ISSN 2007-9737

Optimizing the Performance of the IDS through Feature-Relevant Selection Using PSO and Random Forest Techniques

Benaissa Safa[1,*], Reda Mohamed-Hamou[2], Adil Toumouh[1]

[1] Djillali Liabes University, Computer Science Department, EEDIS Laboratory, Sidi Bel Abbes, Algeria

[2] Dr. Tahar Moulay University, Computer Science Department, GeCoDe Laboratory, Saida, Algeria

benaissa.safa@univ-sba.dz, hamoureda@yahoo.fr, toumouh@gmail.com

Abstract. As the world becomes more digitalized, the potential for attacks increases, therefore, effective techniques for intrusion detection on network are needed. In this study, the authors propose a two steps approach. First, the Correlation-based Features Selection as a feature evaluator based on Particle Swarm Optimization is used to select the relevant features. This evaluator is compared with other evaluators. Second, the Random Forest algorithm is used to classify attacks in a network. A comparative study is also performed conducted with different classifiers such as Naïve Bayes, Stochastic Gradient Descent, Deep Learning, k-Nearest Neighbors and Support Vector Machine. Experiments were conducted on the NSL-KDD database and the results show an efficiency of 98.78% for binary classification. The performance results obtained show that the proposed technique performs better than other competing techniques.

Keywords. Classification, feature selection, intrusion detection system, machine learning, NSL-KDD data set, particle swarm optimization, random forest.

1 Introduction

Nowadays, the Internet has sparked a great technological revolution in terms of the exchange of information, knowledge and science between individuals and even institutions; at the same time, the use of the web has become one of the essential necessities of our daily life. Unfortunately, this dependence on the web has led some individuals to exploit it illegally through hacking, espionage, data theft, extortion and other malicious activities.

This reality poses a significant security threat to both individuals and companies. This issue is also becoming a real challenge for computer science researchers and developers.

Therefore, it is necessary to implement a security policy to protect company data and personal information from unexpected attacks. several tools are available to ensure data protection and personal information. The purpose of this protection is to reduce the risks associated with the confidentiality, integrity and availability of data.

An Intrusion Detection System (IDS) is considered to be the most important tool to ensure the functionality of computer security systems, because the IDS is the only tool that can guarantee the stability of the system, and then, because most attacks occur after an intrusion or by the injection of a malicious application. It is in charge of the response in the event of an attack as well as the stop or continuity strategies [8].

There are two main types of intrusion detection approaches in the literature: those based on scenarios (such as signature research, pattern matching, etc.) and those based on behavioral

Table 1. Summary table of some related works

Used algo/model	Data set	Classification	Accuracy (%)	Ref.
NDAE (DL - AE - RF)	NSL-KDD	5-class	85.42	[27]
	10% KddCup'99	5-class	97.85	
DNN - AE – SM	NSL-KDD	2-class, 5-class	-	[26]
DL - AE – SM (STL, SMR)	NSL-KDD	2-class, STL 2-class, SMR 5-class, STL 5-class, SMR	88.39 78.06 79.10 75.23	[19]
AE – DBN	10% KddCup'99	2-class	92.10	[3]
DBN	40% NSL-KDD	5-class	97.45	[15]
DBN	10% KddCup'99	5-class	93.49	[4]
DBN – LR	10% KddCup'99	5-class	97.90	[14]
DRBM	10% KddCup'99	2-class	94.00	
DNN	10% KddCup'99 NSL-KDD	2-class 5-class 2-class 5-class	93.00 93.50 80.10 78.50	[31]
RNN	NSL-KDD Test+ Test-21 Test+ Test-21	2-class 5-class	83.28 68.55 81.29 64.67	[35]
LSTM RNN	KddCup'99	5-class	97.54	[21]
HC + SVM	KddCup'99	5-class	95.72	[17]
CT + SVM	1998 DARPA	5-class	69.80	[20]
NB + KNN	NSL-KDD	5-class	84.86	[25]
KNN + SVM + PSO	KddCup'99	5-class	88.72	[1]
K-means + KNN	KddCup'99	5-class	99.01	[30]
GMMs + PSO + SVM	KddCup'99	5-class	99.99	[18]
FL + GA	10% KddCup'99	5-class	94.60	[10]
K-Means + NB + BNN	KddCup'99	5-class	99.90	[11]

approaches (for example, Bayesian analysis, statistical analysis and neural networks).

This last category aims to recognize abnormal behavior, compared to a definition or a modeling of normal or abnormal behaviors learned from a prior observation of the system, and in this case, learning seems possible. In contrast, in a scenario-based approach, the IDS relies on a pre-existing knowledge base referencing the various known attacks likely to be implemented in a computer system.

This knowledge is used by the IDS for the recognition of events produced by intrusion actions in the computer system that it observes. Therefore, this method requires regular updating of the knowledge base and the IDS focuses directly on the identification of misuse.

It is also possible to compare intrusion detection systems based on the data sources they rely on. Some IDS, known as HIDS (Host IDS) are based on the execution histories of specific programs or instruction sequences, which are often provided by the operating system but sometimes also by applications. Other IDS, typically known as NIDS (Network IDS), analyzes the packets sent over the network.

