Indonesian Journal of Electrical Engineering and Computer Science
Vol. 34, No. 3, June 2024, pp. 1868~1878
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v34.i3.pp1868-1878

Distributed denial of service attacks classification system using features selection and ensemble techniques

Leila Bagdadi, Belhadri Messabih
Department of Computer Science, University Mohamed Boudiaf, Oran, Algeria

Article Info

Article history:

Received Nov 30, 2023
Revised Feb 20, 2024
Accepted Mar 3, 2024

Keywords:

CICDDoS-2019
DDOS attacks
Ensembles methods
Features selection
Machine learning

ABSTRACT

Distributed denial-of-service (DDoS) attacks are expanding threat to online services and websites. These attacks overwhelm targets with traffic from multiple sources to exhaust resources and make services unavailable. The frequency of DDoS attacks exhibits an ongoing upward trajectory over time. This persistent escalation highlights the need for effective countermeasures. While machine learning approaches have been extensively investigated for binary classification of DDoS attacks, multi-class classification has received comparatively less examination in the literature despite its greater practical utility. In this paper, we propose an intrusion detection system for detecting and classifying DDoS attacks, based on two main axes: feature selection for selecting the best relevant features and ensemble learning technique for improving performance by combining weak learners. The proposed model has been trained and evaluated on the CICDDoS2019 dataset. Experimental evaluation demonstrates improved performance using a subset of 16 relevant features identified, with a test accuracy of 82.35% attained for discriminating between the 12 classes represented in the dataset. By aggregating attacks sharing common characteristics resulting in 7 classes, the approach achieves surpassing 97% accuracy. Additionally, a binary classification delineating benign and DDoS attacks attain 99.90% accuracy.

Corresponding Author:

Leila Bagdadi
Department of Computer Science, University Mohamed Boudiaf
BP 1505, El M'naouer, Oran-31000, Algeria
Email: leila.bagdadi@univ-usto.dz

1. INTRODUCTION

Distributed denial-of-service (DDoS) is one of the most hazardous internet attacks that aims to exhaust the resources of a targeted system, network, or application, making it unavailable to legitimate users by flooding it with flows from varied sources. In a 2020 analysis by Cisco, it was forecasted that the quantity of DDoS attacks would increase twofold, rising from 7.9 million in 2018 to 15.4 million by 2023. In light of the analysis, it has become crucial to seriously consider counterattacking this type of attack. The attack is carried out by sending a great number of requests to the target server, thus causing it to crash or become unresponsive. Any website or online service that relies on internet connectivity can be vulnerable to such attacks. Various protocols such as hypertext transfer protocol (HTTP), user datagram protocol (UDP), transmission control protocol (TCP) and internet control message protocol (ICMP) can be employed for these attacks in the application layers and transport [1]. In the network security research area, considerable research efforts have been dedicated to developing effective countermeasures against DDoS attacks.

Sharafaldin *et al.* [2] analyzed eleven existing intrusion detection systems (IDS) datasets, finding most fail real-world criteria due to limited traffic diversity, outdated attacks covered, and heavy anonymization. This motivated developing a new benchmark with contemporary attacks. The highest accuracy obtained by their proposed model was given by random forest (RF) algorithm. In later work [1], the authors presented another modern benchmark dataset for anomaly detection captured from a real-world testbed. Evaluating multiple machine learning techniques evidenced superior performance by the ID3 algorithm compared to RF when leveraging this improved dataset. Collectively, these studies highlight significant constraints with current IDS benchmarks and demonstrate the efficacy of updated data coupled with machine learning for intrusion detection. Abdulraheem and Ibraheem [3] proposed a multi-stage process using the CICIDS2017 dataset, first addressing limitations and then applying feature standardization and selection to identify the most effective 36 features. They trained an multi-layer perceptron (MLP) model on this improved dataset and compared results to those obtained by Sharafaldin *et al.* [2], demonstrating performance gains from their data preprocessing approach. Alamri and Thayananthan proposed a two-stage DDoS mitigation scheme for software-defined networking using adaptive bandwidth thresholding and extreme gradient boosting classification, evaluated on CICDDoS2019, NSL-KDD and CAIDA datasets [4].

Usha *et al.* [5] proposed a system to efficiently detect and classify DDoS attacks, using the CICDDoS2019 dataset with various machine learning (ML) algorithms including XGBoost and a convolutional neural network (CNN) architecture. Experiments showed that XGBoost achieved the greatest accuracy for attack identification, while CNN and k-nearest neighbour (KNN) performed comparably for classification. The study demonstrates that machine learning techniques can provide efficient DDoS attack detection when applied to recent benchmark datasets. Araujo *et al.* [6] explored using feature selection to improve DDoS attack classification with XGBoost, comparing filter methods like variance filtering and wrapper techniques. Experiments on the CICDDoS2019 benchmark showed analysis of variance (ANOVA) provided the best performance enhancements. The study demonstrates feature selection, specifically ANOVA-based filtering, significantly boosts machine learning for automated DDoS attack detection. Thorat *et al.* [7] introduced TaxoDaCML, a multi-class approach designed for detecting DDoS attacks. The classification problem was decomposed into seven smaller sub-classification tasks, and the feature set was refined through the application of ANOVA and MI feature selection techniques as outlined in [7]. They classified the 11 DDoS attacks present in the CIC-DDOS2019 dataset using a computationally light decision trees algorithm [7].

Chartuni and Marquez [8] proposed a 7-layer neural network model for multiclass DDoS attack classification, using the CIC-DDOS2019 dataset for training after preprocessing. By creating 10 new balanced classes retaining the initial attacks with 78 attributes, throughout three different scenarios, they have successfully classified 13 attacks for the first scenario. They removed the three attacks namely Portmap, UDP, and lightweight directory access protocol (LDAP) to ameliorate the obtained results in the second scenario. At the least, by grouping attacks sharing similarities, they obtained a better result. Lai and Nguyen [9] proposed a hybrid machine learning approach combining hierarchical temporal memory and k-nearest neighbors for DDoS attack detection and classification. The model that they proposed demonstrates incremental learning capabilities and it was evaluated on the CICDDoS 2019 benchmark, achieving high attack identification performance [9].

