

Machine learning-based intrusion detection system for detecting web attacks

Fatimetou Abdou Vadhil, Mohamed Lemine Salihi, Mohamedade Farouk Nanne

Department of Mathematics and Computer Science, Faculty of Sciences and Technics, University of Nouakchott, Nouakchott, Mauritania

Article Info

Article history:

Received Mar 20, 2023

Revised Sep 11, 2023

Accepted Sep 16, 2023

Keywords:

CIC-IDS-2017

Cybersecurity

Intrusion detection systems

Machine learning

Web attacks

ABSTRACT

The increasing use of smart devices results in a huge amount of data, which raises concerns about personal data, including health data and financial data. This data circulates on the network and can encounter network traffic at any time. This traffic can either be normal traffic or an intrusion created by hackers with the aim of injecting abnormal traffic into the network. Firewalls and traditional intrusion detection systems detect attacks based on signature patterns. However, this is not sufficient to detect advanced or unknown attacks. To detect different types of unknown attacks, the use of intelligent techniques is essential. In this paper, we analyse some machine learning techniques proposed in recent years. In this study, several classifications were made to detect anomalous behaviour in network traffic. The models were built and evaluated based on the Canadian Institute for Cybersecurity-intrusion detection systems dataset released in 2017 (CIC-IDS-2017), which includes both current and historical attacks. The experiments were conducted using decision tree, random forest, logistic regression, gaussian naïve bayes, adaptive boosting, and their ensemble approach. The models were evaluated using various evaluation metrics such as accuracy, precision, recall, F1-score, false positive rate, receiver operating characteristic curve, and calibration curve.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Fatimetou Abdou Vadhil

Department of Mathematics and Computer Science, Faculty of Sciences and Technics

University of Nouakchott

Nouakchott, Mauritania

Email: fatiab38@gmail.com

1. INTRODUCTION

Recent technologies such as artificial intelligence, big data, and internet of things. have made our lives exponentially dependent on the internet. Alongside this, however, the number of anomalous behaviours is also becoming increasingly important [1], [2]. Detecting anomalous network activity is a critical cybersecurity task that is becoming more and more of a focus, especially as we rely more and more on computers and smartphones in recent years [3]–[5]. Due to the global pandemic, we can say that our daily lives are shifting to the internet, which makes security issues more complicated than before. To detect abnormal activities in a computer or network, there is a special security device called network intrusion detection system (NIDS) [6], [7].

Intrusion detection systems (IDS) are security tools that aim to defend a system, perform countermeasures or generate alerts for a facility to take appropriate action when an attack occurs [8]. An IDS may be a software or hardware system designed to detect malicious actions on computer systems to enable the maintenance of system security [9]. The main purpose of IDS is to detect various types of abnormal network traffic that cannot be detected by a simple traditional firewall [10]. This is critical to achieve solid protection against malicious acts that compromise the availability, integrity or confidentiality of computer

systems [11], [12]. These systems can also be targeted at different areas. Some IDS play an extended firewall role and detect attacks as they enter the network, others monitor the network internally to intercept intruders or even collect network-wide information for central analysis. Most of these systems have a similar structure and set of components. There are two main types of intrusion detection systems [13].

Host intrusion detection system (HIDS) and network intrusion detection system (NIDS). The HIDS monitors the characteristics of a single host and the events occurring on that host to detect suspicious activity. NIDS monitors network traffic for specific network segments or devices and analyses the network.

Four main approaches are used for intrusion detection [13]: Signature-based IDS, anomaly-based IDS, hybrid-based IDS, and protocol-based IDS. Signature-based IDS detect hosts and malicious network activities based on known malicious patterns or sequences. Anomaly-based IDS show abnormal or anomalous system behaviour. It creates a profile of normal activity. If the normal activity exceeds the predefined threshold, it is considered an intrusion. Any deviation from the threshold is considered abnormal behaviour. Hybrid-based IDS combine the above two approaches (signature and anomaly-based IDS) to avoid the disadvantages and integrate the advantages. Protocol-based IDS monitor the protocols used by the system while performing an analysis of the state and dynamic behaviour and apply the legal use of the protocol.

Most of these types of IDS are still in traditional use and the cost of generating an appropriate signature for such an attack can be a considerable motivation for the use of learning-based approaches such as ML algorithms. These algorithms are progressing rapidly [14], [15], and are being followed with great interest in the field of cyber security [16]. Machine learning (ML) techniques can automatically learn to make decisions based on existing data, which is a very valuable advantage for monitoring computing environments.

In the field of intrusion detection, two types of ML algorithms are generally used: Supervised classification for misuse detection and unsupervised outlier/novelty classification for anomaly detection [17]. ML algorithms are increasingly being used to improve IDS and make it more efficient. A large number of research papers on intrusion detection fields using ML techniques have been published in the literature. For example, the authors of [18], [19] used support vector machine (SVM) to find anomalies in the knowledge discovery in database (KDD) dataset. Stein *et al.* [20] constructed an IDS model with artificial neural networks (ANN) based on the same dataset. Authors in [21], [22] proposed the use of decision trees and random forest. In addition, a hybrid approach combining two or more ML algorithms was presented in [23]. For more information on intrusion detection systems using ML methods, the reader is strongly advised to read the reviews provided in [24]–[26].

However, some recent papers find that conventional ML algorithms still perform poorly compared to other alternatives. The reason may be simply because the model parameters are not set appropriately or not set at all. Most algorithms provide many parameter values that can be used to improve model performance. Therefore, these parameter values can be adjusted to select the most optimal model. This article will review the optimization of conventional ML algorithms using hyper parameter tuning to achieve good results based on the values available in each algorithm.

This section analyses various research papers that use ML algorithms and exploit their performance for intrusion detection. The focus here is on studies that use the Canadian Institute for Cybersecurity - intrusion detection systems dataset released in 2017 (CIC-IDS-2017) dataset. This dataset was proposed by Sharafaldin *et al.* [27] in 2017. They tested it with seven algorithms, namely random forest (RF), ID3, k-nearest neighbours (KNN), multilayer perceptron (MLP), Adaboost, Naive Bayes (NB), quadratic discriminant analysis (QDA). They prove that the KNN, RF and ID3 algorithms give the best results with 98% accuracy compared to the other algorithms.