In theory, two response modes can be distinguished for IDS. Usually, a passive response

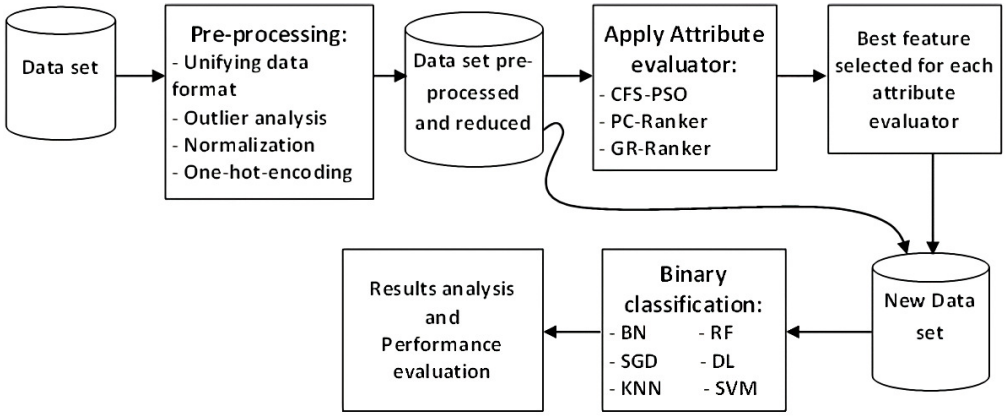


Fig. 1. Proposed research methodology

is adopted: the IDS broadcasts an alert and identifies the detected attack to an analysis or broadcast system by recording the detected intrusions in a log file.

However, more active responses should be considered, where the IDS aims to stop an attack at the moment of its detection by interrupting a connection or even counter-attacking [23].

In order to improve the efficiency of intrusion detection systems, several solutions have been proposed in this field. The authors remain focused on achieving this objective by conducting research on the use and integration of bio-inspiration techniques in general and particle swarm optimization (PSO) in particular.

PSO is a bioinspired optimization metaheuristic that was proposed by Eberhart and Kennedy in 1995 [12]. The technique of optimizing particle swarm was inspired by the collective behavior of birds or fish schools.

Each particle in the PSO is a fish or a bird in search space, with its own specific coordinates: position and velocity. Prior to searching for the optimum global position, particles try to maintain their local best positions [9]. In this paper, it is proposed to use the Correlation based Features Selection (CFS) feature evaluator, based on the bio-inspired technique of PSO, for selecting only the relevant features. Subsequently, the Random Forest (RF) classifier is chosen for attack classification in a network.

The RF algorithm is one of the most popular machine learning techniques. The sections of this article are arranged in the following order: Section 2 provides the related works in the field of intrusion detection systems, distinguishing, those that are based on machine learning methods and some others that focus on deep learning.

Section 3 presents the author's proposal, followed by a brief analysis of the KDDCup'99 data set and its versions used in this article, such as statistics and data preprocessing. This section concludes with a description of the different evaluation metrics used. Section 4 explores the analysis and discussion of the experimental results. Finally, section 5 presents a conclusion and future research directions suggested.

2 Related Work

Information security is an interesting area of research for its importance in the daily lives of individuals and even for institutions.

An intrusion detection system (IDS) is considered an important policy to improve the quality of computer security. In recent years, a considerable number of literature searches on intrusion detection have been published. In this section, a selection of this works is presented.
During the preceding decade, several studies have been done in the intrusion detection area, some of them based on machine learning methods and

Table 2. Composition of KDDCup'99 training data set (before and after preprocessing)

Connection Type	Before preprocessing	After preprocessing	
	No. of instances	No. of unique instances	Reduction (%)
DoS	3,883,370	247,267	93.63
Probe	41,102	13,860	66.28
R2L	1,126	999	11.28
U2R	52	52	00.00
T. Attacks	3,925,650	262,178	93.32
Normal	972,781	812,814	16.44
Total	4,898,431	1,074,992	78.05

Table 3. Composition of NSL-KDD data sets

Connection Type	Training set		Test set	
DoS	45,927	36.46%	7,458	33.08%
Probe	11,656	9.25%	2,421	10.74%
R2L	995	0.79%	2,754	12.22%
U2R	52	0.04%	200	0.89%
Total Attacks	58,630	46.54%	12,833	56.93%
Normal	67,343	53.46%	9,711	43.07%
Total	125,973	100%	22,544	100%

others focusing on deep learning. First, a few studies based on machine learning techniques are presented, followed by a few others based on deep learning.

2.1 IDS based on Machine Learning Techniques

In [6], the authors propose an algorithm for feature selection. The authors used these selected features to build an intrusion detection system based on the least squares support vector machine LSSVM-IDS.

They tested their experiment on three data sets such as KDDCup'99, NSL-KDD and Kyoto 2006+, and they showed that their algorithm gives improved accuracy per attack class. The paper presented by Altwaijry and Algarny in 2012 [5] explains the use of a Naïve Bayesian classifier for intrusion detection.