While machine learning approaches for binary classification of DDoS attacks have been extensively studied, there is relatively limited research on multi-class classification, despite it is practical relevance. To address this gap, we propose a novel approach that emphasizes the importance of feature selection and ensemble methods in improving the detection and classification of DDoS attacks using the most realistic public CICDDoS2019 dataset. To the extent of our knowledge, this study is the first to explore this approach.

In this work, the principal contribution is a novel approach combining the efficacy of ensemble methods with a majority voting feature selection technique to improve multi-class classification accuracy for DDoS attacks. This is achieved using the most recent CICDDoS2019 dataset. Additionally, by conducting experiments to group similar attacks based on common underlying tactics into unified sub-categories, misclassification errors are reduced, further enhancing model accuracy.

The remainder of this paper is structured as follows. Section 2 gives the theoretical basis underpinning the study. Section 3 delineates the proposed approach in a comprehensive manner, emphasizing the various steps undertaken to derive the final model. Section 4 delineates the diverse metrics utilized to assess model performance, with experimental results and attendant discussions being presented therein. Finally, section 5 furnishes conclusions stemming from the study and outlines potentially fruitful future research directions.

2. THEORICAL BASIS

In the network security research area, various IDS have been proposed for detecting DDoS attacks, such as anomaly-based, signature-based and hybrid IDS [10], [11]. Machine learning has surfaced as a potentially advantageous technique in anomaly detection of DDoS attacks. Most popular individual machine learning algorithms have been widely utilized for detecting and classifying DDoS attacks, however single model may not always yield optimal predictions [12]-[14]. These algorithms are susceptible to errors caused by variance and bias. To overcome their limitations, ensemble methods can be useful. The concept behind ensemble learning is that by aggregating the predictions of several weak models, we can mitigate variance and/or bias [12], ultimately achieving better results compared to individual models [15]. Furthermore, feature selection is a critical process in machine learning, involving the identification of the most relevant features in a dataset. The substantial number of input features in the most recent datasets can readily prompt the overfitting of machine learning models. To mitigate this effect, feature selection techniques must be applied to reduce the dimensionality of the input space. The primary objectives of feature selection are to identify a parsimonious subset of discriminative input features [16] while discarding redundant and correlated variables, this plays an essential role in improving prediction accuracy, reducing computational complexity, and decreasing prediction latency [17].

2.1. Features selection

Feature selection algorithms can be categorized into three main groups: filter, wrapper, and embedded methods [16]. Filter methods rank features based on statistical measures such as ANOVA, mutual information and correlation [18]. ANOVA is a well-established statistical approach for analyzing differences between multiple independent group means [19]. ANOVA ranks features [19]. A higher F-ratio signifies increased association between the feature and grouping factor, thereby indicating greater relevance of that feature in predicting the categorical target label. Mutual Information quantifies the amount of information shared between one random variable and another variable [20], and only features with higher mutual information values are considered more relevant and are thus retained [21].

Correlation analysis is a technique for pruning redundant predictor variables within multivariate data by quantifying the extent of their interdependence [22]. Mathematically, this involves the calculation of correlation coefficients assessing the association between two variables, with Pearson’s coefficient r adopted as predominant measure [23]. Conceptually, r is the quotient of the covariance of two variables and the product of their standard deviations when mathematically expressed [22]. Its bounded range of [-1, 1] lends interpretability, with proximity to +1/-1 indicating a strong relationship, while trending toward zero denotes independence [24].

Unlike filter methods, wrapper methods are focus on optimizing the performance of specific machine learning model by selecting a subset of features [25]. The most greedy methods frequently employed are Boruta, recursive feature elimination (RFE), Forward selection and Backward elimination. Boruta is an extension of the RF classifier that identify all important features with the target variable [26], [27]. The algorithm generates shuffled versions of all features, known as Shadow Features, trains a forest classifier on the expanded dataset and the significance of each feature is then evaluated using metrics such as mean decrease accuracy [28]. The algorithm stops when either all features are accepted or rejected or when it attains a number of specified iterations [27].

As described in [29], RFE is a method that recursively eliminating the least significant features from a dataset. Initially, a model is constructed using all the features and importance scores are assigned to each feature [29]. This procedure is repeated until the wanted number of features is attained [29], [30]. Forward feature selection (FFS) iteratively adds features one at a time, at each step adding the feature that decreases the model error the most [31]. This is repeated until addition of remaining features does not sufficiently improve performance.

Backward feature selection (BFS): in contrast, backward selection starts with the all features, iteratively pruning the least useful feature at each step [31]. Importance is determined by the increase in model error after excluding a feature [31]. The process is repeated until no inclusion or exclusion of features significantly improves model performance.

Embedded methods incorporate feature selection within the machine learning algorithm [32]. In the training phase, the classifier adjusts its internal parameters and assigns suitable importance to each feature to enhance classification accuracy [32]. Examples of embedded techniques encompass decision tree-oriented algorithms like RF and extreme gradient boosting [32]. They provide feature importance as part of their output [33]. The importance is calculated based on how much each feature reduces the impurity in the tree [34]. The measure of impurity is either the Gini impurity or the information Gain/entropy [34].

2.2. Ensemble methods

There are three popular classes of ensemble learning methods namely boosting, bagging, and stacking [35], [36]. Bagging, also known as bootstrap aggregation, is a technique in which multiple base models are trained on equal-sized subsets independently and in parallel, then results obtained from the different models are then combined by averaging or voting to get a final prediction [36]. Contrary to the bagging algorithms which are parallel trained, the boosting technique, mostly homogeneous, trains a sequence of models on a weighted training set. Boosting works by sequentially adding models to the ensemble, each new model corrects the errors made by the previous models [37] until the performance is satisfactory or other stopping conditions are met. In contrast, stacking is an ensemble learning framework where a distinct machine learning algorithm is trained to merge the predictions of multiple ensemble members [35]. The result of the individual predictions is then treated as the next training data, which serves as input for another model called meta learner.

