International Journal of Advances in Scientific Research and Engineering (ijasre)

E-ISSN : 2454-8006

DOI: 10.31695/IJASRE.2024.1.3

Volume 10, Issue 1
January - 2024

IoT Intrusion Detection System based on Machine Learning Algorithms using the UNSW-NB15 dataset

Nogbou Georges ANOH[1], Tiémoman KONE[2], Joel Christian ADEPO[3], Jean François M'MOH[4], Michel BABRI[5]

Research Scholar [1-4], Professor [5]

Department of Computer and Digital Science

Virtual University of Côte d'Ivoire

Abidjan

Côte d'Ivoire

ABSTRACT

The evolution of communications systems with the advent of IoT is leading to an increase in attacks against them. This is due to the fact that the security of connected objects in the IoT is an emerging area which still requires preventive solutions against various attacks. At the network security level, Intrusion Detection Systems (IDS) are used to analyze network data and detect abnormal behavior in the network. In this work, we implemented different machine learning models to build an intrusion detection system based on the UNSW NB15 dataset. To do this, we did data cleaning and feature engineering on the data in the pre-processing phase. Then we used various models such as logistic regression, support vector machine (SVM) classifier, decision tree, random forest, eXtreme Gradient Boosting (XGBoost) in order to predict attacks. Finally, an intrusion detection system is trained on various machine learning algorithms and we selected the most effective model. Experiments were carried out on the UNSW-NB15 dataset and subsequently we compared other machine learning algorithms, and this means that the random forest model on important parameters has a clear advantage in the detection of rare abnormal behaviors.

Key Words: Intrusion detection system, Machine learning algorithms, Random forest, SVM, UNSW-NB15 dataset.

1. INTRODUCTION

The evolution of communications systems with the advent of IoT is leading to an increase in attacks against them. This is due to the fact that the security of connected objects in the IoT is an area still in development which requires prevention solutions against various attacks. At the network security level, Intrusion Detection Systems (IDS) are used to analyze network data and detect abnormal behavior in the network [1]. These IDSs aim to recognize different attacks by analyzing different data sources, mainly network traffic and system event logs. To implement this concept of intrusion detection, specific tools are necessary: IDS (Intrusion Detection System). They will automatically collect data representative of system activity (server, application, system, network), analyze it and alert administrators when signs of attack are detected. The different types of intrusion detection system can be classified into two categories [1]: (i) The network IDS or NIDS (Network based IDS), (ii) The System IDS or HIDS (Host based IDS). One of the biggest challenges in intrusion detection is ensuring that warnings are only caused by real attacks and that each attack is an escalation alert for administrators. A complete intrusion detection system consists of several parts, each of which has a specific and essential task in the detection process. We distinguish the following different blocks: (i) Data source from which we can check if an intrusion is taking place, (ii) Detection engine which will analyze the data received from previous sources to report events, (iii) Response to detection. The main use of an IDS is to warn of intrusion. To achieve this, they have several features at their disposal, such as triggering alarms in the management interface or sending e-mails to security engineers.

IDS performance remains mixed, given the high number of false alarms. In order to find solutions, the construction of

innovative intrusion detection systems becomes necessary and must have means of assessing them. In this work, we will implement an intrusion detection system based on artificial intelligence in IoT. The designed system takes into account different machine learning models and a UNSW NB15 dataset.

The rest of this work is structured into four (4) sections. The first section presents intrusion detection systems from the literature. The materials and methods are presented in section 2. The results from the implementations are presented in section 3. Finally, in section 4, we present the conclusion.

2. LITERATURE SURVEY

In this section, we present summaries of the work carried out regarding AI-based intrusion detection systems.

According to AlSawafi et al., intrusion detection system (IDS) approaches using machine learning to detect and mitigate attacks in Internet networks are not suitable for analyzing Internet of Things traffic. Therefore, the authors propose an IDS system using the hybridization of supervised and semi-supervised deep learning for the classification of known and unknown abnormal network traffic in the Internet of Things environment. Additionally, they developed a new specialized Internet of Things dataset, called IoTR-DS, using the RPL protocol [2].

In this work, Al-Haija et al. proposed an efficient and intelligent attack detection and classification system based on deep learning in IoT networks. The proposed system consists of feature engineering, feature learning and traffic classification subsystems. These subsystems have been implemented and tested in this work. For the evaluation of system, the authors used the NSL-KDD dataset which includes all key attacks in IoT [3].

To detect a range of popular cyber-attacks that could threaten IoT devices, Anthi et al., proposed a three-layer intrusion detection system (IDS) based on a supervised approach. The three main layers are: 1) the first layer analyzes the network, identifies IoT devices and classifies them taking into account their behavior in the network, 2) the second layer helps identify malicious/normal packets originating IoT devices on the network when an attack occurs, and 3) the third layer is responsible for classifying the attack that was deployed into the corresponding type [4].

According to Çakir et al., methods for detecting and preventing DIS Flooding attacks have not been sufficiently presented in the literature. As a result, they offer a high-performance detection system for DIS Flooding attacks by applying logical regression (LR) and Support Vector Machine type automatic learning methods [5].

In this book [6], security challenges in IoT networks were presented, as well as well-known attacks, APT attacks and threat models in IoT systems. the authors presented signature- and anomaly-based IDSs and then hybrid IDSs in IoT networks. Statistical perspectives on ML-based methods frequently applied to combat network intrusions have been highlighted in this work.