The authors evaluated their proposal by category of attacks on the 10 percent of KDDCup'99 and the corrected-KDD data set. In their article referenced by [22], the authors focused their work on the cluster center and nearest neighbor (CANN) approach to feature representation with the aim of detecting intrusions.

Table 4. Confusion matrix for binary classification

		Predicted class	
		Instance Normal	Attack
Actual class	Instance Normal	TN	FP
	Attack	FN	TP

They evaluated their experimentation on the KDDCup'99 data set. They used four types of attacks. In 2016, Han X. et al. [16] suggested principal component analysis for feature extraction and proposed an algorithm for intrusion detection based on the traditional Naïve Bayesian classification algorithm.

The authors used the 10 percent subset of KDDCup'99 (494,020 records, including 19.69 percent normal and 80.31 percent attack) to evaluate the performance of their solution.

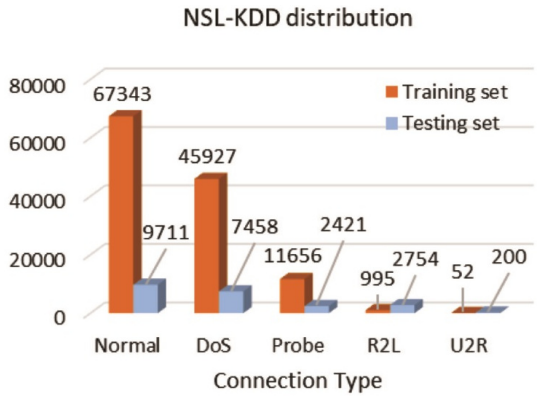
2.2 IDS based on Deep Learning Techniques

Since 2006, several studies on deep learning methods for intrusion detection have been published. the paper presented by Tang et al. in 2016 [28] explains an approach based on a deep neural network composed of an input layer of 6 dimensions, three hidden layers of 12, 6 and 3 neurons respectively and a 2-dimensional output layer.

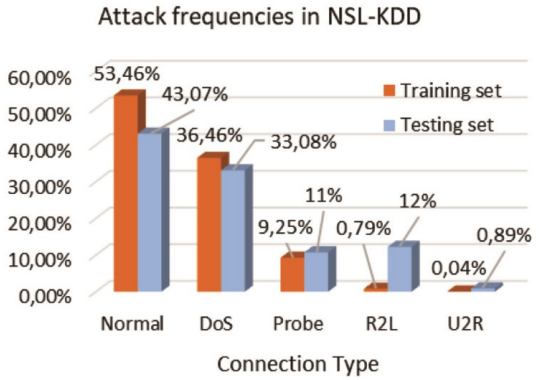
The authors tested their approach on the NSL-KDD data set and their model achieved an accuracy around 75.75 percent. The NDAE (Non-symmetric Deep Auto-Encoder) model, based on a Deep Auto-Encoder is proposed by Shone et al. in 2018 [27].

In this study, the number of attributes was reduced to 28 instead of a total of 41 attributes by this Auto-Encoder. The proposed model is composed of an input layer, six hidden layers and an output layer.

Their model is evaluated using the 10 percent subset of KDDCup'99 and NSL-KDD data set, and



(a) NSL-KDD distribution



(b) Attak frequencies in NSL-KDD

Fig. 2. Description of NSL-KDD data set

an accuracy of 97.85 percent and 85.42 percent for the two data sets respectively is obtained by the authors after using a random forest-based classifier for 5-class classification.

In 2016, Javaid et al. [19] developed a flexible and efficient NIDS (Network Intrusion Detection System) based on a proposed deep learning approach. The authors apply the technique of self-directed learning (STL).

They use the NSL-KDD data set to evaluate their system which achieved an accuracy rate of 88.39 percent and 79.10 percent for 2-class and 5-class respectively. In [4], the paper presented by authors explains the application of a Restricted Boltzmann Machine (RBM) and a Deep Belief Network (DBN) for a suggested deep learning approach to detect anomalies.

A feature reduction is performed by a first RBM. And the resulting weights are passed to a second RBM to create the DBN. the authors tested their approach on the KDDcup'99 data set, and their model showed improved accuracy (97.9 percent).

In 2017, Yin et al. [35] proposed an approach based on deep learning using a recurrent neural network (RNN-IDS). They chose the sigmoid function for activation and SoftMax as a classification function. The authors implemented their solution and tested it on the NSL-KDD data set.

The evaluation of their proposal shows an accuracy rate of 83.28 percent and 81.29 percent for a binary and multi-class (5-class) classification respectively on the KDDTest+ data set and an accuracy rate of 68.55 percent and 64.67 percent for a 2-class and 5-class respectively on the KDDTest-21 data set. A summary of some related works is shown in Table 1 below.