RFs consist of a collection of decision trees where each tree is trained on the value of the subset created from a random resampling of the training dataset [38], [39]. For the final prediction, each individual tree casts a vote for a specific class, and the forest predicts the class that achieves the majority of votes [39]. The extra trees algorithm operates akin to the RF technique. A fundamental distinction between these methods lies in how extra trees randomly selects splitting points, as opposed to RFs selection of the best splitting point [35]. This randomization makes extra trees faster to train than RF. Adaptive boosting (AdaBoost) algorithm combines weak learners to create a strong one [40]. It operates based on a weighting principle [32], where subsequent models in the ensemble sequence focus their training on instances misclassified by prior models in order to correct those errors.

XGBoost is a decision tree-oriented method using regularization techniques to minimize overfitting [6]. It employs a gradient-boosting algorithm to construct a collection of weak models. At each iteration, a new weak model is incorporated into the ensemble, and the current ensemble is updated [41]. The algorithm operates in a stage-by-stage approach, adding and updating models in a systematic order where each step follows the preceding one.

LightGBM stands for light gradient boosting, utilizes decision trees and is engineered to be effective and scalable for large datasets. Unlike other tree-based models, LightGBM constructs decision trees by growing the leaves first, rather than growing the levels from the root, which accelerates the training process. It uses gradient-based one-side sampling and exclusive feature bundling to separate out the data instances for finding optimal split points and deal with excessive features [42], [43].

CatBoost stands for categorical boosting, is a gradient-boosting algorithm that iteratively adds decision trees to the ensemble to build a strong predictive model. Like XGBoost, catBoost uses regularization techniques to minimize overfitting. Although CatBoost is used for categorical features, it works seamlessly with numerical features and can be a good choice for building models using heterogeneous data [44]. Histogram gradient boosting machine (HGBM) is a similar to other gradient boosting algorithms. However, instead of working with the original data, HGBM first constructs feature histogram representation of the features optimizing thus the training process and reducing computaional complexity [42], [45].

3. METHOD

In order to counter DDoS intrusions, the proposed approach, as delineated in Figure 1, comprises three primary stages: data preprocessing, feature selection to identify the most relevant explanatory variables, and application of ensemble machine learning techniques to generate the model. Experiments were conducted using the CIC-DDoS2019 dataset publicly accessible [46]. This benchmark dataset was generated by the Canadian Institute for Cybersecurity (CIC) [1] utilizing the CICFlowMeter software [2]. It encompasses over 50 million samples, comprising benign traffic as well as 12 recent common DDoS attack types enumerated in Table 1. These attacks include LDAP, NetBIOS, MSSQL, UDP, UDP-Lag, WebDDoS, Syn, SSDP, SNMP, DNS, NTP, and TFTP floods. Each flow in CIC-DDoS2019 is characterized by approximately 87 features with ground truth labels provided for method evaluation.

3.1. Data preprocessing

As preprocessing is an essential step and we had prior knowledge that socket-related features including source and destination IP, source and destination port, and protocol can vary from network to network, we started by removing these ones [46]. Features that don’t contain useful values for model training such as Timestamp, Inbound, Unnamed, Fwd Header Length.1, and SimilarHTTP are also removed. The inclusion sof those features can easily lead to shortcut learning issues [46] and affect clearly the performance of the models. So, the feature number was reduced from 87 to 77. We also dropped all rows with missing and infinite values. After conducting a thorough analysis of the dataset, which revealed a significant number of redundant records, we were prompted to perform a treatment to obtain a non-redundant dataset. The results presented in Table 1 demonstrate a clear reduction in the number of benign and attack instances following the data cleaning process.

Given the negligible number of WebDDoS attacks, we have removed the corresponding class from the prepared dataset, then we adjusted the different classes to match the minority class, which is NetBios with 17,925 samples. This adjustment resulted in a dataset of 17,000 samples per attack, effectively mitigating the impact of class imbalance on our modeling process. We complete this step by normalizing the data using the

Min-Max scaler technique to scale all feature values within the [0, 1] interval ensuring equal importance for each feature.

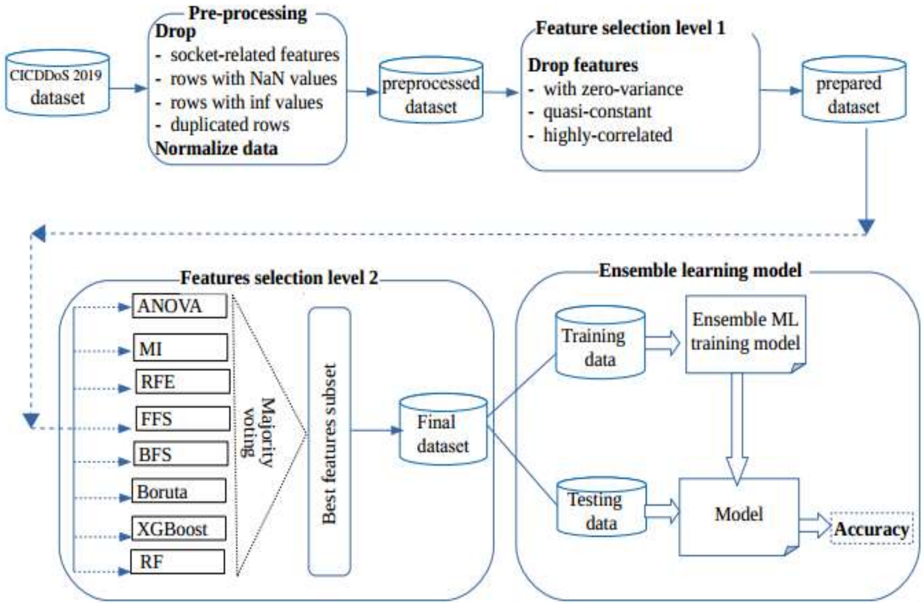


Figure 1. The proposed approach framework

Table 1. CIC-DDoS2019 Before/ After pre-processing

Attacks	Before cleaning		After cleaning	
	Attacks number	Benign	Attacks number	Benign
LDAP	2,179,930	1,612	28,871	1,393
NetBIOS	4,093,279	1,707	17,925	1,627
MSSQL	4,522,492	2,006	193,346	1,871
UDP	3,134,645	2,157	1,074,465	2,042
UDP-Lag/WebDDoS	366,461/439	3,705	88,986/ 414	3,542
SYN	1,582,289	392	155,501	374
SSDP	2,610,611	763	890,292	732
SNMP	5,159,870	1,507	112,066	1,302
DNS	5,071,011	3,402	108,119	3,035
NTP	1,202,642	14,365	1,112,756	13,309
TFTP	20,082,580	25,247	5,549,475	23,248
BENIGN		56,863		52,475