According to Gassais et al., intrusion detection systems developed on the basis of network activity for the protection of smart devices are made difficult by the heterogeneous technologies involved in the IoT. However, on many systems, intrusions cannot be detected easily or reliably from network traces. It is in this context that the authors proposed an architecture based on several sensors and actuators and an analysis system for intrusion detection. This solution uses tracking techniques performed by smart devices to automatically obtain their behavior. Finally, the analysis device processes the data collected by the devices to detect anomalies and issue alerts whenever an intrusion is detected, so that intrusion prevention measures can be taken [7].

In their work, Jain et al. proposed a new hybrid intrusion detection model considering spatial and temporal characteristics using Deep Learning, Long Short-Short Memory (LSTM) techniques and Convolutional Neural Network (CNN) to achieve better attack detection accuracy. To train and test their model, the authors used two different datasets, namely UNSW-NB15 and NSL-Botnet, to check the adaptability of the model to different datasets [8].

According to Sahani et al., most of the existing ML-based IDS for smart grid are based on anomaly or signature-based detection methods. Improved versions of anomaly or signature-based detection methods or other directions of IDS using new ML techniques are necessary to achieve the ambitious goals of an IDS to be deployed in the smart grid

DOI: 10.31695/IJASRE.2024.1.3

with its vulnerability characteristics. Although an intrusion detection system is designed to detect any anomaly present in the system and accordingly alert the system operator to take appropriate action, the ML method should be able to evaluate the impact factor. Since the smart grid performs time-sensitive operations, early detection of anomalies can enable the system to take more proactive measures to minimize potential damage to the system [9].

In their work, Sharma et al. were interested in security threats in the RPL protocol and attacks that could affect the network. To build, train and study the performance of a new intrusion detection system, four different attack vectors were experimented. To achieve this, the methodology used was to: i) Design a simulation environment to simulate RPL attacks using contiki-Cooja simulator, ii) Create a dataset that can be used to classify five different scenarios and iii) Analyze features extracted from network traffic packets using a correlation-based approach to propose a new machine learning model [10].

In their work, Somashekar et al. proposed a novel prediction model fusing classifiers for intrusion detection using machine learning techniques. They also proposed model retraining for unknown attacks to increase classification efficiency in intrusion detection systems [11].

This research examines several machine learning and deep learning strategies and standard datasets to improve IoT security performance. The authors developed an algorithm for detecting denial of service (DoS) attacks using a machine learning algorithm [12].

To accurately and automatically differentiate normal traffic from malicious traffic, an intrusion detection system (IDS) based on a new set of features synthesizing the BoT-IoT dataset is developed by the authors. From this dataset is extracted a set of IoT-specific characteristics to help the machine learning model detect any suspicious activity in the network. This study also compares various supervised machine learning classifiers for anomaly and attack detection in IoT networks [13].

In their work, Zahra et al., proposed a hierarchical algorithm named ULEACH, adapted to the heterogeneity of perceptual nodes of the IoT in order to select the group head nodes for the IDS placement strategy. A dynamic intrusion detection model based on game theory was built to reduce the problem of energy consumption and attack detection efficiency [14].

Zhou et al. proposed a machine learning-based classification model for detecting RPL-related attacks in IoT networks. A new dataset was generated by constructing various network models to address the unavailability of the required dataset, optimal feature selection to improve model performance, and a lightweight machine-based algorithm with gradient boosting optimized for attack detection based on multi-class classification [15].

We note that in the literature, different deep learning methods and architectures have been applied to intrusion detection. These proposed deep learning algorithms have different performance depending on the selected input datasets and features. However, using the same learning methods and techniques does not always guarantee the same results between different articles. It should be noted that all researchers approach this problem in the same way. Although some articles provide different IDS configuration and discovery methods, most of them are focused on alert management technology. Among the different methods proposed, data mining techniques have recently aroused great interest. Data mining is the primary solution for assessing alert quality and addressing nuisance issues in aggressive intrusion detection systems. Then, researchers used hybrid data mining techniques to obtain better results. Despite the reduction in false positives, these methods still need to be improved because they still have weaknesses.

3. RESEARCH METHODOLOGY

3.1. Machine learning methods

In this section, several machine learning algorithms are presented.

Support vector machine is a machine learning algorithm that solves classification problems. It aims at finding the separation between two classes of objects with the idea that the wider the separation, the more robust the classification [16]. It can be used to solve discrimination problems, that is, to determine which class a sample belongs to. The SVM

DOI: 10.31695/IJASRE.2024.1.3

method has been generalized to be applied to a regression problem or to the prediction of time series [17].

Random forests are made up of a set of decision trees. These trees are distinguished from each other by the subsample of data on which they are trained. They provide predictive models for classification and regression. Finally, this machine learning algorithm makes it possible to make the prediction based on the aggregation of several decision trees [18].

Neural networks or artificial neural networks are a subset of machine learning and form the backbone of deep learning algorithms. Neural networks are made up of a set of neurons distributed over a set of layers. They are characterized by the number of neurons in each layer, the number of layers, the nature of the relationships between neurons, the assembly and activation functions of each neuron [18].