NDAE: Non-symmetric Deep Auto-Encoder; DL: Deep Learning; BNN: Back-propagation Neural Network; ANN: Artificial Neural Network; DNN: Deep Neural Network; RNN: Recurrent Neural Network; DBN: Deep Belief Network; DRBM: Discriminative Restricted Boltzmann Machine; AE: Auto-Encoder; SM: Soft-Max; SMR: Soft-Max Regression; STL: Self-Taught Learning; CT: Clustering Tree; LSTM: Long Short-Term Memory; GMMs: Gaussian Mixture Models; IDS: intrusion detection system; MDS: Malicious Detection System; NADS: Network Anomaly Detection System; LR: Logistic Regression; RF: Random Forest; HC: Hierarchical Clustering; NB: Naïve Bayes; K-Means; FL: Fuzzy Logic, GA: Genetic Algorithm; KNN: K-Nearest Neighbor; SVM: Support Vector Machine; PSO: Particle Swarm Optimization. SGD: Stochastic Gradient Descent.

3 Proposed Approach

As stated in some research, such as presented by Maniriho and Ahmad in 2018 [24], certain features have no influence in the attack detection process, or in other words, these unnecessary features may have a negative impact on attack determination performance.

Table 5. Feature set of KddCup'99 and NSL-KDD data set

No. f	Feature label	Type	No. f	Feature label	Type
Basic features class (B)			**Traffic 'same-Service' features class (TS)**		
1	Duration	Continuous	23	Count	Continuous
2	protocol_type	Symbolic	24	srv_count	Continuous
3	service	Symbolic	25	serror_rate	Continuous
4	flag	Symbolic	26	srv_rerror_rate	Continuous
5	src_bytes	Continuous	27	serror_rate	Continuous
6	dst_bytes	Continuous	28	srv_rerror_rate	Continuous
7	land	Symbolic	29	same_rerror_rate	Continuous
8	wrong_fragment	Continuous	30	diff_srv_rate	Continuous
	urgent	Continuous	31	srv_diff_host_rate	Continuous
Content features class (C)			**Traffic 'same-Host' features class (TH)**		
10	Hot	Continuous	32	dst_host_count	Continuous
11	num_failed_logins	Continuous	33	dst_host_srv_count	Continuous
12	logged_in	Symbolic	34	dst_host_same_srv_rate	Continuous
13	num_compromised	Continuous	35	dst_host_diff_srv_rate	Continuous
14	root_shell	Continuous	36	dst_host_same_src_port_rate	Continuous
15	su_attempted	Continuous	37	dst_host_srv_diff_host_rate	Continuous
16	num_root	Continuous	38	dst_host_serror_rate	Continuous
17	num_file_creations	Continuous	39	dst_host_srv_serror_rate	Continuous
18	num_shells	Continuous	40	dst_host_rerror_rate	Continuous
19	num_access_files	Continuous	41	dst_host_srv_rerror_rate	Continuous
20	num_outbound_cmds	Continuous			
21	is_host_login	Symbolic			
22	is_guest_login	Symbolic			

The study described in this section aims to propose an IDS model based on the machine learning methods for the attack detection, based on the features selection that has an important influence in the attack determination process.

To achieve this objective and select only the relevant features for training of the proposed model, various feature evaluators were employed by conducting multiple tests.

Three evaluators, namely Correlation based Features Selection (CFS), Pearson's Correlation (PC) and Gain Ratio (GR), were the focus of these test.

The ranking scores generated by feature class using these three evaluators were used to select twenty-one considered as relevant out of a total of forty-one. The proposed model is illustrated in the block diagram in Figure 1 below.

After is a brief description of the three evaluators used.

3.1 Correlation based Feature Selection

The principle of the Correlation based Features Selection (CFS) is to measure Pearson's correlation between an attribute and the class, it

Table 6. Features selected by different techniques for binary classification

Attribute Evaluator: **Search Method:**	**CFS** **PSO**	**Pearson's Correlation** **Ranker**	**Gain Ratio** **Ranker**
Features class	Position of the 21 Best selected features	Position of the 21 Best selected features	Position of the 21 Best selected features
Basic (B)	1, 3, 4, 5, 6, 7	3, 4, 8	3, 4, 5, 6, 8
Content (C)	12, 14, 15, 16, 21, 22	12	12
Traffic 'same-Service' (TS)	26, 27, 29, 30	23, 25, 26, 27, 28, 29, 30, 31	23, 25, 26, 28, 29, 30, 31
Traffic 'same-Host' (TH)	34, 35, 37, 38, 39	32, 33, 34, 35, 36, 38, 39, 40, 41	32, 33, 34, 35, 37, 38, 39, 41

determines the value of the attribute. By treating each value as an indicator, nominal properties are evaluated individually.

A weighted average is used to determine the overall correlation of a nominal attribute. The particle swarm optimization method is chosen as the search method for this feature evaluator. This approach was invented by Eberhart and Kennedy in 1995 [12]. The principle of PSO is population-based, which aims to find a sub-optimal solution in the search space.

At each iteration of the PSO algorithm, each individual (particle X_i) changes and updates by the two best values, the best solution (local position) based on its speed that the particle X_i has obtained so far and the best position global [13].

3.2 Pearson's Correlation

The Pearson coefficient indicator denoted r is a measure used to detect the presence or absence of a linear relationship between two variables.

The value of this measure of correlation varies from -1 to $+1$, a positive measure indicates that the two variables vary together in the same direction, when the value of r is close to $+1$, we say that there is a strong correlation.