Vijayanand *et al.* [28] proposed an IDS that uses a genetic algorithm (GA) for variable selection and a support vector machine (SVM) for classification. In this paper, classification is based on a combination of several SVMs, each designed to detect a specific type of attack. The same algorithm is used by Aksu *et al.* [29] with other algorithms such as k-nearest neighbor (KNN) and decision tree (DT). The authors of this paper use the fisher score method for variable selection and achieved detection rates of 99.70%, 57.76%, and 99.00% for SVM, KNN, and DT classifications, respectively, using the denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks of CIC-IDS-2017 for testing.

Bansal [30] describes a data dimensionality reduction method called data dimensionality reduction (DDR) and uses conditional tree (CTree) [8], extreme gradient boosting (XGBoost) [31], SVM, and artificial neural network algorithms (ANN). Among these estimators, XGBoost was the most efficient with an accuracy of 98.93%. The authors of this paper used the entire dataset except for the normal traffic provided on Monday.

Boukhamla and Gaviro [32] used principal component analysis (PCA) to reduce the overall size of the CIC-IDS-2017 dataset. This work was used for Thursday and Friday data targeting various attacks such as DDoS, web attacks, port scans, infiltration, and botnets. KNN, Naive Bayes (NB), and C4.5 are the algorithms used for classification. DDoS attacks are perfectly detected by Naive Bayes and KNN with a very high

detection rate. However, Naive Bayes has a high false alarm rate, whereas KNN does not. In particular, the number of variables was reduced by about 75% of the total number of variables.

Ustebay *et al.* [33] proposes a hybrid IDS that combines three algorithms, namely reduced error pruning (REP) tree and random forest. They claim that the experimental results of this IDS demonstrate good performance in terms of detection rate, false alarm rate, accuracy, and duration compared to existing systems. The accuracy achieved in this proposal is 96.66%.

Hou *et al.* [34] presented a ML approach based on DDoS attack detection using netflow traffic analyzer (NTA). It mainly used four algorithms, namely AdaBoost, C4.5, support vector machine and random forest, against the data collected by netflow. This approach is then evaluated against the object dataset. Based on the results of the experiment, 97.4% accuracy was achieved using this approach.

Bansal and Kaur [35] proposed an intrusion detection approach using the XGBoost algorithm. This approach uses DoS/DDoS data from CIC-IDS-2017. The work is completed with 99.54% accuracy.

Aksu *et al.* [29] proposed an IDS using fisher score algorithm for selecting variables for normal and DDoS traffic. The algorithms used for attack classification are: SVM, KNN, and DT. In this work, the evaluation showed that KNN performed best with 30 variables selected, while SVM failed with 80 and 30 variables. After applying the fisher score, the amount of data was reduced by 60%. As a result of this work, KNN, and DT models scored 99% and 99%, and SVM scored 57% with 30 variables selected.

Alrowaily *et al.* [36] investigated the efficiency of seven ML algorithms, including RF, NB, DT, AdaBoost, MLP, quadratic discriminant analysis (QDA), and k-nearest neighbors (KNN). The result confirms the superiority of KNN on various performance evaluation metrics with 99% accuracy among the other selected algorithms. However, all the selected algorithms trained within acceptable time frame except this algorithm.

Thapa *et al.* [37] propose an ensemble model that combines ML and deep learning (DL) algorithms to achieve high performance metrics. They compared their models using the CIC-IDS-2017 dataset. In this article, the significance of variables was studied using classification and regression tree (CART) and classification using convolutional neural network (CNN). The accuracy obtained in this article is 99%.

Maseer *et al.* [38] review previous studies on ML and DL based IDS using a set of criteria with different datasets. In this article, 10 common supervised and unsupervised ML algorithms are evaluated. The supervised algorithms used are ANN, DT, KNN, NB, RF, SVM, and CNN, while the unsupervised algorithms include k-means, expectation-maximisation (EM), and self-organising maps (SOM). The best results were obtained with KNN (accuracy = 99.52%), DT (accuracy = 99.49%), and CNN (accuracy = 99.47%) as these models have higher recognition performance compared to other models.

The variables that can be used for designing and implementing an effective ML based IDS are analysed [39]. The selected variables are applied to different ML methods to test the efficiency. This research is conducted on the CIC-IDS-2017 dataset using 30% of the data and 100% of the data from Wednesday. The best result is obtained with the random forest, which achieves an accuracy of 99.9% and a false positive rate (FPR) rate of 0.02%.

A systematic approach to decision support in the selection of algorithms for the design of an IDS is presented [40]. The authors of this paper used the CIC-IDS-2017 dataset and selected 51 variables using the mean decrease in impurity (MDI) technique. They then evaluated the recognition performance of eight algorithms. The decision tree, random forest and multi-layer perceptron algorithms achieved 99% accuracy.

Elmrabit *et al.* [41] evaluated twelve ML algorithms in terms of their ability to detect abnormal behaviour in network practice. The evaluation is performed on the CIC-IDS-2017, UNSW-NB15, and industrial control system (ICS) cyberattacks datasets. The results of the evaluation show that the random forest algorithm performs better in terms of accuracy, precision, recall, F1 score, and receiver operating characteristic (ROC) curve for the three datasets used.

In terms of web attacks, Goryunov *et al.* [42] presented a study that includes the analysis and evaluation of different classifiers such as decision tree, random forest, AdaBoost, and logistic regression. The evaluation was done using the CIC-IDS-2017, more specifically web attacks (brute force, cross site scripting (XSS), and structured query language injection (SQLi)). Moreover, the top 10 features were selected, which is why the training time was very low in this study.

2. DATASET

The type of data is crucial in terms of quality and quantity for any ML problem, as a lot of important data is required for good training of ML algorithms. The outcome of any classifier depends on the trained data, so the quality of the data helps to achieve good results. Unfortunately, such quality datasets are also expensive and difficult to produce. In the field of intrusion detection, two free datasets are particularly popular despite some shortcomings: KDD Cup 99 and NSL-KDD. Three other recent datasets that address some of the shortcomings of earlier datasets are CIC-IDS-2017, CIC-IDS2018, and LITNET-2020.