3.2. Description of dataset

The dataset chosen for this study is UNSW-NB15 provided by the University of New South Wales (UNSW). Raw network packets from the UNSW-NB 15 dataset were created by the IXIA PerfectStorm tool in the Australian Center for Cyber Security (ACCS) Network Span Lab to generate a mix of actual modern normal activity and of synthetic contemporary attack behavior [19]. The tool was used to capture 100 GB of raw traffic (e.g. Pcap files). The dataset contains nine attack families, namely Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Recognition, Shellcode and Worms. The use of Argus, Bro-IDS tools and development of twelve (12) algorithms to generate a total of forty-nine (49) features with class labels. These features are described in the UNSW-NB15_features.csv file. The total number of records is two (2) million five hundred and forty thousand zero forty-four (2,540,044), stored in four CSV files, UNSW-NB15_1.csv, UNSW NB15_2.csv, UNSW-NB15_3.csv and UNSW-NB15_4.csv, respectively. The ground truth table is named UNSW-NB15_GT.CSV and the list of event files is named UNSW-NB15_LIST_EVENTS. A partition of this dataset is configured as training set and testing set, UNSW_NB15_training-set.csv and UNSW_NB15_testing-set, respectively. The number of records in the training set is one hundred and seventy-five thousand, three hundred and forty-one (175,341) records and the test set is eighty-two thousand, three hundred and thirty-two (82,332) records of different attack types and normal types. The UNSW-NB15 dataset attempts to simulate a modern network environment by incorporating most modern stealth attacks. This table details the ten traffic types presented in the dataset: normal, obfuscated, analytical, backdoor, DOS, exploit, generic, reconnaissance, and worm. The detailed breakdown of these categories in terms of number of records per attack and the distribution of training/testing sets are shown in the table below.

As shown in Table 1 below, the dataset consists of 45 attributes, two (2) of which are dependent variables. Two subsets of data can be obtained from the original data set depending on the dependent variable or one of these subsets was used to develop a two-class anomaly IDS and the other a multi-attack IDS. Attack distribution sites are included in the attack_cat attribute, and the label attribute consists of normal instances and attack instances, denoted by 0 and 1, respectively.

Table 1. Description Dataset UNSW-NB15

Description of the dataset		
Number of attributes		45
Number of independent variables		43
Number of dependent variables		2
Details of the first variable	Name : label	
	Normal	Attack
	37,000	45,332
Details of the second variable	Name : attack_cat	
	Normal	37,000
	Acknowledgement	3,496
	Backdoors	583

DOI: 10.31695/IJASRE.2024.1.3

	Back	4,089
	Exploits	11,132
	Analysis	677
	Fuzzers	6,062
	worms	44
	shellcode	378
	Generic	18,871

As explained in [20], UNSW NB-15 is a modern dataset consisting of many variations of contemporary attack types. Tools like Argus and Bro-IDS were used in the process of creating this dataset to extract all possible features and label the records. To ensure the authenticity of the current attack vectors, attack behavior was considered based on the Common Vulnerabilities and Exposures (CVE) website.

3.3. Evaluation parameters

This part explains the most commonly used evaluation metrics to measure the performance of machine learning methods for IDS. All evaluation metrics are based on various attributes used in the confusion matrix, which is a two-dimensional matrix that provides information about the actual class and the predicted class, including the following: (i) True Positive (TP) : Data instances correctly predicted as an attack by the classifier, (ii) False negatives (FN): Data instances incorrectly predicted as normal instances, (iii) False positives (FP): Data instances classified as wrong as an attack, (iv) True Negative (TN): Instances correctly classified as normal instances.

The diagonal of the confusion matrix represents the correct predictions, while the off-diagonal elements are the wrong predictions of a certain classifier. In addition, the different evaluation measures used in this study are as follows.

3.3.1. Accuracy

It is the ratio of correctly classified instances to the total number of instances. This is a useful performance measure only when a data set is balanced. It is defined by expression (1).

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

3.3.2. False alarm rate

It is also called false positive rate and is defined as the ratio of falsely predicted attack samples to all samples that are normal.

$$FPR=\frac{FP}{FP+TN} \quad (2)$$

3.3.3. Area under the ROC Curve

The AUC (Area Under the ROC Curve) measures the area under the ROC curve and the performance of an interpretable and robust model.

3.3.4. F1-Score

F1 is a classification metric that measures the ability of a model to predict well in terms of accuracy. It is also complex but has the advantage of being robust in the presence of unbalanced data. It combines the notions of precision and recall as indicated in the following expressions.

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

$$recall=\frac{TP}{TP+FN}, \quad (4)$$

DOI: 10.31695/IJASRE.2024.1.3

$$F1=2\times\frac{precision\times recall}{precision+recall}. \quad (5)$$

3.4. Descriptive diagram of the methodology used

Machine learning methods usually involve following a few main steps, as shown in figure 1, namely the data preprocessing phase, the training phase, then the training model and the testing phase. For all proposed solutions, the dataset is first preprocessed to transform it into the format suitable for use by the algorithm. This step usually involves encoding and normalization. Sometimes the dataset requires cleaning in terms of removing entries with missing data and duplicate entries, which is also done during this phase. The preprocessed data is then randomly divided into two parts, the training dataset and the testing dataset.