While a negative measure indicates that one variable increases, the other decreases and when this value is close to -1, we say that there is a strong negative correlation, we say that an absence of a relationship between two variables if r takes 0 value. We remind that the formula (1) below to calculate the Pearson correlation coefficient:

$$r = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}}. \quad (1)$$

3.3 Gain Ratio

An extension of information gain, called gain ratio, was used to select the best feature feature for splitting the dataset. The gain ratio is calculated by normalizing the information gain with aid of division information.

A feature will be favored by the gain of information if it has a large number of values. The Gain Ratio (GR) is calculated as follows:

$$\mathrm{GR}(S, f_j) = \frac{\mathrm{IG}(S, f_j)}{\mathrm{SI}(S, f_j))}, \quad (2)$$

where:
IG: is the Information Gain.
SI: is the Split Information can be calculated as follows:

$$\mathrm{SI}(S, f_j) = -\sum_{S_{jk} \in S_j} (\frac{|S_{jk}|}{|S|} * log_2(\frac{|S_{jk}|}{|S|})), \quad (3)$$

where:
C: a set of classes.
S: a training data set.
$\overrightarrow{f}$: a feature vector.
S_j : a hyper-set containing sets with the same values of the feature f_j.
$IG(S, f_j)$: the information gain by splitting the dataset S with the feature f_j.

Table 7. DR and FAR of different classifiers when using different feature selection evaluators

	DR			**FAR**		
ML	CFS-PSO	PC-Ranker	GR-Ranker	CFS-PSO	PC-Ranker	GR-Ranker
NB	0.6977	0.6892	0.6889	0.0485	0.0461	0.0483
RF	0.9884	0.9851	0.9887	0.0130	0.0288	0.0158
SGD	0.9056	0.9605	0.9495	0.0460	0.0745	0.0764
DL	0.9107	0.9518	0.9522	0.0491	0.0716	0.0730
KNN	0.9801	0.9813	0.9792	0.0268	0.0355	0.0278
SVM	0.9108	0.9606	0.9476	0.0721	0.0738	0.0754

3.4 Dataset Description

Research in the field of intrusion detection (ID) requires the use of data sets to evaluate the efficiency and effectiveness of the proposed solutions by researchers in order to achieve concrete objectives. In this context, there are a variety of freely accessible network-based data sets available for intrusion detection research.

Among these data sets, we focused on the KDDCup'99 data set in our work; this data set is primarily concerned with intrusion detection and was constructed and modified from original network traffic data collected by the DARPA 1998 evaluation program under the supervision of the Massachusetts Institute of Technology (MIT) Lincoln Laboratory.

The data set in question is often used in the literature and comported of around 4,900,000 connection records, each of which is composed of 41 values and is labeled as either normal or an attack, each value corresponding to a different feature [29]. The KDDCup'99 data set can be listed as a normal traffic class and four categories to group the different kinds of attacks as shown below:

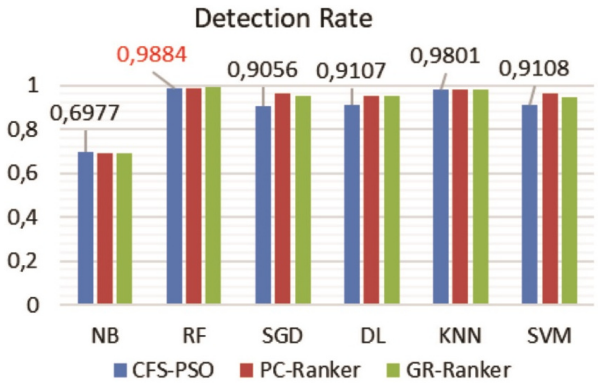
- Normal: it indicates that the network traffic record is normal or benign.
- Denial of Service attack (DoS): an intrusion or a kind of attack that tries to make some computing resources (server, host, memory, ...) inaccessible for the client, such as memory that is too full, with the objective of using the victim's resources.
- Probing attack (Probe): this category of attack includes all kinds of malicious activity, in which the perpetrator gathers detailed information about the system infrastructure and its security configurations, and for the goal by passing the firewall and conducting critical attacks.
- Remote to Local attack (R2L): the intruder does not belong to the computer network, but sends packets to the server or to another machine as a local user in order to gain access.
- User to Root attack (U2R): after several attempts to access network resources, the intruder has the character of a legitimate or normal user. Then, it attempts to access root or superuser privileges.

In 2009, Tavallaee et al. [29] provided and developed a new refined and improved version of the KDDCup'99 corpus under the appellation NSL-KDD.

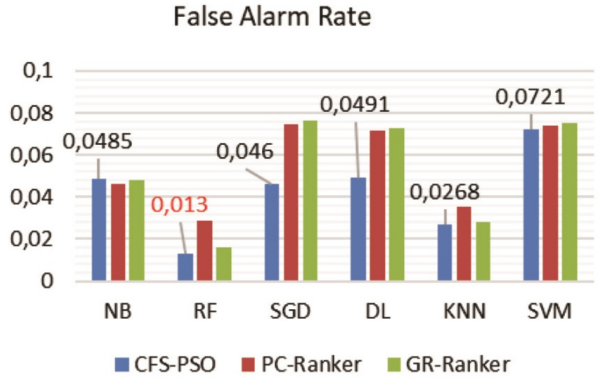
For security researchers, the number of publicly available data sets for network IDS (NIDS) is limited. KDDCup'99 and NSL-KDD are the most widely utilized and publicly available data sets for testing the effectiveness of different existing and newly announced machine learning methods [32].