Based on some reviews and studies [27], [43]–[45], we decided to use CIC-IDS-2017. This dataset was created by the Canadian Institute of Cybersecurity - University of New-Brunswick and generated by Sharafaldin *et al.* [27] in 2017. It is fully labelled and contains 78 features and seven main attack classes such as: web attacks, portscan, botnet, heartbleed, DoS/DDoS, and infiltration. In this paper, we use the data collected under Web Attacks, which consists of SQL Injection, XSS, and Brute Force.

- SQL injection: This is a vulnerability in an application where the attacker interferes with an application's queries to the database to allow unauthorised users to access the data.
- XSS: This attack occurs when the attacker injects malicious code into the victim's web application.
- Brute force: This attack tries a number of possible passwords to crack the administrator's password.

We focus on web attacks because, despite their importance, they are rarely the subject of research compared to other types of attacks. For example, many researchers focus on DoS/DDoS attacks. This could also be the reason why most datasets do not include this type of attack. Therefore, it is a matter of popularity of certain attacks over others.

3. METHODOLOGY

This section discusses the methodology used in this paper. After selecting the dataset, the next and most important step is to prepare it. This step is called pre-processing and includes analysis, processing, coding, and normalization. The dataset CIC-IDS-2017 is analysed and trained with five supervised ML algorithms: decision tree (DT) [46], random forest (RF) [47], logistic regression (LR) [48], Gaussian Naive Bayes (GNB) [49], and AdaBoost or adaptive boosting (AB) [50]. The models implemented with these algorithms are optimised by tuning their hyperparameters. This optimisation is performed despite the GridSearchCrossValidation technique.

3.1. Data acquisition

The first step of the proposed methodology is to import dataset. The data used contains 170,231 records. Table 1 shows the number of records by attack and the distribution between training and test data.

3.2. Data cleaning and analysis

The selected data (the "web attacks" file recorded on Thursday) contains 170,231 samples. In this section, we analyse this data to identify potential problems and possible errors such as missing values and duplicate columns. While searching for these values, we found that this dataset contains 270 missing values, ranging from not a number (NaN) to infinity. On the other hand, the Fwd header length column is found twice. All values that are NaN or infinity are removed, and then the duplicate column is also removed. It is also important to check the class equilibrium in the response features (or target) as this is very important for most ML algorithms. We have therefore analysed the feature in question and the classes are really unbalanced.

The majority class is the normal traffic class, which takes 168051 of the data. The rest is distributed among other classes, 1,507 for brute force, 652 for XSS, and 21 for SQL injection. This problem of imbalance can be solved by oversampling, followed by pre-emptive cleaning of possible overlap points.

3.3. Encoding

To train a ML model, the raw data must be prepared in a form that can be understood by the model. For this reason, digitisation of the data is an essential phase. In the proposed method, the one-hot encoder scheme is used to convert non-numeric variables into vectors. One-hot encoding is the most widely used encoding method [51]. It converts categorical values into vectors with minimal processing. It is a defensive feature of certain techniques. This is the most common method for handling such multiclass classification tasks. As shown in Table 2 the categorical classes are converted to vectors.

Table 1. Data distribution

Class	Data points	Training data	Testing data
Benign	168,051	134,461	33,590
Brute Force	1,507	1,187	320
XSS	652	518	134
SQLi	21	18	3

Table 2. One hot encoding

Class	Benign	Brute force	SQLi	XSS
Benign	1	0	0	0
Brute force	0	1	0	0
SQLi	0	0	1	0
XSS	0	0	0	1

3.4. Features scaling

Feature scaling is a technique often used in data preparation to facilitate its use by ML algorithms. The purpose of this scaling is to bring the values of the numerical features in the data set to a common scale, while preserving the differences in the value ranges of the individual features. Thus, by scaling the features,

the range of data-independent features is normalised. In data preprocessing, scaling is often done using one of two methods: normalisation or standardisation. Feature scaling should be performed during data preprocessing. We use a normalisation method known as min-max. It is the simplest method and involves scaling the range of features to scale the range in [0, 1]. The general normalisation is as (1).

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{1}$$

3.5. Classification and hyper-parameters tuning

In this phase, the data is submitted to the decision tree (DT), random forest (RF), logistic regression (LR), Gaussian Naïve Bayes (GNB), and adaptive boosting (AB) algorithms for training. To obtain the best model parameters, we used the GridSearchCV technique. This is a method for selecting the best model from a family of models parameterised by a grid of parameters, i.e. all the possibilities of the parameters are traversed in a grid of parameters. In addition, GridSearch performs cross-validation. Table 3 shows the details of the values used for each model. For each algorithm, the best values selected by GridSearchCV are fitted to build the model. Algorithm 1: Hyper parameter tuning algorithm used in this research for decision tree, random forest, and AdaBoost.

- a. $x \leftarrow 0$
- b. hyper parameter 1
- c. while $x \leq 314$ do
- d. Perform a 5-fold cross-validation on CIC-IDS-2017 train set
- e. Record the average 5-fold cross-validation accuracy
- f. Increment hyper parameter by 1
- g. Increment x by 1
- h. Choose the best hyper parameter which gives the highest average GridSearchCV accuracy (in step. e)

The Algorithm 1 shows the process of hyper parameters tuning for the models. This algorithm is used with GridSearchCV to select the good model. An overview of the proposed methodology is presented in Figure 1.

Table 3. Model’s hyper parameters values

Classifier	Hyper parameter	Values	Best values
DT	Estimator_criterion	['gini', 'entropy']	'entropy'
	Estimator_max_depth	[4, 5, 6, 7, 8]	7
	Estimator_criterion	['gini', 'entropy']	'entropy'
RF	Estimator_max_depth	[4, 5, 6, 7, 8]	8
	Estimator_max_features	['auto', 'sqrt', 'log2']	'auto'
	Estimator_n_estimators	[100, 200]	200
	Estimator_penalty	['L1', 'L2']	'L2'
LR	Estimator_C	[100; 10; 1; 0.1; 0.01; 0.001; 0.0001]	0.01
	Solver	['sag', 'lbfgs', 'saga']	'lbfgs'
GNB	Estimator_var_smoothing	[1e-11, 1e-12, 1e-13, 1e-14, 1e-15]	1e-13
	Estimator_criterion	['gini', 'entropy']	'entropy'
AB	Estimator_splitter	['best', 'random']	'best'
	N_estimators	[1, 2, 3, 4, 5, 6, 7, 8]	2
	Learning_rate	[0.01, 0.1, 1]	1