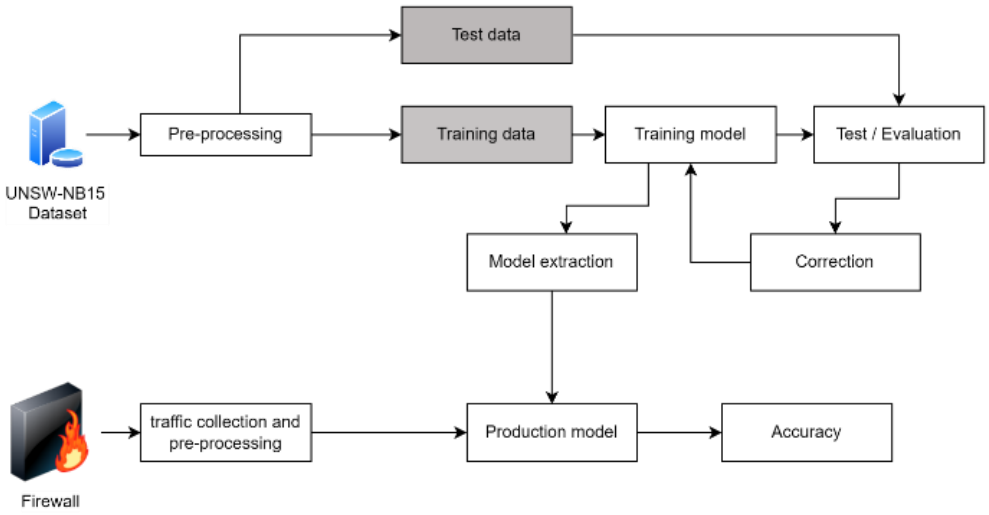

Figure 1. Diagram of the system implementation phases

4. RESULTS AND DISCUSSION

The first phase consisted of pre-processing the data in order to clean it and format it. Exploratory data analysis was used to represent the heat map of the dataset of correlation values of all features as shown in figure 2.

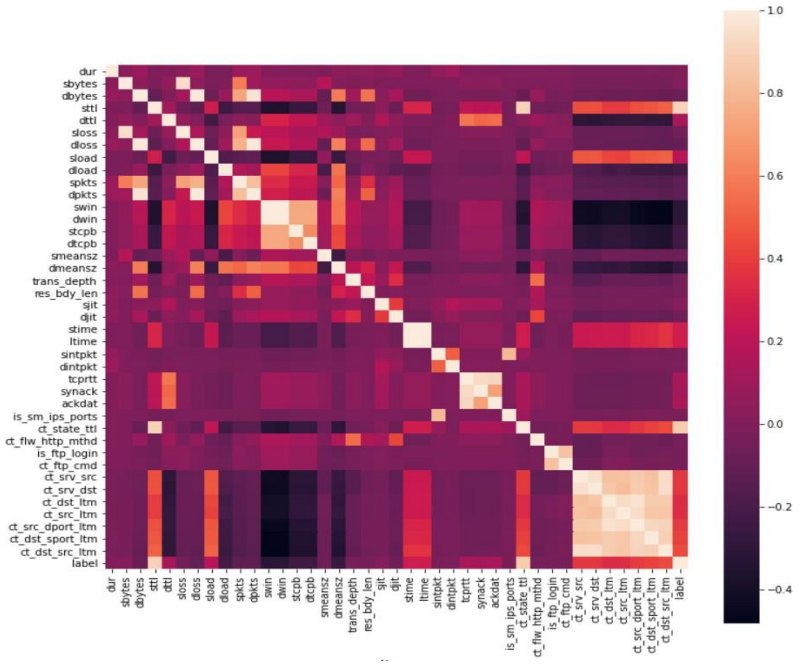

Figure 2. Correlation heat map of all features

DOI: 10.31695/IJASRE.2024.1.3

This representation makes it possible to highlight the most correlated features.

The stage of the intrusion detection training process will consist of a binary classification benchmark using a number of machine learning algorithms to learn, compare, and improve the accuracy of the results. After training the model with the best hyper parameters, we check the performance of the model with the machine learning algorithms.

4.1. Logistic regression

The best performance of the model on training and testing data is shown in figure 3.

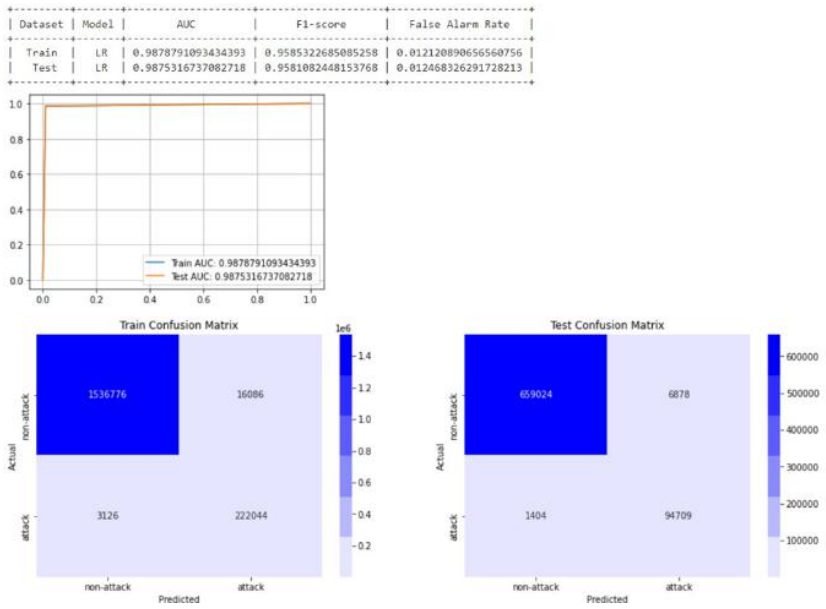

Figure 3. Results on train and test confusion matrix with logistic regression

We see that the AUC scores for training and testing are very close, meaning the model is not overfitted. When we take the confusion matrix, the model has some false positives in the results but this model is efficient and works well.

4.2. SVM

The performance of the SVM model is shown in figure 4.

