.95     91180
   macro avg       0.21      0.20      0.21     91180
weighted avg       0.93      0.95      0.94     91180
```

Fig. 16. The Classification Report of the Catboost Model without Okiru and benign Labels.



```
                           precision    recall  f1-score   support

                   Attack       0.99      0.99      0.99       773
                      C&C       0.98      0.41      0.58      3035
         C&C-FileDownload       1.00      1.00      1.00         6
            C&C-HeartBeat       1.00      0.24      0.39        74
C&C-HeartBeat-FileDownload       0.00      0.00      0.00         0
                C&C-Torii       0.00      0.00      0.00         3
                     DDoS       1.00      0.82      0.90     27420
             FileDownload       1.00      0.33      0.50         3
PartOfAHorizontalPortScan       0.96      1.00      0.98    151045

                 accuracy                           0.96    182359
                macro avg       0.77      0.53      0.59    182359
             weighted avg       0.96      0.96      0.96    182359
```

Fig. 17. The Classification Report of the LSTM Model without Okiru and benign Labels.

V. Discussion

First, only 100000 rows from the twenty-malware traffic captures were loaded, in which 11 malware labels and one benign label have existed. We have done this way to use a balanced dataset that includes all labels. However, some steps must be carried out before using the classifiers, a preprocessing step was applied to remove irrelevant or missing data from the dataset, and then a feature engineering technique was performed to extract the most significant features and thus reduce the dataset's dimensionality. Three experiments have been done to investigate the best accuracy of detection based on multi-class classifications. The first one was to include all existed labels but the best accuracy achieved is not exceeded 74%. In the second and third experiments, one or more labels are removed from being classified. According to the experiments' outcomes done in the second one, RF, Catboost, and XGBoost models achieved the highest accuracy with 89% for all malware labels except the Okiru label. The CNN results were lower than RF, Catboost, and XGBoost algorithms, with a value of 84%. The worst accuracy was attained by using LSTM with a value of 78%, and we have to improve the parameters of cells size up to 4000 but not enhance. Furthermore, to enhance the accuracy more than the ones recorded in the first and second experiments, a third one was conducted. Furthermore, when we remove the benign and Okiru labels from being trained, the accuracy of RF, Catboost, and XGBoost algorithms raised to 96%. Besides, the CNN results recorded lower accuracy than RF, Catboost, and XGBoost algorithms, for all labels except Okiru; however, when we excluded the benign label data from being trained, the overall accuracy increased to 96% based on the classification report outcomes.

When comparing our methodology outcomes, almost the same accuracy of detection has been achieved as the authors had in [6] and [7], where the best-recorded accuracy was 74% for the best model by including all labels of the IoT-23 dataset. However, the model achieved a higher accuracy of detection reached 89% for the best model by including all labels except Okiru malware.

Moreover, some studies achieved a higher accuracy more than 96% which was attained by the best models of this work. However, these studies are not analyzed all existed labels of the IoT-23 dataset. For example, the authors in [8] involved detection for four malware types, while in [9] and [10] papers only five malware types were classified, and eight and nine malware types were analyzed in [11] and [12], respectively. Finally, in ML context RF and boosting algorithms are the best candidates for the proposed security system. Based on the performance of the RF in this paper it performs the training in less time. The authors [22] recommend Catboost for better prediction in their model. They have less consumed of time cost and high performance to embed in IoT devices to detect in real-time. Table III shows the comparison our models with other work.