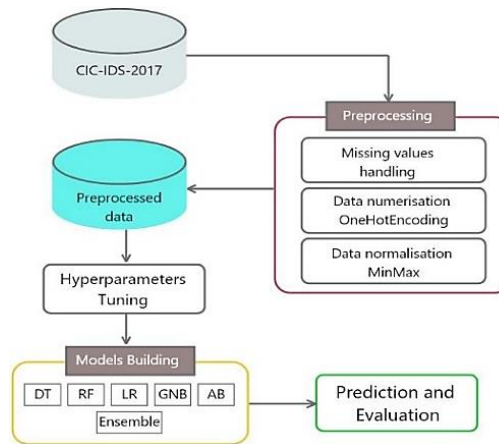


Figure 1. Methodology followed by the experiment

4. RESULTS AND DISCUSSION

The effectiveness of ML techniques has led researchers to use them in topical areas such as IoT [52], [53], online social network [54], DoS detection, DDoS [55]–[60], and distributed reflexion denial of service (DRDoS) [61]. However, obtaining good results depends on the techniques used, and the way in which we create our models. Generally speaking, improving the results of ML models can be done in several ways, for example by adjusting the model's hyperparameters as proposed in this article or by selecting the most relevant features [62]–[68]. On the other hand, this selection is in the framework of traditional models [69]–[74], this concept is still not valid, when it comes to DL models [75]–[80], if the model needs optimisation, in this case other methods must be sought, as the selection of the most relevant features is done automatically in most cases [81]. In this section, we present our experimental results using different evaluation metrics for the various ML techniques based on the proposed methodology. Afterward, we provide a comparison with some related works and discuss the performance of the proposed model. The models are implemented with a portable PC with the following characteristics: i7-9750H CPU, processor, 8 GB memory, hard disk 1 TB, and 64-bit Windows 10.

4.1. Evaluation metrics

After setting up the models, it is time to measure the performance by going through an evaluation stage. Each model needs to be tested to confirm its reliability based on four possible outputs, true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). The models used in this work were evaluated based on accuracy, precision, recall, F1 score, FPR, and training and prediction time. Where:

- TP: True positives are events that are correctly identified as abnormal
- FP: False positives are legal events that are incorrectly identified as abnormal
- TN: True negatives are incidents that are correctly identified as legal activities
- FN: False negatives can be defined as possible intrusive activity that the IDS passes through as normal activity.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (2)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (3)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (4)$$

$$\text{F1_score} = 2 * \frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \quad (5)$$

$$\text{FPR} = \frac{FP}{FP+TN} \quad (6)$$

4.2. Experimental results

The best results are shared by the ensemble method (accuracy and precision), AdaBoost (recall and FPR), and DT (F1_score). On the other hand, GNB was trained in a very reasonable time; the same is true for the prediction time of LR. The computation time depends on the number of hyperparameters and the set of values defined in each of the hyperparameters in grid search. The Table 4 presents the evaluation results to show how the models performed.

Table 4. Performance evaluation of models on CIC-IDS-2017 (web attacks)

Evaluation	Decision Tree	Random Forest	Logistic Regression	Gaussian Naïve Bayes	AdaBoost	Ensemble
Accuracy (%)	99.53	99.50	98.27	98.30	98.34	99.57
Precision (%)	99.56	99.52	99.36	97.97	99.16	99.59
Recall (%)	99.57	99.51	98.27	99.66	99.79	99.59
F1_score (%)	99.57	99.52	98.46	98.81	99.39	99.44
FPR (%)	0.004	0.004	0.017	0.003	0.002	0.004
Training time (s)	349.16	459.91	146.68	77.84	626.24	32.48
Prediction time (s)	0.23	0.20	0.13	1.02	0.55	1.19

In order not to be limited to the previously used evaluation metrics, we evaluated the models with other metrics such as ROC and calibration curves as shown in Figure 2. Other algorithms like gradient boosting, support vector machine, k-nearest neighbors were tested, but since we use the GridSearchCV technique, which

has very high computational costs, these algorithms could not be used. The subfigures of Figure 2 present (a) the results of ROC, (b) zoomed-in ROC with scores, and (c) calibration curve.

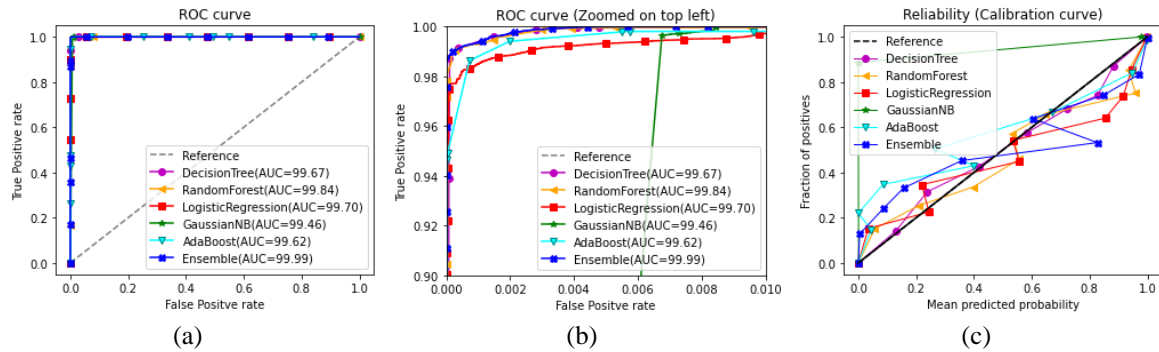


Figure 2. Area under curve (AUC) score, ROC, and calibration curves

4.3. Comparison with similar researches

The models implemented in this work were compared with some recent works. The selection of works is based on the CIC-IDS-2017 dataset and the five algorithms DT, RF, LR, GNB, and AB. Table 5 shows our results compared to those of other works. The results seem to be in the middle range of previous research. At this stage, there are some important factors to consider: (1) Some of this research uses feature selection, which always improves the results; and (2) Another factor is that some of them use the whole dataset, which also helps to improve the model, especially if it is a binary classification.