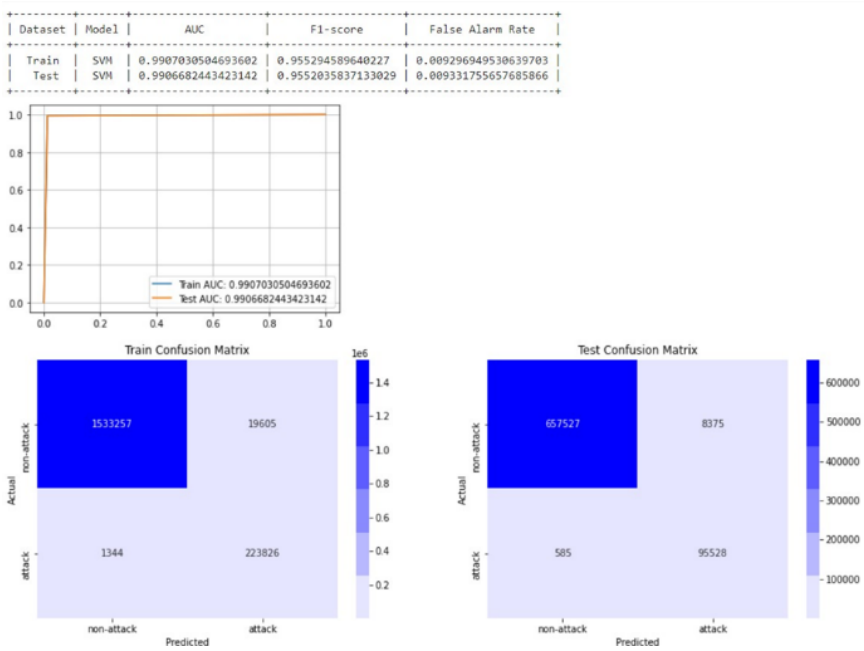

Figure 4. Results on train and test confusion matrix with SVM

DOI: 10.31695/IJASRE.2024.1.3

In the first figure and the curve we see that the training and test score are very close, so no overfitting here, obtaining a better AUC and FAR value than logistic regression. When we take the confusion matrix we see that the number of FPs has increased compared to LR but there are very few FN points in the result. The FAR value of this model is really good.

4.3. Decision tree

The performance of this model is shown in figure 5.

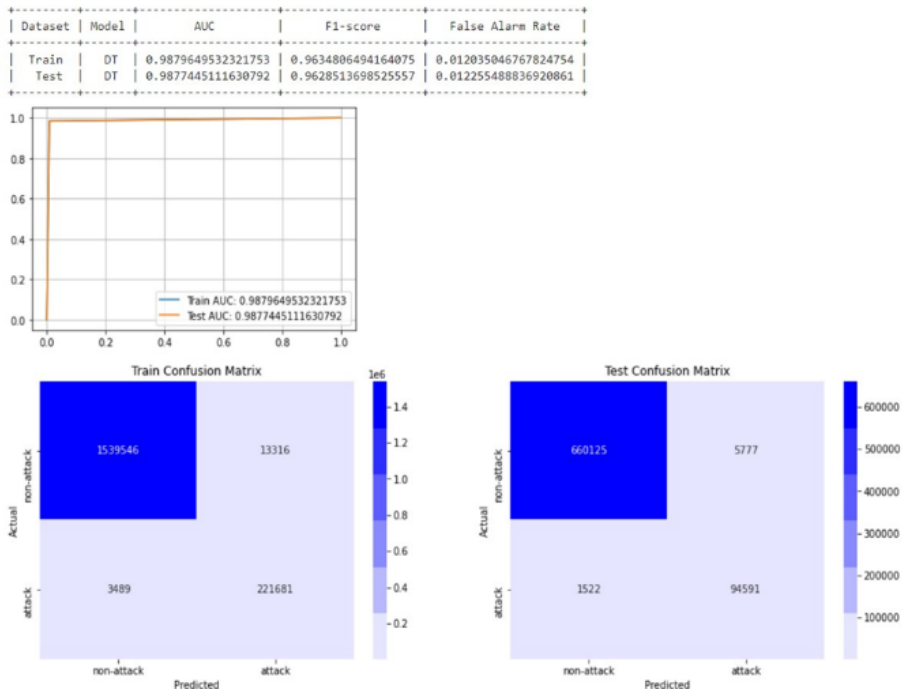

Figure 5. Results on train and test confusion matrix with decision tree

On the first figure and the curve we see that the train and test scores are very close, so there is no overfitting here, the F1-score value obtained with this model is better than both (2) others above. For the confusion matrix we see that the number of false positives for this model has also been reduced.

4.4. Random Forest

We have used various parameters for this classifier with appropriate values. Performance mainly depends on "n_estimators", "max_depth" and two (2) other parameters. The best parameters for the model: criterion='gini', max_depth=22, min_samples_split=6, n_estimators=300.

The performance of this model is shown in figure 6.