3.2. Best feature subset

The resultant dataset from the preceeding step contains a substantial number of features, necessitating selection of the best subset retaining only the most relevant features. The proposed methodology incorporates two pivotal stages of feature selection: Level 1 employs supervised techniques, while Level 2 leverages unsupervised ones. According to [47], if the target variable is not taken into account during the elimination of predictors, the method is classified as unsupervised; otherwise, it is categorized as supervised. The unsupervised feature selection phase focuses on eliminating non-informative and redundant predictors. We initiate this by discarding all features exhibiting zero variance, which provide no discriminative information for target prediction. Additionally, quasi-constant features taking identical values for over 99% of observations are removed. Successively, highly correlated features with absolute Pearson's coefficient correlation exceeding 0.98 are discarded. Through this streamlined process, the number of features is appreciably reduced from 76 to 42 as shown in Table 2. Subsequent to the initial unsupervised feature selection phase, implementation of all the aforementioned feature selection methodologies was undertaken. Each technique yielded a subset of highest-ranked attributes. To further augment the selection process, we leveraged majority voting to consolidate the individual subsets into a unified set. Beginning with the features favored by all methods, features were incrementally incorporated into the model one at a time on the basis of vote score.

At each iteration, the performance enhancement resulting from supplemented features was quantified. This iterative inclusion procedure persisted until reaching a threshold where augmenting the remaining features failed to considerably improve model accuracy. The predetermined models delineated in section 2 validated that the subset consisting of the 16 top-ranked features from the voting list, specifically "Flow_Duration, Flow_ IAT_Min, Flow_IAT_Max, Flow_IAT_Mean, Flow_Packets/s, Total_Fwd_Packets, Total_Length_of_Fwd_Packets, Fwd_Header_Length, Fwd_Packets_Length_Min, Fwd_Packets_Length_Max, Total_Length_of_Bwd_Packets, Bwd_Packets/s, Init_Win_bytes_forward, Init_Win_bytes_backward, min_seg_size_forward, and Max_Packets_Length", yielded optimal performance. These attributes were identified as impactful by at least six of the explored feature selection techniques.

Table 2. Removed features after an unsupervised features selection

Features	Dropped features
With zero variance	Bwd_PSH_Flags, Fwd_URG_Flags, Bwd_URG_Flags, FIN_Flag _Count, PSH_Flag_Count, ECE_Flag_Count, Fwd_Avg_Bytes/Bulk, Fwd_Avg_Packets/Bulk, Fwd_Avg_Bulk_Rate, Bwd_Avg_Bytes/Bulk, Bwd_Avg_Packets/Bulk, Bwd_Avg_Bulk_Rates
Quasi constant	Fwd_PSH_Flags, SYN_Flag_Count, RST_Flag_Count
Highly correlated	Average_Packet_Size, Avg_Bwd_Segment_Size, Avg_Fwd_Segment_Size, Bwd_IAT_Std, Flow_IAT_Std, Fwd_IAT_Max_Fwd_IAT_Mean, Fwd_IAT_Min, Fwd_Packet_Length_Mean, Idle_Max, Min_Packet_Length, Packet_Length_Mean, Subflow_Bwd_Bytes, Subflow_Bwd_Packets, Subflow_Fwd_Bytes, act_data_pkt_fwd, Fwd_IAT_Total, Fwd_Packets/s, Subflow_Fwd_Packets

4. RESULTS AND DISCUSSION

4.1. Performance metrics

The selection of appropriate evaluation metrics is crucial for quantitatively assessing and comparing the performance of machine learning classifier. In this work, we utilize a suite of widely adopted metrics (accuracy, precision, recall and F1-score) that offer complementary insights into model behavior. These metrics are evaluated form the counts of true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN).

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

$$F1 - score = \frac{2*Precision*Recall}{Precision+Recall} \tag{4}$$

4.2. Experimental results

Throughout the experimentation and performance evaluation process, we used several python libraries particularly numpy, matplotlib, scikit-learn and pandas. In order to rigorously assess the performance of our proposed approach, 5-fold cross-validation methodology was instantiated across the entire dataset. This validation strategy separates the dataset into 5 mutually exclusive partitions or "folds". A model training iteration is then conducted on each unique combination of 4 folds, with model assessment carried out on the remaining single holdout fold not utilized during the training phase. Repeating this approach permits approximating model generalization aptitude by surveying efficacy over multiple distinct train-test splits of the data.

Inspired by the well-established efficacy of ensemble learning approaches for enhancing model performance [12], we implemented the previously outlined ensemble models for both binary and multiclass classification tasks using the best feature subset identified during prior feature selection. Furthermore, to highlight that this feature subset encapsulates the core information for differentiating DDoS attacks, we trained these models on the preprocessed dataset (77 features). As shown in Tables 3 and 4, the ensemble models trained on the best feature subset demonstrated superior classification accuracy compared to models using the full preprocessed dataset. The consistency of the results provides compelling evidence for the utility of conducting rigorous feature selection before model training. For the binary classification task, we merged the various DDoS attack classes into a unified attack class, which was then classified against the benign traffic class. As shown in Table 5, the binary classification outcomes indicate the LightGBM model attained the maximum observed performance at 99.90%, followed by XGBoost and CatBoost yielding 99.86% and 99.83% accuracy, respectively.