Table 5. Comparison of the results obtained with similar research result

Ref.	Dataset	Algorithm	A (%)	P (%)	R (%)	F1 (%)	FPR (%)	Training (S)	Prediction (S)
[27]	CIC-IDS-2017	RF	-	98.00	98.00	98.00	-	1908.23	-
	(All)	AB	-	77.00	84.00	77.00	-	1126.24	-
[39]	CIC-IDS-2017	DT	99.91	-	99.91	99.91	0.05	6.46	0.038
	(Dos / DDos)	RF	99.94	-	99.94	99.94	0.03	96.02	2.40
		LR	84.82	-	84.82	83.08	1.66	51.43	0.03
		GNB	74.42	-	25.57	35.55	66.22	1.01	0.15
		AB	94.26	-	94.26	94.02	2.01	84.20	3.15
[40]	CIC-IDS-2017	DT	-	99.00	99.00	99.00	-	-	0.23
	(All)	RF	-	99.00	98.00	98.00	-	-	1.32
		AB	-	75.00	85.00	80.00	-	-	21.26
[41]	CIC-IDS-2017	DT	99.80	99.80	99.80	99.80	-	-	-
	(All)	RF	99.90	99.90	99.90	99.90	-	-	-
		LR	91.50	91.40	91.50	91.00	-	-	-
		AB	81.80	76.90	81.80	76.00	-	-	-
[42]	CIC-IDS-2017	DT	97.50	97.30	94.60	96.90	-	1.53	-
	(Web Attacks)	RF	97.10	97.80	94.30	97.00	-	1.14	-
		LR	95.50	93.90	91.40	96.30	-	15.80	-
		AB	97.80	96.20	96.50	97.30	-	23.40	-
[56]	DDoS DNS	DT	99.46	-	-	-	-	-	-
[62]	NSL-KDD	LR	83.46	92.51	77.20	84.16	-	-	0.17
		RF	80.50	92.04	71.97	80.78	-	-	2.68
[63]	CIC-IDS-2018	LR	-	91.21	91.12	91.14	-	-	-
		RF	-	98.79	99.02	98.97	-	-	-
Our work	CIC-IDS-2017	DT	99.53	99.56	99.57	99.57	0.004	412.06	0.33
	(Web attacks)	RF	99.50	99.52	99.51	99.52	0.008	1516.11	2.65
		LR	98.27	99.36	98.27	98.46	0.017	185.82	0.08
		GNB	98.30	97.97	99.66	98.81	0.003	79.85	0.99
		AB	98.34	99.16	99.79	99.39	0.002	637.97	0.50
		Ensemble	99.57	99.59	99.59	99.44	0.004	33.60	1.18

A: Accuracy; P: Precision; R: Recall; F1: F1-measure; and FPR: False positive rate

4.4. Discussion

The results presented in this paper were obtained without a selection of variables. Thus, it may seem that our models can provide much better results by selecting the most relevant variables. On the other hand, we have to mention the imbalance of the CIC-IDS-2017 dataset. This phenomenon sometimes affects the

results of the models in such a way that most of the learning is done on the majority of the dataset, which means that the detection of minority samples is weak. Some researchers believe that binary classification can partially solve the imbalance problem. However, in this case, it is no longer possible to know in detail the positive detection for each attack. Finally, data balancing can solve this problem. There are a number of methods such as random oversampling, synthetic minority over-sampling technique (SMOTE), SMOTE-EEN, and SMOTE-Tomek.

5. CONCLUSION

An efficient intrusion detection system must be able to detect any kind of abnormal behaviour with high accuracy and low false alarm rate. In this work, we have used five ML algorithms and their ensemble approach to detect web attacks (such as SQL injection, brute force, and XSS) in the CIC-IDS-2017 dataset by applying k-fold cross validation with $K=5$. We used GridSearchCV for hyperparameter tuning to achieve better performance of the models. It provides high accuracy, precision, recall, and F1 score while maintaining low FPR. According to the results obtained, conventional classifiers can give competitive performance without any additions or adjustments. Thus, the research questions stated in section 2 can be answered. However, the results could be even better if the most important variables in the dataset had been selected. In the future, we will focus on the models that are optimised using feature selection. We will also go through other optimisation methods.