DOI: 10.31695/IJASRE.2024.1.3

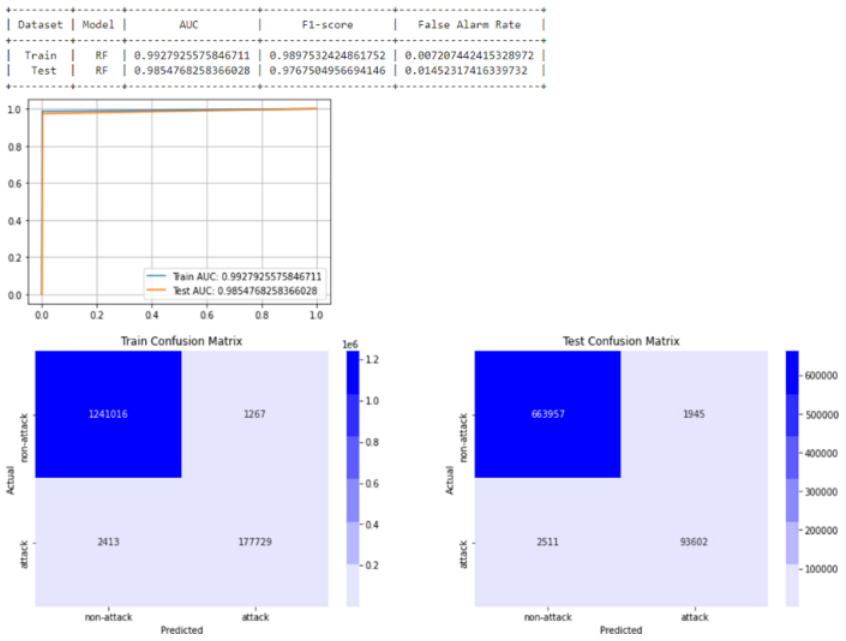

Figure 6. Results on train and test confusion matrix with Random Forest

When we take the first figure and the curve, the training and test scores are close, but there is a difference between the training and test scores compared to the above models. So if we compare it with the above model, it is an overfitting of the training data. But the deviation is very small, so there is not much overfitting. For the confusion matrix the number of false positives dropped sharply, but the number of false negatives increased.

4.5. XGBoost

For this model, there are many hyperparameters to tune, such as 'learning_rate', 'max_depth', 'colsample_bylevel', 'subsample' and 'n_estimators'.

The performance of this model is shown in figure 7.

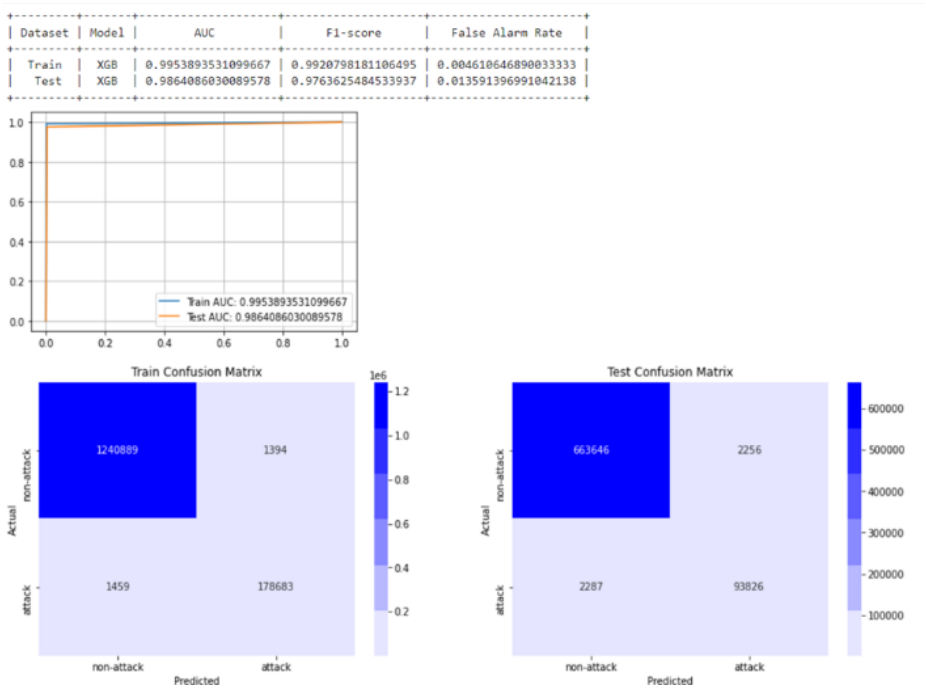

Figure 7. Results on train and test confusion matrix with Xgboost

When we take the first figure and the curve, the training and test score are close, but compared to the above models,

DOI: 10.31695/IJASRE.2024.1.3

there is a gap between the training score and the test score. So it is overfitted on the training data if we compare its result with the models above. But the deviation is very small so not much overfitting. For the confusion matrix of the training data, the FAR is very low, but in the test, FN and FP are almost equal.

4.6. Voting classifier

This method combines several different models for a better prediction. This model is trained using three (3) models namely the best Decision Tree, the best Random Forest and the best XGBoost. We will apply this method on the important features of the dataset. The list of important features is shown in the figure 8. below.

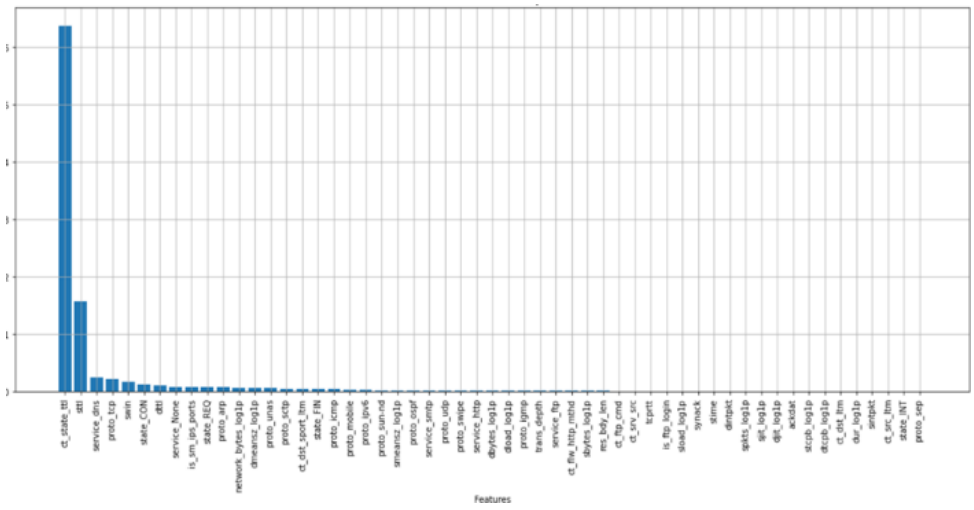

Figure 8. List of important features of the dataset

So there are a total of 55 features that have some importance for this classification task. The results of this model on important data features are shown in the figure 9 below.