In this paper, the NSL-KDD data set is used to train and test the proposed solution for intrusion detection. This version of the data set is derived from the main KDDCup'99 data set.

It reduced and improved the data set version which contains 125,973 instances. A brief description of these data sets is reported in Table



(a) Detection rate



(b) False Alarm rate

Fig. 3. Performance evaluation of different classifiers when using different feature selection evaluators

2 and Table 3. The different connections types for KddCup'99 and NSL-KDD data set are:

- Probe (Probing): ipsweep, nmap, portsweep, satan.
- DoS (Denial of Service): back, land, neptune, pod, smurf, teardrop.
- U2R (User to Root): buffer_overflow, loadmodule, perl, rootkit.
- R2L (Remote to Local): ftp_write, guesspasswd, imap, multihop, phf, spy, warezclient, warezmaster.

3.5 Data Preprocessing

As mentioned above, the KDDCup'99 and NSL-KDD data sets gather 41 features of different types and are distributed as follows, three of a nominal type such as 'Protocol type', 'Service' and 'Flag', four are binary and the thirty-four remaining features are of continuous type.

Knowing that most of the algorithms and methods only work with numbers and in order to obtain better results from experiments, a preprocessing must be performed on the data sets.

Firstly, using the One-hot-encoding [36] for transformed the nominal features to discrete features, for example, dummy variables are used to encode the textual values of the 'Protocol Type' feature (i.e. [1,0,0], [0,1,0], [0,0,1] for tcp, udp, icmp), knowing that the nominal features 'Protocol type', 'Service' and 'Flag' of the 10% KDDCup'99 training data set have 3, 66 and 11 categories respectively.

Secondly, another main step to complete is the standard normalization, also called standardization or z-score normalization. The purpose of this step is to scale all features in order to guarantee that all predictor values are on the same scale.

The principle of z-score normalization is to subtract from the data their empirical mean μ and divide them by their standard deviation σ. In this case, we apply the formula of equation (1) shown below.

Such that, for each feature j, $\mu(j)$ and $\sigma(j)$ denote respectively the mean and the standard deviation of the data vector X^j of the feature j, where each value of the vector X_i^j is transformed according to equation (4):

$$X_i^j = \frac{X_i^j - \mu(j)}{\sigma(j)}. \tag{4}$$

During the data set preprocessing phase, the training and testing databases in the KDDCup'99 collection have a multitude of duplicate instances.

This duplication represents one of the main disadvantages of this data set. These redundancies have a negative impact on the results of the experiments, and must therefore be removed. It is noted that, the training and test data sets, respectively, had about 78.05 percent and 80.68 percent of duplicated instances [29], (see Table 2).

Table 8. Precision and accuracy rate of different classifiers when using different feature selection evaluators

	Precision			**Accuracy**		
ML	CFS-PSO	PC-Ranker	GR-Ranker	CFS-PSO	PC-Ranker	GR-Ranker
NB	0.9500	0.9518	0.9496	0.8070	0.8032	0.8021
RF	0.9902	0.9783	0.9881	0.9878	0.9791	0.9868
SGD	0.9630	0.9446	0.9426	0.9264	0.9454	0.9383
DL	0.9608	0.9462	0.9452	0.9280	0.9417	0.9413
KNN	0.9797	0.9733	0.9790	0.9771	0.9741	0.9762
SVM	0.9435	0.9450	0.9432	0.9182	0.9458	0.9377

Often, in the preprocessing procedure for data sets, it is also important to remove records that contain incorrect values in the fields, such as character strings arranged in numerical fields or vice versa, missing values, etc.

After preprocessing the KDDCup'99 databases. It was noticed that 4,898,431 records which constitute the initial training set was reduced to 1,074,992 unique data points due to redundancy, this significant reduction represents a rate of 78.05 percent as shown in Table 2.

Similarly, for the KDDCup99's test set, it was noted that, a total number of 2,984,154 data points was reduced to 576,449 unique instances which represents a reduction rate of 80.68 percent. The results of this table (Table 3) are interpreted in Figures 2a and 2b below.

As previously stated, it is noted that all instances of the same KDDCup'99 data set or its derivatives are composed of 41 features. each feature has only one type of continuous, discrete or symbolic variable [33].

Generally, features are divided into four aspects or classes (see Table 5), the first nine features relate to basic intrinsic properties of the network connection, such as connection duration, protocol type, network service (http, telnet, etc.), etc.

Are grouped to form a first aspect or base class (B). The following thirteen features correspond to domain knowledge or the content of a network connection. The purpose of the content aspect features (C) is to assess the payload of the original TCP packets and to detect attacks that are hidden and not commonly present such as those of the U2R and R2L classes.