REFERENCES

- [1] H. Chourabi *et al.*, "Understanding smart cities: an integrative framework," in *2012 45th Hawaii International Conference on System Sciences*, Jan. 2012, pp. 2289–2297. doi: 10.1109/HICSS.2012.615.
- [2] F. A. Vadhil, M. L. Salihi, and M. F. Nanne, "Toward a secure ELK stack," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 17, no. 7, pp. 139–143, 2019.
- [3] A. Singh, J. Amutha, J. Nagar, S. Sharma, and C.-C. Lee, "LT-FS-ID: Log-transformed feature learning and feature-scaling-based machine learning algorithms to predict the k-barriers for intrusion detection using wireless sensor network," *Sensors*, vol. 22, no. 3, p. 1070, Jan. 2022, doi: 10.3390/s22031070.
- [4] K. Albulayhi, Q. Abu Al-Haija, S. A. Alsubihany, A. A. Jillepalli, M. Ashrafuzzaman, and F. T. Sheldon, "IoT intrusion detection using machine learning with a novel high performing feature selection method," *Applied Sciences*, vol. 12, no. 10, p. 5015, May 2022, doi: 10.3390/app12105015.
- [5] A. Singh, J. Amutha, J. Nagar, S. Sharma, and C.-C. Lee, "AutoML-ID: automated machine learning model for intrusion detection using wireless sensor network," *Scientific Reports*, vol. 12, p. 9074, May 2022, doi: 10.1038/s41598-022-13061-z.
- [6] Y. Diogenes and E. Ozkaya, "Counter modern threats and employ state-of-the-art tools and techniques to protect your organization against cybercriminals," in *Cybersecurity – Attack and Defense Strategies*, 2nd Editio., Packt Publishing Ltd., 2019.
- [7] 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, doi: 10.1109/COMST.2019.2896380.
- [8] T. Hothorn, K. Hornik, and A. Zeileis, "ctree: Conditional inference trees," pp. 1–34, [Online]. Available: <https://cran.r-project.org/web/packages/partykit/vignettes/ctree.pdf>
- [9] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, Jan. 2013, doi: 10.1016/j.jnca.2012.09.004.
- [10] S. Axelsson, "Intrusion detection systems: a survey and taxonomy," pp. 1–27, 2000, [Online]. Available: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=7a15948bdc530e2c1deedd8d22dd9b54788a634>
- [11] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets, and challenges," *Cybersecurity*, vol. 2, p. 20, Dec. 2019, doi: 10.1186/s42400-019-0038-7.
- [12] F. A. Vadhil, M. F. Nanne, and M. L. Salihi, "Importance of machine learning techniques to improve the open source intrusion detection systems," *Indonesian Journal of Electrical Engineering and Informatics (IJEI)*, vol. 9, no. 3, Sep. 2021, doi: 10.52549/ijeie.v9i3.3219.
- [13] J. A. M. Sidey-Gibbons and C. J. Sidey-Gibbons, "Machine learning in medicine: a practical introduction," *BMC Medical Research Methodology*, vol. 19, p. 64, Dec. 2019, doi: 10.1186/s12874-019-0681-4.
- [14] C.-J. Wu *et al.*, "Machine learning at Facebook: understanding inference at the edge," in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Feb. 2019, pp. 331–344. doi: 10.1109/HPCA.2019.00048.
- [15] K. Bresniker, A. Gavrilovska, J. Holt, D. Milojicic, and T. Tran, "Grand challenge: applying artificial intelligence and machine learning to cybersecurity," *Computer*, vol. 52, no. 12, pp. 45–52, Dec. 2019, doi: 10.1109/MC.2019.2942584.
- [16] A. Sapegin, "High-speed security log analytics using hybrid outlier detection," 2019. doi: <https://doi.org/10.25932/publishup-42611>.
- [17] D. S. Kim, H.-N. Nguyen, and J. S. Park, "Genetic algorithm to improve SVM based network intrusion detection system," in *19th International Conference on Advanced Information Networking and Applications (AINA'05) Volume 1 (AINA papers)*, 2005, vol. 2, pp. 155–158. doi: 10.1109/AINA.2005.191.
- [18] Y. Guang and N. Min, "Anomaly intrusion detection based on wavelet kernel LS-SVM," in *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology*, Oct. 2013, pp. 434–437. doi: 10.1109/ICCSNT.2013.6967147.
- [19] S. Kumar and A. Yadav, "Increasing performance of intrusion detection system using neural network," in *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, May 2014, pp. 546–550. doi: 10.1109/ICACCCT.2014.7019145.
- [20] G. Stein, B. Chen, A. S. Wu, and K. A. Hua, "Decision tree classifier for network intrusion detection with GA-based feature selection," in *Proceedings of the 43rd annual Southeast regional conference - Volume 2*, Mar. 2005, pp. 136–141. doi: 10.1145/1167253.1167288.