Table 3. Classification results for 12 classes with 77 features

Classifier	RF	Extra Tree	Light GBM	CatBoost	HGBC	AdaBoost	XGBoost
Accuracy (%)	78.36	77.57	81.03	81.09	80.43	76.91	82.24

Table 4. Classification results for 12 classes with 17 features

Classifier	RF	Extra Tree	Light GBM	CatBoost	HGBC	AdaBoost	XGBoost
Accuracy (%)	**78.50**	**77.70**	**81.59**	**81.16**	**80.65**	76.92	**82.35**

Table 5. Binary classification using 17 features

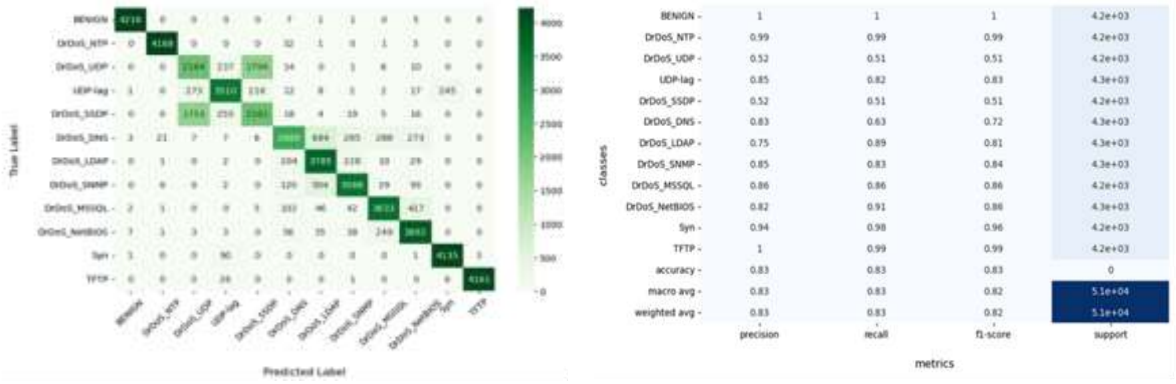
Classifier	RF	Extra Tree	Light GBM	CatBoost	HGBC	AdaBoost	XGBoost
Accuracy (%)	99.82	99.80	**99.90**	99.83	99.86	99.80	99.86

Additionally, comparisons were undertaken between the ensemble methodologies and five single model classifiers: decision tree (DT), multinomial Naive Bayes (MNNB), stochastic gradient descent (SGD), support vector machine (SVM), and logistic regression (LR) to substantiate the superior capabilities of ensemble paradigms. The accuracy attained by these models is delineated in Table 6, with the DT approach exhibiting the highest performance at 76.96%. However, the multinomial Naive bayes classifier resulted in markedly lower accurate classification of 50.22%. In contrast, almost all implemented ensemble models demonstrated categorically elevated accuracy levels relative to any individual model. The maximum accuracy registered across the individual models as consistently exceeded across the ensemble models, clearly evidencing enhanced generalization aptitude.

Table 6. Individual models classification results for 12 classes with 17 features

Classifier	DT	MNNB	SGD	SVM	LR
Accuracy (%)	**76.96**	50.22	60.88	61.19	60.89

The multiclass classification outcomes indicate the XGBoost as shown in Figure 2 model attained the maximum observed performance at 82.35%, followed by LightGBM and CatBoost yielding 81.59% and 81.16% accuracy, respectively. For deeper examination of the classification efficacy across individual classes, the confusion matrix produced via the XGBoost classifier is outlined in Figure 2(a). Additionally, the class-wise evaluation metrics is summarized through the classification report depicted in Figure 2(b).



classes	precision	recall	f1-score	support
BENIGN	1	1	1	4.2e+03
DrDoS_NTP	0.99	0.99	0.99	4.2e+03
DrDoS_UDP	0.52	0.51	0.51	4.2e+03
UDP-lag	0.85	0.82	0.83	4.3e+03
DrDoS_SSDP	0.52	0.51	0.51	4.2e+03
DrDoS_DNS	0.83	0.63	0.72	4.3e+03
DrDoS_LDAP	0.75	0.89	0.81	4.3e+03
DrDoS_SNMP	0.85	0.83	0.84	4.3e+03
DrDoS_MSSQL	0.86	0.86	0.86	4.2e+03
DrDoS_NetBIOS	0.82	0.91	0.86	4.3e+03
Syn	0.94	0.98	0.96	4.2e+03
TFTP	1	0.99	0.99	4.2e+03
accuracy	0.83	0.83	0.83	0
macro avg	0.83	0.83	0.82	5.1e+04
weighted avg	0.83	0.83	0.82	5.1e+04

(a) (b)

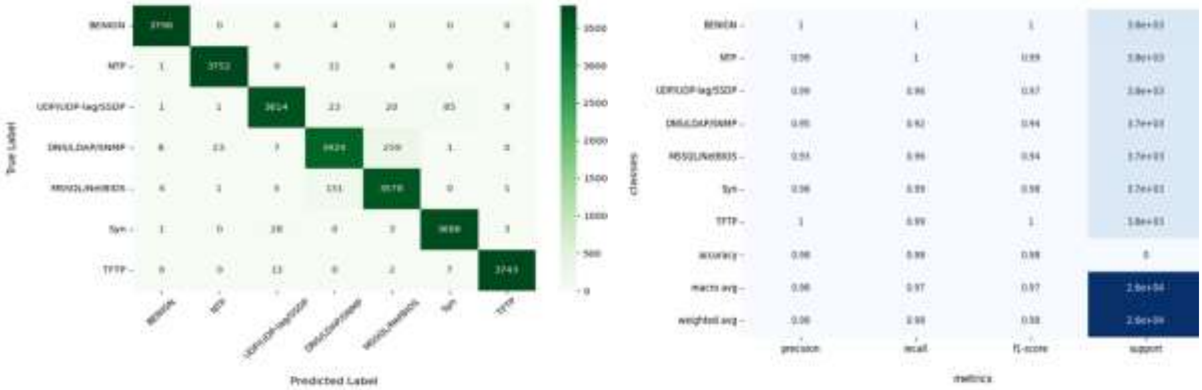
Figure 2. The XGBoost (a) confusion matrix and (b) classification report for 12 classes

Careful examination of the confusion matrix outcomes exposes challenges in discriminating between particular attack types exhibiting analogous traffic characteristics. This implies substantial overlap in the underlying features across certain classes, intrinsically complicating efforts to reliably distinguish between them, despite inclusion of only the best subset of relevant features. As noted in prior scholarship [8], domain name system (DNS), lightweight directory access protocol (LDAP), and simple network management protocol (SNMP) attacks demonstrate considerable commonalities in their exploitation of amplification and reflection to overwhelm victims. Hence, amalgamating them into a consolidated DDoS attack class appears judicious. Furthermore, UDP, UDP-LAG, and SSDP attacks uniformly leverage user datagram protocol (UDP) [9] as the foundational transport mechanism, primarily effectuating amplification attacks. Therefore, merging them into a unified category for modeling purposes is reasonable. While existing literature characterizes MSSQL attacks as TCP-based reflection attacks [1], inspection of traffic traces reveals UDP predominance in MSSQL attack packets [7], given the protocol's intended functionality for handling UDP query requests.