TABLE III. Comparison our Models with other Work

Dataset	Number of labels	Technique	Accuracy
A. K. Sahu [13]	8	CNN	96 %
Dr. R. Thamaraiselvi [15]	binary	RF, Naive Bayes, Support Vector Machine, and Decision Tree	99 %
B. Roy et al. [23]	5	LSTM and BRNN	72 %
H. HaddadPajouh et al. [24]	CC	LSTM and BNN	84 %
Amiya Kumar Sahu[25]	9 without benign	CNN and LSTM	96 %
Our models all labels	12	RF, CatBoost, Xgboost, CNN, and LSTM	74 %
Our models without okiru	11	RF, CatBoost, Xgboost, CNN, and LSTM	89 % except CNN 84% and LSTM 78%
Our models without benign	10	RF, CatBoost, Xgboost, CNN, and LSTM	96 % except LSTM 95%

VI. Conclusion

The IoT environment deals with massive data that must be protected from being modified or stolen by attackers. This research paper proposed the application of five deep and ML algorithms to detect malware in network traffic based on the IoT-23 dataset. RF, Catboost, Convolutional Neural Network, Long Short-Term Memory (LSTM), and XGBoost classifiers are implemented and evaluated. Three experiments have been conducted for each classifier, to evaluate the best performance matrices that can be recorded among all classifiers. In the first evaluation task, the best accuracy achieved is 74% by using RF, Catboost, and XGBoost models for classifying all labels of the IoT-23 dataset. While the second experiment was to detect all labels except Okiru malware, the same models achieve the highest accuracy with a value of 89%. Finally, the last one was done without Okiru and benign labels, all models except LSTM get an accuracy of 96%.

In conclusion, machine and DL models can perform detection tasks with a high degree of accuracy and reliability. Analysis of a huge amount of traffic data can be accomplished to detect various types of intrusion. The paradigm of distributed detection can provide a deep analysis functionality particularly if the device operating pattern is involved. However, the available datasets in the literature are intended for central traffic characterization.

VII. Further Work

The work applied in this paper is limited to using malware traffic captures and only reading the first 100000 rows from each capture, which leads to unbalance data, especially for C&C-Mirai (1 one row), C&C-HeartBeat (341 rows), FileDownload (13), C&C-FileDownload (43), C&C-Torii (30), and C&C-HeartBeat-FileDownload (8). Therefore, to enhance the overall accuracy, we can remove the small size malware types from being classified. Besides, extra datasets can be used such as IoT Network Intrusion, BoT-IoT, and MQTT-IoT-IDS2020, to collect more balanced data for creating a big dataset to improve the detection accuracy, especially in DL techniques. Moreover, using more advanced CNN models such as vgg16 or vgg19 can be an avenue for further evaluation. Finally, the available datasets for evaluation consider centralized methodologies; the research community demands other datasets involving end-system operating patterns which may detect novel attack.

Acknowledgment

This research is supported by Aljouf University - Vice Presidency for Graduate Studies and Scientific Research as a part of scientific publication support program.

References

[1] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, “Network intrusion detection for IoT security based on learning techniques,” IEEE Communications Surveys \& Tutorials, vol. 21, no. 3, pp. 2671–2701, 2019.

[2] T. Qiu, N. Chen, K. Li, M. Atiquzzaman, and W. Zhao, “Thirdquarter 2018,‘How can heterogeneous Internet of things build our future: a survey,’” IEEE Commun. Surv. Tutorials, vol. 20, no. 3, pp. 2011–2027, 2018.

[3] T. P. J. Kanimozhi V, “Artificial Intelligence for anomaly detection by employing deep learning strategies in IoT networks using the trendy IoT-23 big data from Google’s Tensorflow2.2,” Research Square, 2021, doi: https://doi.org/10.21203/rs.3.rs-364763/v1.

[4] “2022 SonicWall Cyber Threat Report,” sonicwall, 2022. https://www.sonicwall.com/2022-cyber-threat-report/.

[5] N.-A. Stoian, “Machine Learning for anomaly detection in IoT networks: Malware analysis on the IoT-23 data set,” University of Twente, 2020.

[6] N. Liang, Y. and Vankayalapati, “Machine Learning and Deep Learning Methods for Better Anomaly Detection in IoT-23 Dataset Cybersecurity,” Canada, 2021.

[7] SNEHA, “IoT Network Anomaly Detection Using Machine Learning and Deep Learning,” Rajasthan, INDIA, 2021.

[8] J. Vitorino, R. Andrade, I. Praça, O. Sousa, and E. Maia, “A Comparative Analysis of Machine Learning Techniques for IoT Intrusion Detection,” arXiv preprint arXiv:2111.13149, 2021.

[9] V. Dutta, M. Choraś Michałand Pawlicki, and R. Kozik, “A deep learning ensemble for network anomaly and cyber-attack detection,” Sensors, vol. 20, no. 16, p. 4583, 2020.