- [21] A. Tesfahun and D. L. Bhaskari, "Intrusion detection using random forests classifier with SMOTE and feature reduction," in *2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies*, Nov. 2013, pp. 127–132. doi: 10.1109/CUBE.2013.31.
- [22] S. Abdulrezzak and F. Sabir, "An empirical investigation on snort NIDS versus supervised machine learning classifiers," *Journal of Engineering*, vol. 29, no. 2, pp. 164–178, Feb. 2023, doi: 10.31026/j.eng.2023.02.11.
- [23] T. Shon and J. Moon, "A hybrid machine learning approach to network anomaly detection," *Information Sciences*, vol. 177, no. 18, pp. 3799–3821, Sep. 2007, doi: 10.1016/j.ins.2007.03.025.
- [24] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in Cloud," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 42–57, Jan. 2013, doi: 10.1016/j.jnca.2012.05.003.
- [25] E. E. Abdallah, W. Eleisah, and A. F. Ootom, "Intrusion detection systems using supervised machine learning techniques: a survey," *Procedia Computer Science*, vol. 201, pp. 205–212, 2022, doi: 10.1016/j.procs.2022.03.029.
- [26] Y. Hamid, M. Sugumaran, and V. Balasaraswathi, "IDS using machine learning - current state of art and future directions," *British Journal of Applied Science & Technology*, vol. 15, no. 3, pp. 1–22, Jan. 2016, doi: 10.9734/BJAST/2016/23668.
- [27] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 2018, pp. 108–116. doi: 10.5220/0006639801080116.
- [28] R. Vijayanand, D. Devaraj, and B. Kannapiran, "Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection," *Computers & Security*, vol. 77, pp. 304–314, Aug. 2018, doi: 10.1016/j.cose.2018.04.010.
- [29] D. Aksu, S. Üstebay, M. A. Aydin, and T. Atmaca, "Intrusion detection with comparative analysis of supervised learning techniques and fisher score feature selection algorithm," in *Communications in Computer and Information Science*, 2018, pp. 141–149. doi: 10.1007/978-3-030-00840-6_16.
- [30] A. Bansal, "DDR scheme and LSTM RNN algorithm for building an efficient IDS," 2018.
- [31] T. Chen and T. He, "xgboost: eXtreme gradient boosting," 2023. [Online]. Available: <https://cran.r-project.org/web/packages/xgboost/vignettes/xgboost.pdf>
- [32] A. Boukhamla and J. C. Gavro, "CICIDS2017 dataset: performance improvements and validation as a robust intrusion detection system testbed," *International Journal of Information and Computer Security*, vol. 16, no. 1/2, pp. 20–32, 2021, doi: 10.1504/IJICS.2021.117392.
- [33] S. Ustebay, Z. Turgut, and M. A. Aydin, "Intrusion detection system with recursive feature elimination by using random forest and deep learning classifier," in *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*, Dec. 2018, pp. 71–76. doi: 10.1109/IBIGDELFT.2018.8625318.
- [34] J. Hou, P. Fu, Z. Cao, and A. Xu, "Machine learning based DDoS detection through NetFlow analysis," in *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, Oct. 2018, pp. 1–6. doi: 10.1109/MILCOM.2018.8599738.
- [35] A. Bansal and S. Kaur, "Extreme gradient boosting based tuning for classification in intrusion detection systems," in *Advances in Computing and Data Sciences*, 2018, pp. 372–380. doi: 10.1007/978-981-13-1810-8_37.
- [36] M. Alrowaily, F. Alenezi, and Z. Lu, "Effectiveness of machine learning based intrusion detection systems," in *Security, Privacy, and Anonymity in Computation, Communication, and Storage*, 2019, pp. 277–288. doi: 10.1007/978-3-030-24907-6_21.
- [37] N. Thapa, Z. Liu, D. B. KC, B. Gokaraju, and K. Roy, "Comparison of machine learning and deep learning models for network intrusion detection systems," *Future Internet*, vol. 12, no. 10, p. 167, Sep. 2020, doi: 10.3390/fi12100167.
- [38] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa, and C. F. M. Foozy, "Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset," *IEEE Access*, vol. 9, pp. 22351–22370, 2021, doi: 10.1109/ACCESS.2021.3056614.
- [39] N. Meemongkolkiat and V. Suttichaya, "Analysis on network traffic features for designing machine learning based IDS," *Journal of Physics: Conference Series*, vol. 1993, no. 1, p. 012029, Aug. 2021, doi: 10.1088/1742-6596/1993/1/012029.
- [40] M. G. da S. Neto and D. G. Gomes, "Network intrusion detection systems design: a machine learning approach," in *Anais do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2019)*, May 2019, pp. 932–945. doi: 10.5753/sbrc.2019.7413.
- [41] N. Elmrbait, F. Zhou, F. Li, and H. Zhou, "Evaluation of machine learning algorithms for anomaly detection," in *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, Jun. 2020, pp. 1–8. doi: 10.1109/CyberSecurity49315.2020.9138871.
- [42] M. N. Goryunov, A. G. Matskevich, and D. A. Rybolovlev, "Synthesis of a machine learning model for detecting computer attacks based on the CICIDS2017 dataset," *Proceedings of the Institute for System Programming of the RAS*, vol. 32, no. 5, pp. 81–94, 2020, doi: 10.15514/ISPRAS-2020-32(5)-6.
- [43] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers & Security*, vol. 86, pp. 147–167, Sep. 2019, doi: 10.1016/j.cose.2019.06.005.
- [44] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Jul. 2009, pp. 1–6. doi: 10.1109/CISDA.2009.5356528.
- [45] A. Gharib, I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "An evaluation framework for intrusion detection dataset," in *2016 International Conference on Information Science and Security (ICISS)*, Dec. 2016, pp. 1–6. doi: 10.1109/ICISSEC.2016.7885840.
- [46] L. Rokach and O. Maimon, "Top-down induction of decision trees classifiers—a survey," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 35, no. 4, pp. 476–487, Nov. 2005, doi: 10.1109/TSMCC.2004.843247.
- [47] L. Breiman, "Random forests," *Machine Learning*, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
- [48] J. S. Cramer, "The origins of logistic regression," *SSRN Electronic Journal*, 2003, doi: 10.2139/ssrn.360300.
- [49] K. P. Murphy, "Naive bayes classifiers," 2006. [Online]. Available: <https://www.ic.unicamp.br/~rocha/teaching/2011sl1/mc906/aulas/naive-bayes.pdf>
- [50] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, Aug. 1997, doi: 10.1006/jcss.1997.1504.
- [51] G. Hackeling, *Mastering machine learning with scikit-learn*, Second edi. 2014.
- [52] I. Idrissi, M. Azizi, and O. Moussaoui, "An unsupervised generative adversarial network based-host intrusion detection system for internet of things devices," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 2, pp. 1140–1150, Feb. 2022, doi: 10.11591/ijeecs.v25.i2.pp1140-1150.
- [53] T. Ghrib, M. Benmohammed, and P. S. Pandey, "Automated diagnosis of attacks in internet of things using machine learning and frequency distribution techniques," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 2, pp. 950–961, Apr. 2021, doi: 10.11591/eei.v10i2.2766.
- [54] M. Aldwairi and L. Tawalbeh, "Security techniques for intelligent spam sensing and anomaly detection in online social platforms,"