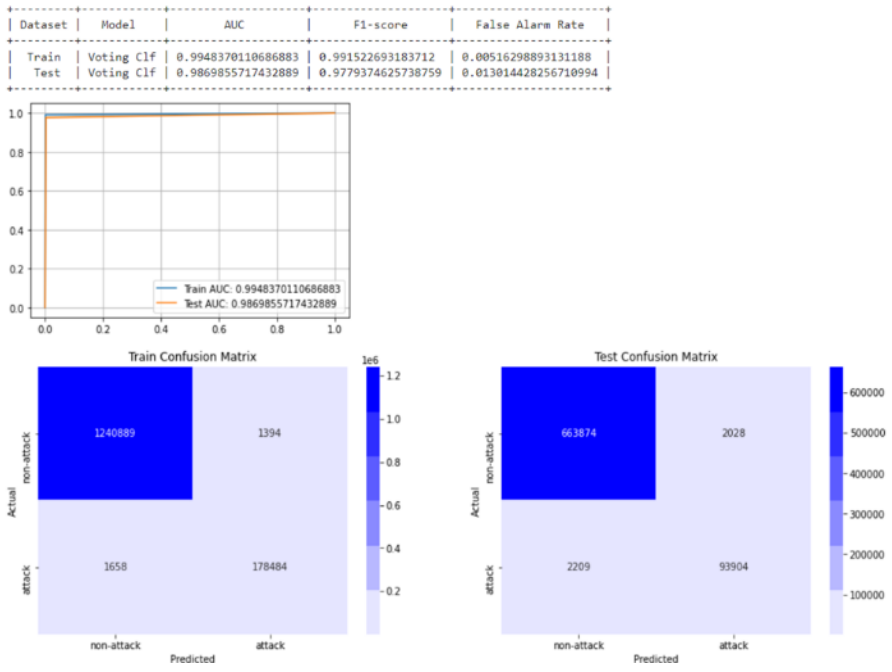

Figure 9. Results on train and test confusion matrix with voting classifier

In the first figure and the curve, this model has a higher AUC score than any other model. There is a gap in training and testing AUC, and a larger gap in training and testing F1 and FAR. For the train confusion matrix, the FAR is very low, but there are still few FPs and FNs in the tests, which are approximately equal in number.

DOI: 10.31695/IJASRE.2024.1.3

The performance of Random Forest related to important features are specified in the figure 10 below. We see an improvement in performance. This model in the first figure gives the highest F1-score of all the models we have trained so far.

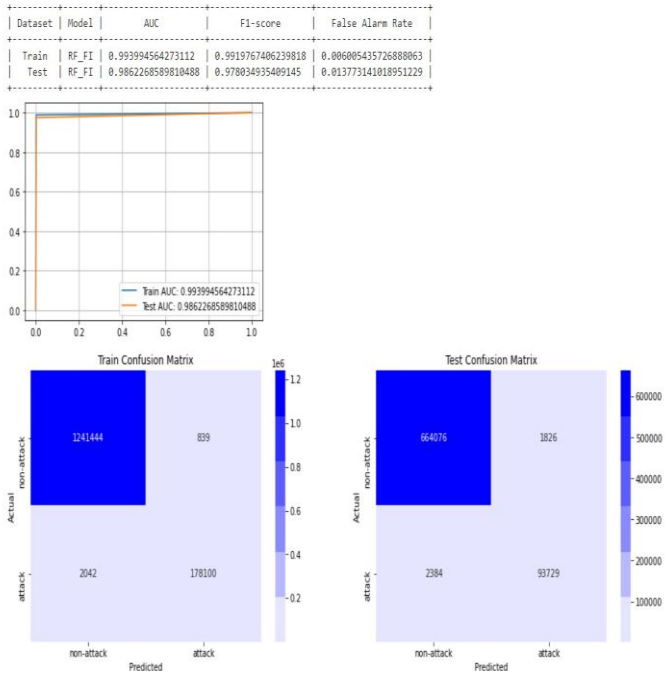

Figure 10. Results on train and test confusion matrix with Random forest and important features

4.7. Comparative study of results

The performances obtained with the different machine learning algorithms in the context of classification are presented in the table 2 below.

Table 2. Performances of machine learning algorithms

Model Name	AUC	F1 score	False alarm rate
Logistic regression	0.9875	0.9581	0.0124
Linear SVM	0.9906	0.9552	0.0093
Decision tree	0.9877	0.9628	0.0122
Random Forest	0.9854	0.9767	0.0145
XGB classifier	0.9864	0.9763	0.0135
Decision tree_F	0.9873	0.9624	0.0126
Random forest_F	0.9862	0.9780	0.0137
Voting Classifier	0.9869	0.9779	0.0130

All the different learning models we studied detect intrusions, then we took five (5) different models on the dataset from which we retrained two (2) good models on important features, and finally one voting classifier.

The random forest model on important parameters has the best F1 score out of all with the 97.80%, the FAR is very good at only 1.37%, that's why we can say that this model is good.

5. CONCLUSION

In this study, we have performed a cleaning of the dataset of the dataset, then an exploratory analysis of the data,

DOI: 10.31695/IJASRE.2024.1.3

then we created the pipeline composed of several sequential steps that execute everything from extraction and preprocessing of the data to the training and deployment of the model. We applied the machine learning models on the dataset data. In the implementation phase, we saw how feature selection and multiple classifier approaches affect the performance of IDS. This is why we worked on setting up an IDS detection system with a UNSW-NB15 dataset which contains all the new modern attacks. For future work, we intend to: (i) Perform the experiment on a balanced data set, (ii) Generate any new network-related functionality from the data provided or transform existing functionality, (iii) Use deep learning with an artificial neural network, (iv) Use other ensemble models such as the stacking classifier.