In this case, to identify such attacks, the researchers retrieved information on the amount of login failures, which suggest intrusive behavior [34]. The other two classes are encapsulated under the name Traffic; this large traffic aspect groups features which are called time-based and calculated with respect to a time interval.

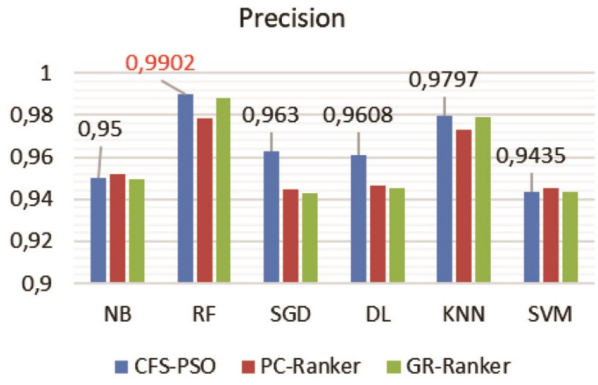
The first of the traffic aspects includes the "same service" (TS) features, consists to examine only connections established during the last two continuous seconds which have the same service as the present connection.

The second traffic aspect includes the last ten features that are from "same host" (TH), consists of an analysis of the connections made in the last continuously two seconds which have the identical final host as the present connection in order to calculate the behavioral statistical properties of the network connection, relating to protocol, serving, etc. [2].

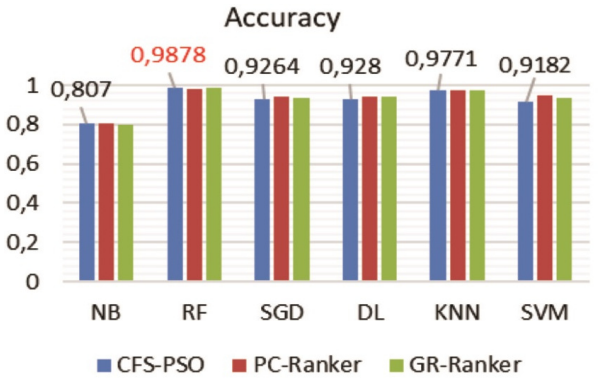
3.6 Evaluation Criteria

Generally, to evaluate the IDS detection precision, the following measures are often used:

– True Positive (TP): this metric represents the number of attacks detected and correctly classified by the model.
– True Negative (TN): a metric that indicates the number of normal instances predicted and correctly classified as normal traffic.



(a) Precision



(b) Accuracy

Fig. 4. Precision and accuracy rate of different classifiers when using different feature selection evaluators

– False Positive (FP): this metric represents the number of normal instances recognized and incorrectly classified as attacks by the model.
– False Negative (FN): a metric that indicates the number of attacks predicted and incorrectly classified as normal traffic by the model.

These metrics often form the confusion matrix values shown in Table 4 below for a binary classification problem.

Other measures were used that can be calculated based on the values of this confusion matrix as presented in Table 4, as follows: Detection Rate (DR) or True Positive Rate (TPR):

$$\mathrm{DR} = \frac{TP}{TP + FN}. \tag{5}$$

False Alarm Rate (FAR) or False Positive Rate (FPR):

$$\mathrm{FAR} = \frac{FP}{TN + FP}. \tag{6}$$

Precision:

$$\mathrm{Precision} = \frac{TP}{TP+FP}. \tag{7}$$

Overall accuracy is defined as the proportion of instances in a set of occurrences that have been correctly classified. This metric is less useful in the case where there is a significant imbalance between the classes:

$$\mathrm{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{8}$$

4 Experiment Results and Discussion

After applying the three attribute evaluation metrics (CFS-PSO, PC-Ranker, GR-Ranker), the results obtained for binary classification are shown in Table 6. Therefore, for each feature class, it is also important to choose the most relevant or influential features for the intrusion detection process.

So, the most relevant features are chosen for each class (Basic: B, Continent: C, Traffic same Service: TS and Traffic same Host: TH) by following the order of features based on their order of merit in their respective classes. These features are presented in Table 6.

For example, if CFS-PSO technique used in binary classification case, the best features selected for Basic Class are (Duration, service, flag, src_bytes, dst_bytes, land).

4.1 Analysis of Experimental Results

After selecting the top twenty-one features for each attribute evaluator from the entire data set. In the binary classification experiments, the resulting data set can be trained and tested using a variety of machine learning techniques, such as Naïve Bayes, Random Forest, Stochastic Gradient Descent, Deep Learning, K-Nearest Neighbors and Support Vector Machine.