- International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, pp. 275–287, Feb. 2020, doi: 10.11591/ijece.v10i1.pp275-287.
- [55] M. A. Al-Shareeda, S. Manickam, and M. A. Saare, “DDoS attacks detection using machine learning and deep learning techniques: analysis and comparison,” *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 2, pp. 930–939, Apr. 2023, doi: 10.11591/eei.v12i2.4466.
- [56] H. Kamel and M. Z. Abdullah, “Distributed denial of service attacks detection for software defined networks based on evolutionary decision tree model,” *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 4, pp. 2322–2330, Aug. 2022, doi: 10.11591/eei.v11i4.3835.
- [57] M. I. Kareem and M. N. Jasim, “Fast and accurate classifying model for denial-of-service attacks by using machine learning,” *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 3, pp. 1742–1751, Jun. 2022, doi: 10.11591/eei.v11i3.3688.
- [58] A. H. B. Alghuraibawi, R. Abdullah, S. Manickam, and Z. A. A. Alyasseri, “Detection of ICMPv6-based DDoS attacks using anomaly based intrusion detection system: A comprehensive review,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 6, pp. 5216–5228, Dec. 2021, doi: 10.11591/ijece.v11i6.pp5216-5228.
- [59] A. A. Ojugo and R. E. Yoro, “Forging a deep learning neural network intrusion detection framework to curb the distributed denial of service attack,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 2, pp. 1498–1509, Apr. 2021, doi: 10.11591/ijece.v11i2.pp1498-1509.
- [60] N. M. and Y. B. N., “Preemptive modelling towards classifying vulnerability of DDoS attack in SDN environment,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 2, pp. 1599–1611, Apr. 2020, doi: 10.11591/ijece.v10i2.pp1599-1611.
- [61] R. R. Nuiaa, S. Manickam, A. H. Alsaedi, and E. S. Alomari, “A new proactive feature selection model based on the enhanced optimization algorithms to detect DRDoS attacks,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 2, pp. 1869–1880, Apr. 2022, doi: 10.11591/ijece.v12i2.pp1869-1880.
- [62] M. E. Magdy, A. M. Matter, S. Hussin, D. Hassan, and S. A. Elsaid, “Anomaly-based intrusion detection system based on feature selection and majority voting,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 30, no. 3, pp. 1699–1706, Jun. 2023, doi: 10.11591/ijeecs.v30.i3.pp1699-1706.
- [63] S. Chimphee and W. Chimphee, “Machine learning to improve the performance of anomaly-based network intrusion detection in big data,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 30, no. 2, pp. 1106–1119, May 2023, doi: 10.11591/ijeecs.v30.i2.pp1106-1119.
- [64] M. Aljanabi, R. Altaie, S. Talib, A. Hussien Ali, M. A. Mohammed, and T. Sutikno, “Distributed denial of service attack defense system-based auto machine learning algorithm,” *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 1, pp. 544–551, Feb. 2023, doi: 10.11591/eei.v12i1.4537.
- [65] O. Sbai and M. Elboukhari, “Mobile Ad Hoc networks intrusion detection system against packet dropping attacks,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 26, no. 2, pp. 819–825, May 2022, doi: 10.11591/ijeecs.v26.i2.pp819-825.
- [66] T. B. Seong, V. Ponnusamy, N. Z. Jhanjhi, R. Annur, and M. N. Talib, “A comparative analysis on traditional wired datasets and the need for wireless datasets for IoT wireless intrusion detection,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, no. 2, pp. 1165–1176, May 2021, doi: 10.11591/ijeecs.v22.i2.pp1165-1176.
- [67] A. M. Bamhdi, I. Abrar, and F. Masoodi, “An ensemble based approach for effective intrusion detection using majority voting,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 2, p. 664, Apr. 2021, doi: 10.12928/telkomnika.v19i2.18325.
- [68] N. S. Zaini *et al.*, “Phishing detection system using machine learning classifiers,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 17, no. 3, pp. 1165–1171, Mar. 2020, doi: 10.11591/ijeecs.v17.i3.pp1165-1171.
- [69] A. O. Salau, T. A. Assegie, A. T. Akindadelo, and J. N. Eneh, “Evaluation of Bernoulli Naive Bayes model for detection of distributed denial of service attacks,” *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 2, pp. 1203–1208, Apr. 2023, doi: 10.11591/eei.v12i2.4020.
- [70] I. Laassar and M. Y. Hadi, “Intrusion detection systems for internet of thing based big data: a review,” *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol. 12, no. 1, pp. 87–96, Mar. 2023, doi: 10.11591/ijres.v12.i1.pp87-96.
- [71] N. P. Shetty, J. Shetty, R. Narula, and K. Tandona, “Comparison study of machine learning classifiers to detect anomalies,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 5, pp. 5445–5452, Oct. 2020, doi: 10.11591/ijece.v10i5.pp5445-5452.
- [72] S. Rajagopal, P. P. Kundapur, and H. K. Siddaramappa, “A predictive model for network intrusion detection using stacking approach,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 3, pp. 2734–2741, Jun. 2020, doi: 10.11591/ijece.v10i3.pp2734-2741.
- [73] S. Rajagopal, K. Siddaramappa Hareesha, and P. Panduranga Kundapur, “Performance analysis of binary and multiclass models using azure machine learning,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, pp. 978–986, Feb. 2020, doi: 10.11591/ijece.v10i1.pp978-986.
- [74] M. C. Belavagi and B. Muniyal, “Multiple intrusion detection in RPL based networks,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, pp. 467–476, Feb. 2020, doi: 10.11591/ijece.v10i1.pp467-476.
- [75] T. A. J. Ali and M. M. T. Jawhar, “Detecting network attacks model based on a convolutional neural network,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 3, pp. 3072–3078, Jun. 2023, doi: 10.11591/ijece.v13i3.pp3072-3078.
- [76] B. I. Farhan and A. D. Jasim, “Performance analysis of intrusion detection for deep learning model based on CSE-CIC-IDS2018 dataset,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 26, no. 2, pp. 1165–1172, May 2022, doi: 10.11591/ijeecs.v26.i2.pp1165-1172.
- [77] R. I. Farhan, A. T. Maalood, and N. F. Hassan, “Performance analysis of flow-based attacks detection on CSE-CIC-IDS2018 dataset using deep learning,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 3, pp. 1413–1418, Dec. 2020, doi: 10.11591/ijeecs.v20.i3.pp1413-1418.
- [78] S. Laqtib, K. El Yassini, and M. L. Hasnaoui, “A technical review and comparative analysis of machine learning techniques for intrusion detection systems in MANET,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 3, pp. 2701–2709, Jun. 2020, doi: 10.11591/ijece.v10i3.pp2701-2709.
- [79] J. Majidpour and H. Hasanzadeh, “Application of deep learning to enhance the accuracy of intrusion detection in modern computer networks,” *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 3, pp. 1137–1148, Jun. 2020, doi: 10.11591/eei.v9i3.1724.
- [80] A. Boukhalfa, A. Abdellaoui, N. Hmina, and H. Chaoui, “LSTM deep learning method for network intrusion detection system,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 3, pp. 3315–3322, Jun. 2020, doi: 10.11591/ijece.v10i3.pp3315-3322.




- [81] M. Ramasamy and P. V. Eric, "A tree growth based forward feature selection algorithm for intrusion detection system on convolutional neural network," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 1, pp. 472–482, Feb. 2023, doi: 10.11591/eei.v12i1.4015.

BIOGRAPHIES OF AUTHORS






Fatimetou Abdou Vadhil    received her Ph.D. degree in computer science and information technology from the Faculty of Sciences and Techniques, University of Nouakchott, Nouakchott, Mauritania in 2021. She is currently a researcher at the research unit: scientific computing, computer science and data science. Her research area of interest includes the use of artificial intelligence for the improvement of security systems. She can be contacted at email: fatiab38@gmail.com.



Mohamed Lemine Salihi    is a teacher researcher in the Mathematics and Computer Science department at the Faculty of Sciences and Techniques, University of Nouakchott, Nouakchott, Mauritania. His area of research of interest includes scientific computing, artificial intelligence, network security, and data science. He can be contacted at email: mlsalihi@gmail.com.



Mohamedade Farouk Nanne    is a teacher researcher in the Mathematics and Computer Science department at the Faculty of Sciences and Techniques, University of Nouakchott, Nouakchott, Mauritania. His area of research of interest includes artificial intelligence, conversational agents, network security, bioinformatics, and internet of things. He can be contacted at email: mohamedade@gmail.com.