REFERENCES

[1] Poulmanogo Illy, "Intrusion detection systems (IDS)", 2018, doi: 10.13140/RG.2.2.10055.04001.

[2] Y.AlSawafi, A. Touzene, and R. Hedjam, "Hybrid Deep Learning-Based Intrusion Detection System for RPL IoT Networks," Journal of Sensor and Actuator Networks, vol. 12, no. 2, Art. No. 2, Apr. 2023, doi: 10.3390/jsan12020021.

[3] QA Al-Haija and S. Zein-Sabatto, "An efficient deep learning-based detection and classification system for cyber-attacks in IoT communication networks," 2020.

[4] E.Anthi, L. Williams, M. Slowinska, G. Theodorakopoulos, and P. Burnap, "A Supervised Intrusion Detection System for Smart Home IoT Devices," IEEE Internet Things J., vol. 6, no. 5, p. 9042-9053, Oct. 2019, doi: 10.1109/JIOT.2019.2926365.

[5] S.Çakir and N. Yalçin, "Detection of DIS Flooding Attacks in IoT Networks Using Machine Learning Methods," European Journal of Science and Technology, Nov. 2021, doi: 10.31590/ejosat.1014917.

[6] Z.Chen*et al.*, "Machine Learning-Enabled IoT Security: Open Issues and Challenges Under Advanced Persistent Threats", ACM Comput. Surv., vol. 55, no. 5, p. 1-37, May 2023, doi: 10.1145/3530812.

[7] A.Gassais, N. Ezzati-Jivan, J. M. Fernandez, D. Aloise, and M. Dagenais, "Multi-level host-based intrusion detection system for Internet of things," Journal of Cloud Computing-Advances Systems and Applications, vol. 9, no. 1, Art. no 1, 2020, doi: 10.1186/s13677-020-00206-6.

[8] S. Jain, P. M. Pawar, and R.Muthalagu, "Hybrid intelligent intrusion detection system for internet of things", Telematics and Informatics Reports, vol. 8, p. 100030, Dec. 2022, doi: 10.1016/j.teler.2022.100030.

[9] NOT.Sahani, R. Zhu, J.-H. Cho, and C.-C. Liu, "Machine Learning-based Intrusion Detection for Smart Grid Computing: A Survey," ACM Trans. Cyber-Phys. Syst., vol. 7, no. 2, p. 1-31, Apr. 2023, doi: 10.1145/3578366.

[10] M. Sharma, H.Elmiligi, F. Gebali, and A. Verma, "Simulating Attacks for RPL and Generating Multi-class Dataset for Supervised Machine Learning," in 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada : IEEE, Oct. 2019, p. 0020-0026. doi: 10.1109/IEMCON.2019.8936142.

[11] H.Somashekar and R. Boraiah, "Network intrusion detection and classification using machine learning predictions fusion," IJEECS, vol. 31, no. 2, p. 1147, August 2023, doi: 10.11591/ijeecs.v31.i2.pp1147-1153.

[12] b.Susilo and RF Sari, "Intrusion Detection in IoT Networks Using Deep Learning Algorithm," Information, vol. 11, no. 5, Art. no 5, May 2020, doi: 10.3390/info11050279.

[13]H.Tyagi and R. Kumar, "Attack and Anomaly Detection in IoT Networks Using Supervised Machine Learning Approaches," RIA, vol. 35, no. 1, p. 11-21, Feb. 2021, doi: 10.18280/ria.350102.

DOI: 10.31695/IJASRE.2024.1.3

[14] F. Zahra, N.Jhanjhi, SN Brohi, NA Khan, M. Masud, and MA AlZain, "Rank and Wormhole Attack Detection Model for RPL-Based Internet of Things Using Machine Learning," Sensors (Basel), vol. 22, no. 18, p. 6765, Sep 2022, doi: 10.3390/s22186765.

[15] M. Zhou, L. Han, H. Lu, and C. Fu, "IntrusionDetection System for IoT Heterogeneous Perceptual Network", Mobile Netw Appl, vol. 26, no. 4, p. 1461-1474, August 2021, doi: 10.1007/s11036-019-01483-5.

[16] b.Schölkopf and AJ Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. The MIT Press, 2018. doi: 10.7551/mitpress/4175.001.0001.

[17] "Support vector machine",*XLSTAT, Your data analysis solution*. https://www.xlstat.com/fr/solutions/functionalites/machine-a-vector-support (accessed September 23, 2023).

[18] M. KHICHANE, "Data Science with Microsoft Azure - Data Science",*https://www-eni-training-com.proxybib-pp.cnam.fr/portal/client/mediabook/home*, 2018. https://www.editions-eni.fr/livre/data-science-avec-microsoft-azure-maitriez-le-machine-learning-sur-cortana-intelligence-suite-9782409012785/la-data-science (accessed September 23, 2023).

[19] "CloudStor - CloudStor is powered by AARNet", CloudStor. https://cloudstor.aarnet.edu.au/plus/s/2DhnLGDdEECo4ys (accessed September 23, 2023).

[20] Mr.Souhail Et. Al., "Network Based Intrusion Detection Using the UNSW-NB15 Dataset", IJCDS, vol. 8, no. 5, p. 477-487, Jan. 2019, doi: 10.12785/ijcds/080505.

DOI: 10.31695/IJASRE.2024.1.3