Hence, categorizing MSSQL attacks under UDP-reflection attacks more accurately aligns with standard server communication conventions. Additionally, notable parallels exist between MSSQL and NetBIOS attacks regarding exploitation of service vulnerabilities [31]. Both offensives target specific network services, with MSSQL concentrating on Microsoft SQL Server, while NetBIOS attacks compromise the NetBIOS Windows file/printer sharing protocol [32]. Attackers frequently manipulate authentication protocols, endeavoring unauthorized database access with MSSQL, and bypassing authentication to access shared NetBIOS resources, enabling potential data exfiltration. In light of the identified similarities, consolidating MSSQL and NetBIOS attacks into a unified class is warranted. Based on the discerned commonalities between particular DDoS attack types, we consolidated these similar classes into 7 unified categories. The multiclass classification after implementation of this consolidated categorization succeeds in reducing ambiguities between similar attack types, as shown in Table 7. The results reveal that the XGBoost model achieved the highest observed performance at 97.48%, followed by LightGBM and CatBoost with accuracies of 97.23% and 97.10% respectively. In Figure 3 explains about XGBoost. The confusion matrix generated by the XGBoost classifier is presented in Figure 3(a), while the class-wise evaluation metrics is summarized in the classification report depicted in Figure 3(b).

Table 7. Classification results after grouping into 7 sub-classes

Classifier	RF	Extra tree	Light GBM	CatBoost	HGBC	AdaBoost	XGBoost
Accuracy (%)	97.14	96.60	97.23	97.10	97.04	96.20	**97.48**



(a)

(b)

Figure 3. The XGBoost (a) confusion matrix (b) and classification report for the 7 sub-catgories

D’Hooge *et al.* [47], demonstrated the significant influence of metadata features, also referred to as socket-related features, on the model’s prediction. They highlighted the prevalence of such features in many published articles [48]. In particular, destination port acts as a prime target for shortcut learning [47]. Taking this remark into account, in our comparative table as illustrated by Table 8, we used an additional column (metadata features) to indicate the use or not of metadata. In terms of performance, the accuracy of our models exceeds by more than 9% the accuracy of models not using any metadata. Specifically, ou accuracy surpasses those models obtained by [1], [6], which did not use metadata. Furthermore, by categorizing the attacks exhibiting commonalities, as described previously, into 7 distinct subgroups, our model accomplished heightened performance with classification accuracy exceeding 97% across the aggregated evaluation set.

Table 8. Comparison of the proposed approach with previous related works using CICDDoS2019

References	Algorithm	F1-score			Metadata features used
		Binary	Multiclass 12 classes	Multiclass 7 classes	
Araujo *et al.* [6]	XGBoost	99.79%	73.40%	--	**No**
Sharafaldin *et al.* [1]	ID3	--	70.00%	--	**No**
Alamri and Thayananthan [4]	XGBoost	100%	92.00%	--	Yes
Devi and Singh [49]	J48	99.65%	--	--	Yes
Sbai and Elboukhari [50]	DNN	99.00%	--	--	Yes
Elsayed *et al.* [51]	DL	99.90%	--	--	--
Cil *et al.* [52]	DNN	99.90%	--	--	--
Thorat *et al.* [7]	RF	99.90%	85.80%	--	Yes
Our approach	XGBoost	99.90%	82.35%	97.48%	**No**

4.3. Discussions

Extensive research has been conducted on machine learning methods for the binary classification of DDoS attacks, yet there is a scarcity of studies focusing on multi-class classification, despite it is practical importance. Additionaly, many studies have included at least one metadata feature in their models that can easily lead to shortcut learning issues [46] and affect clearly the performance. In this study, we focused in particular on the use of feature selection techniques and ensemble learning models for multiclass DDoS attack classification by excluding metadata features from the original dataset. The obtained feature subset demonstrated superior classification accuracy compared to models using the original dataset. The consistency of the results provided compelling evidence for the utility of conducting rigorous feature selection before model training. The comparisons that have been taken between the ensemble models and individual classifiers proved the superior capabilities of ensemble paradigms. We found that the feature subset obtained by the feature selection process combined with XGBoost classifier tended to have an accuracy exceeding by more than 9% the accuracy of models not using any metadata. After observing significant similarities among certain types of attacks, we proposed a new categorization consisting of 7 classes. The multiclass classification after implementation of this consolidated categorization succeeds in reducing ambiguities between similar attack types attaining an accuracy exceeding 97% without the use of any metadata feature. These results can contribute upstream to the engineering of a new dataset in future research endeavors.

5. CONCLUSION

The principal objective of this study was to propose a novel intrusion detection system for DDoS attack identification and classification. The proposed approach combines effective feature selection techniques for dimensionality reduction with ensemble learning methods to enhance model accuracy. Evaluation using the contemporary CICDDoS2019 benchmark demonstrates the efficacy of the methodology, attaining 99.90% accuracy for binary classification and over 82% accuracy across 12 classes. By consolidating attacks sharing common tactics into 7 unified categories, classification performance is further improved, achieving over than 97% test accuracy. The proposed system exhibits significant improvements compared to prior academic works examining multi-class DDoS attack classification utilizing the same dataset. Future work may integrate complementary machine learning paradigms like deep neural networks to investigate potential hybrid model benefits. Additionally, validation on live network traffic could assess real-world performance. Overall, this work provides salient contributions surrounding feature selection and ensemble learning for advanced DDoS attack modeling.

REFERENCES

[1] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, “Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy,” in *2019 international carnahan conference on security technology (ICCST)*, Oct. 2019, doi: 10.1109/ccst.2019.8888419.

[2] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in *ICISSP 2018-Proceedings of the 4^{th} International Conference on Information Systems Security and Privacy*, 2018, vol. 2018-Janua, pp. 108–116, doi: 10.5220/0006639801080116.