[10] I. Ullah and Q. H. Mahmoud, “Design and development of a deep learning-based model for anomaly detection in IoT networks,” IEEE Access, vol. 9, pp. 103906–103926, 2021.

[11] I. Ullah and Q. H. Mahmoud, “Design and development of a deep learning-based model for anomaly detection in IoT networks,” IEEE Access, vol. 9, pp. 103906–103926, 2021.

[12] N. Abdalgawad, A. Sajun, Y. Kaddoura, I. A. Zualkernan, and F. Aloul, “Generative Deep Learning to detect Cyberattacks for the IoT-23 Dataset,” IEEE Access, 2021.

[13] A. K. Sahu, S. Sharma, M. Tanveer, and R. Raja, “Internet of Things attack detection using hybrid Deep Learning Model,” Computer Communications, vol. 176, pp. 146–154, Aug. 2021, doi: 10.1016/j.comcom.2021.05.024.

[14] Bram. Dartel, “Malware detection in IoT devices using Machine Learning.,” University of Twente, 2021.

[15] Dr. R. Thamaraiselvi and S. Anitha Selva Mary, “Attack and Anomaly Detection in IoT Networks using Machine Learning,” International Journal of Computer Science and Mobile Computing, vol. 9, no. 10, pp. 95–103, Oct. 2020, doi: 10.47760/ijcsmc.2020.v09i10.012.

[16] D. Nanthiya, P. Keerthika, S. B. Gopal, S. B. Kayalvizhi, T. Raja, and R. S. Priya, “SVM Based DDoS Attack Detection in IoT Using Iot-23 Botnet Dataset,” in 2021 Innovations in Power and Advanced Computing Technologies (i-PACT), Nov. 2021, pp. 1–7. doi: 10.1109/i-PACT52855.2021.9696569.

[17] B. M. Khammas, “The Performance of IoT Malware Detection Technique Using Feature Selection and Feature Reduction in Fog Layer,” IOP Conference Series: Materials Science and Engineering, vol. 928, no. 2, p. 022047, Nov. 2020, doi: 10.1088/1757-899X/928/2/022047.

[18] T. G. Palla and S. Tayeb, “Intelligent Mirai Malware Detection for IoT Nodes,” Electronics (Basel), vol. 10, no. 11, p. 1241, May 2021, doi: 10.3390/electronics10111241.

[19] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, and S. Venkatraman, “Robust Intelligent Malware Detection Using Deep Learning,” IEEE Access, vol. 7, pp. 46717–46738, 2019, doi: 10.1109/ACCESS.2019.2906934.

[20] S. Garcia, A. Parmisano, and M. J. Erquiaga, “IoT-23: A labeled dataset with malicious and benign IoT network traffic (Version 1.0. 0)[Data set]. Zenodo.” 2020.

[21] N. Abdalgawad, A. Sajun, Y. Kaddoura, I. A. Zualkernan, and F. Aloul, “Generative Deep Learning to detect Cyberattacks for the IoT-23 Dataset,” IEEE Access, 2021.

[22] C. Bentéjac, A. Csörg\Ho, and G. Mart\'\inez-Muñoz, “A comparative analysis of gradient boosting algorithms,” Artificial Intelligence Review, vol. 54, no. 3, pp. 1937–1967, 2021.

[23] B. Roy and H. Cheung, “A Deep Learning Approach for Intrusion Detection in Internet of Things using Bi-Directional Long Short-Term Memory Recurrent Neural Network,” in 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), Nov. 2018, pp. 1–6. doi: 10.1109/ATNAC.2018.8615294.

[24] H. HaddadPajouh, A. Dehghantanha, R. Khayami, and K.-K. R. Choo, “A deep Recurrent Neural Network based approach for Internet of Things malware threat hunting,” Future Generation Computer Systems, vol. 85, pp. 88–96, Aug. 2018, doi: 10.1016/j.future.2018.03.007.

[25] A. K. Sahu, S. Sharma, M. Tanveer, and R. Raja, “Internet of Things attack detection using hybrid Deep Learning Model,” Computer Communications, vol. 176, pp. 146–154, Aug. 2021, doi: 10.1016/j.comcom.2021.05.024.