[3] M. H. Abdulraheem and N. B. Ibraheem, “A detailed analysis of new intrusion detection dataset,” *Journal of Theoretical and Applied Information Technology*, pp. 1992–8645, 2019.

[4] H. A. Alamri and V. Thayananthan, “Bandwidth control mechanism and extreme gradient boosting algorithm for protecting software-defined networks against DDoS attacks,” *IEEE Access*, vol. 8, pp. 194269–194288, 2020, doi: 10.1109/access.2020.3033942.

[5] G. Usha, M. Narang, and A. Kumar, “Detection and classification of distributed DoS attacks using machine learning,” in *Lecture Notes on Data Engineering and Communications Technologies*, Springer Nature Singapore, 2021, pp. 985–1000.

[6] P. Araujo *et al.*, “Impact of feature selection methods on the classification of DDoS attacks using XGBoost,” *Journal of Communication and Information Systems*, vol. 36, no. 1, pp. 200–214, 2021, doi: 10.14209/jcis.2021.22.

[7] O. Thorat, N. Parekh, and R. Mangrulkar, "TaxoDaCML: Taxonomy based divide and conquer using machine learning approach for DDoS attack classification," *International Journal of Information Management Data Insights*, vol. 1, no. 2, Nov. 2021, doi: 10.1016/j.jjimei.2021.100048.

[8] A. Chartuni and J. Márquez, "Multi-classifier of DDoS attacks in computer networks built on neural networks," *Applied Sciences*, vol. 11, no. 22, Nov. 2021, doi: 10.3390/app112210609.

[9] Y.-K. Lai and M.-H. Nguyen, "A real-time DDoS attack detection and classification system using hierarchical temporal memory," *APSIPA Transactions on Signal and Information Processing*, vol. 12, no. 2, 2023, doi: 10.1561/116.00000147.

[10] J. M. Kizza, "System intrusion detection and prevention," in *A Guide to Computer Network Security*, Springer London, pp. 273–298.

[11] M. Najafimehr, S. Zarifzadeh, and S. Mostafavi, "DDoS attacks and machine-learning-based detection methods: A survey and taxonomy," *Engineering Reports*, vol. 5, no. 12, May 2023, doi: 10.1002/eng2.12697.

[12] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," *Frontiers of Computer Science*, vol. 14, no. 2, pp. 241–258, Aug. 2019, doi: 10.1007/s11704-019-8208-z.

[13] A. J. Ferreira and M. A. T. Figueiredo, "Boosting algorithms: A review of methods, theory, and applications," in *Ensemble Machine Learning*, Springer New York, 2012, pp. 35–85.

[14] H. Thanh and T. Lang, "Use the ensemble methods when detecting DoS attacks in Network Intrusion Detection Systems," *EAI Endorsed Transactions on Context-aware Systems and Applications*, vol. 6, no. 19, p. 163484, Nov. 2019, doi: 10.4108/eai.29-11-2019.163484.

[15] N. Najm Abdulla and R. K. Hasoun, "Review of detection denial of service attacks using machine learning through ensemble learning," *Iraqi Journal for Computers and Informatics*, vol. 48, no. 1, pp. 13–20, Jun. 2022, doi: 10.25195/ijci.v48i1.349.

[16] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers and Electrical Engineering*, vol. 40, no. 1, pp. 16–28, Jan. 2014, doi: 10.1016/j.compeleceng.2013.11.024.

[17] J. Zacharias, M. von Zahn, J. Chen, and O. Hinz, "Designing a feature selection method based on explainable artificial intelligence," *Electronic Markets*, vol. 32, no. 4, pp. 2159–2184, Dec. 2022, doi: 10.1007/s12525-022-00608-1.

[18] N. Sánchez-Maroño, A. Alonso-Betanzos, and M. Tombilla-Sanromán, "Filter methods for feature selection-a comparative study," in *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 178–187, 2007.

[19] M. Naveed *et al.*, "A deep learning-based framework for feature extraction and classification of intrusion detection in networks," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–11, Aug. 2022, doi: 10.1155/2022/2215852.

[20] J. R. Vergara and P. A. Estévez, "A review of feature selection methods based on mutual information," *Neural Computing and Applications*, vol. 24, no. 1, pp. 175–186, Mar. 2013, doi: 10.1007/s00521-013-1368-0.

[21] M. Alduailij, Q. W. Khan, M. Tahir, M. Sardaraz, M. Alduailij, and F. Malik, "Machine-learning-based DDoS attack detection using mutual information and random forest feature importance method," *Symmetry*, vol. 14, no. 6, May 2022, doi: 10.3390/sym14061095.

[22] A. Bommert, T. Welchowski, M. Schmid, and J. Rahnenführer, "Benchmark of filter methods for feature selection in high-dimensional gene expression survival data," *Briefings in Bioinformatics*, vol. 23, no. 1, Sep. 2021, doi: 10.1093/bib/bbab354.

[23] O. Solovei, "New organization process of feature selection by filter with correlation-based features selection method," *Innovative Technologies and Scientific Solutions for Industries*, no. 3 (21), pp. 39–50, Nov. 2022, doi: 10.30837/itssi.2022.21.039.

[24] P. Yang, H. Huang, and C. Liu, "Feature selection revisited in the single-cell era," *Genome Biology*, vol. 22, no. 1, Dec. 2021, doi: 10.1186/s13059-021-02544-3.

[25] S. Alelyani, J. Tang, and H. Liu, "Feature selection for clustering: A review," in *Data Clustering*, Chapman and Hall/CRC, pp. 29–60, 2018.

[26] M. B. Kursa, A. Jankowski, and W. R. Rudnicki, "Boruta-A system for feature selection," *Fundamenta Informaticae*, vol. 101, no. 4, pp. 271–285, 2010, doi: 10.3233/fi-2010-288.

[27] N. Farhana, A. Firdaus, M. F. Darmawan, and M. F. Ab Razak, "Evaluation of boruta algorithm in DDoS detection," *Egyptian Informatics Journal*, vol. 24, no. 1, pp. 27–42, Mar. 2023, doi: 10.1016/j.eij.2022.10.005.

[28] I. Y. Abdi *et al.*, "Cross-sectional proteomic expression in Parkinson's disease-related proteins in drug-naïve patients vs healthy controls with longitudinal clinical follow-up," *Neurobiology of Disease*, vol. 177, 2023, doi: 10.1016/j.nbd.2023.105997.

[29] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1/3, pp. 389–422, 2002, doi: 10.1023/a:1012487302797.

[30] J. Brownlee, *Machine learning mastery*. Machine Learning Mastery, 2022.

[31] G. Borboudakis and I. Tsamardinos, "Forward-backward selection with early dropping," *Journal of Machine Learning Research*, vol. 20, no. 8, pp. 1–39, 2019.

[32] N. Pudjihartono, T. Fadason, A. W. Kempa-Liehr, and J. M. O'Sullivan, "A review of feature selection methods for machine learning-based disease risk prediction," *Frontiers in Bioinformatics*, vol. 2, Jun. 2022, doi: 10.3389/fbinf.2022.927312.

[33] K. Kumari and M. Mrunalini, "Detecting denial of service attacks using machine learning algorithms," *Journal of Big Data*, vol. 9, no. 1, Apr. 2022, doi: 10.1186/s40537-022-00616-0.

[34] A. S. Saud, S. Shakya, and B. Neupane, "Analysis of depth of entropy and GINI index based decision trees for predicting diabetes," *Indian Journal of Computer Science*, vol. 6, no. 6, Jan. 2021, doi: 10.17010/ijcs/2021/v6/i6/167641.

[35] I. D. Mienye and Y. Sun, "A survey of ensemble learning: concepts, algorithms, applications, and prospects," *IEEE Access*, vol. 10, pp. 99129–99149, 2022, doi: 10.1109/access.2022.3207287.

[36] U. T. Nagar, *A study on feature analysis and ensemble-based intrusion detection scheme using CICIDS-2017*. Ph.D. diss., 2021.

[37] V. Kumar, A. Kumar, S. Garg, and S. R. Payyavula, "Boosting algorithms to identify distributed denial-of-service attacks," *Journal of Physics: Conference Series*, vol. 2312, no. 1, Aug. 2022, doi: 10.1088/1742-6596/2312/1/012082.

[38] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/a:1010933404324.

[39] A. Sarica, A. Cerasa, and A. Quattrone, "Random forest algorithm for the classification of neuroimaging data in alzheimer's disease: a systematic review," *Frontiers in Aging Neuroscience*, vol. 9, Oct. 2017, doi: 10.3389/fnagi.2017.00329.

[40] I. Colakovic and S. Karakatič, "Adaptive boosting method for mitigating ethnicity and age group unfairness," *SN Computer Science*, vol. 5, no. 1, Nov. 2023, doi: 10.1007/s42979-023-02342-7.

[41] W. Liang, S. Luo, G. Zhao, and H. Wu, "Predicting hard rock pillar stability using GBDT, XGBoost, and LightGBM algorithms," *Mathematics*, vol. 8, no. 5, May 2020, doi: 10.3390/math8050765.

[42] G. Ke *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, 2017.

[43] K. ÇOŞKUN and G. ÇETİN, "A comparative evaluation of the boosting algorithms for network attack classification," *International Journal of 3D Printing Technologies and Digital Industry*, vol. 6, no. 1, pp. 102–112, Apr. 2022, doi: 10.46519/ij3dptdi.1030539.

[44] E. S. Alghoson and O. Abbass, “Detecting distributed denial of service attacks using machine learning models,” *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 12, 2021, doi: 10.14569/ijacsa.2021.0121277.

[45] H. Nhat-Duc and T. Van-Duc, “Comparison of histogram-based gradient boosting classification machine, random Forest, and deep convolutional neural network for pavement raveling severity classification,” *Automation in Construction*, vol. 148, Apr. 2023, doi: 10.1016/j.autcon.2023.104767.

[46] Canadian Institute for Cybersecurity, “DDoS 2019 Datasets,” University of New Brunswick. Accessed: Feb. 11, 2023. [Online]. Available: at https://www.unb.ca/cic/datasets/ddos-2019.html

[47] L. D’hooge, M. Verkerken, B. Volckaert, T. Wauters, and F. De Turck, “Establishing the contaminating effect of metadata feature inclusion in machine-learned network intrusion detection models,” in *Lecture Notes in Computer Science*, Springer International Publishing, pp. 23–41, 2022.

[48] M. Kuhn and K. Johnson, “Applied predictive modeling,” *Applied predictive modeling*, vol. 26, no. 13, 2013.

[49] A. P. Devi and K. J. Singh, “A machine learning approach to intrusion detection system using UNSW-NB-15 and CICDDoS2019 datasets,” in *Smart Innovation, Systems and Technologies*, Springer Singapore, 2021, pp. 195–205.

[50] O. Sbai and M. El boukhari, “Data flooding intrusion detection system for MANETs using deep learning approach,” Sep. 2020, doi: 10.1145/3419604.3419777.

[51] M. S. Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, “DDoSNet: A deep-learning model for detecting network attacks,” in *2020 IEEE 21st International Symposium on "A World of wireless, mobile and multimedia networks"(WoWMoM)*, Aug. 2020, doi: 10.1109/wowmom49955.2020.00072.

[52] A. E. Cil, K. Yildiz, and A. Buldu, “Detection of DDoS attacks with feed forward based deep neural network model,” *Expert Systems with Applications*, vol. 169, May 2021, doi: 10.1016/j.eswa.2020.114520.

BIOGRAPHIES OF AUTHORS

Leila Bagdadi Ph.D. student and Associate Professor at University of Sciences and Technology of Oran Mohamed Boudiaf USTO-MB, Algeria. She Holds a Master degree in Computer Science with specialization in software and network engineering. Area for research: intrusion detection system, pervasive system, security in pervasive systems, injection attacks and aspect-oriented programming. She can be contacted at email: leila.bagdadi@univ-usto.dz.

Belhadri Messabih degree in Computer Sciences from the University of Paris 7, France, in 1997. Professor at University of Science and Technology of Oran-USTO-Algeria, Computer Science Department. Area for research: computer science, information technology, computer systems and computational processes, embedded system, network architecture, network security, intrusion detection system, pervasive system, Java card, and molecular docking. He can be contacted at email: b.messabih@gmail.com.