The obtained results are presented below. Based on the corresponding new NSL-KDD data

Table 9. Comparison of the results with other algorithms (NSL-KDD data set used)

Method	Accuracy (%)	Ref.
DL-AE-SM(SMR)	78.06	[19]
DL-AE-SM(STL)	88.39	
DNN	80.1	[31]
RNN	83.28	[35]
MI-BGSA	88.36	[7]
Proposed-method	98.78	

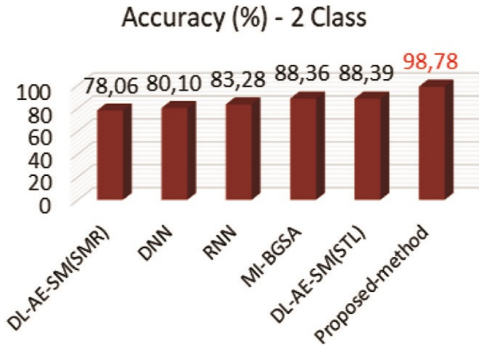


Fig. 5. Comparison of accuracy rate with other algorithms (2-class) which NSL-KDD used

set, which contains only the twenty-one best selected features for each attribute evaluator (presented in Table 6), various performance measures can be calculated such as DR, FAR, precision and system accuracy, based on the results of the confusion matrix.

Table 7 presents the obtained results of the Detection Rate and False Alarm Rate measurements of each of the machine learning techniques and for each used attribute evaluator. These results are interpreted in Figures 3a and 3b.

In the same way, precision and accuracy measurements can be calculated. Table 8 presents the obtained results of precision and accuracy measurements of each machine learning techniques and for each used attribute evaluator. The results of this table are interpreted in Figures 4a and 4b.

Finally, Table 9 shows a performance comparison of the proposed method with some other recent methods using the same data set (NSL-KDDTest) in terms of accuracy. It can be seen from the table that the proposed method (CFS-PSO + RF) ranks first in terms of accuracy in binary classification case.

Therefore, the proposed CFS-PSO attribute evaluator-based RF classifier performs better than all other competitive techniques for binary classification case (see Figure 5).

4.2 Discussion of Experimental Results

In a data set, applying a method for eliminating unnecessary features is indispensable because these extra features decrease the precision and efficiency of the prediction algorithms. Additionally, as the number of features in a data set grows, so does the searchable space.

In this research, feature selection and reduction were performed by keeping only the most relevant features. To accomplish this, three attribute evaluation metrics were applied: CFS-PSO, PC-Ranker and GR-Ranker in binary classification. The results are shown in Table 6.

In order to improve the DR and optimize the performance of the IDS, the three attribute evaluation metrics can be applied to the data set, by selecting the same number of relevant features for each of these metrics.

After running several tests, twenty-one relevant features were selected. Various performance measures were calculated, including DR, FAR, precision and system accuracy, based on the results of the confusion matrix, the obtained results are discussed as follows: In the binary classification case, the performance comparison results are shown in Tables 7 and 8, which indicate that the proposed technique (CFS-PSO attribute evaluation metric combined with RF classifier) achieved a higher DR of 98.84%, while the False Alarm Rate (FAR is 1.3%) is also the lowest compared to other machine learning techniques. In terms of precision and accuracy, Figures 4a and 4b also show a comparison of performances and prove that the proposed method takes the first place with a precision rate of 99.02% and an accuracy rate of 98.78%.

5 Conclusion and Future Work

This paper discusses an effective intrusion detection technique that is divided into two phases. In the first phase, relevant features were selected by eliminating those that do not have a significant influence on the intrusion detection procedure.

This was achieved by using an attribute elevator technique called Correlation based Features Selection (CFS) technique based on the Particle Swarm Optimization (PSO) method, resulting in a feature space reduction of approximately 50%.

In the second phase, the proposed classification algorithm Random Forest (RF) and different machine learning algorithms were tested to evaluate the performance of the proposed method, experiments were conducted on the new NSL-KDD data set containing only twenty-one features.

The experiments carried in this study are divided into three classes, Firstly, a comparison is made between the chosen attribute evaluator (CFS-PSO) and two other evaluators, such as PC-Ranker and GR-Ranker, in the second set of experiment, a comparison is made between the proposed classifier (RF) and other machine learning classifiers, namely NB, SGD, DL, KNN and SVM.

The experimental results on the NSL-KDD data set show the promising performance of the proposed techniques in terms of accuracy and detection rate compared to competitive methods.

In the final class of experiments, the proposed technique is compared to different previously existing methods.

The obtained performance results indicate that the proposed technique outperforms other methods in the binary classification. Finally, it should be noted that the current study has two major limitations, namely real-time operation and the ability to detect zero-day attacks.

To address these limitations and further improve the proposed technique, future work could focus on finding more efficient solutions for detecting zero-day attacks and developing an IDS that works in real-time. It is recommended to test the technique on other data sets such as UNSW-NB15, CSE-CIC-IDS2018.

References

1. **Aburomman, A. A., Reaz, M. B. I. (2016).** A novel SVM-kNN-PSO ensemble method for intrusion detection system. Applied Soft Computing, Vol. 38, pp. 360–372. DOI: 10.1016/j.asoc.2015.10.011.
2. **Aggarwal, P., Sharma, S. K. (2015).** Analysis of KDD dataset attributes - class wise for intrusion detection. Procedia Computer Science, Vol. 57, pp. 842–851. DOI: 10.1016/j.procs.2015.07.490.
3. **Alom, M. Z., Bontupalli, V., Taha, T. M. (2016).** Intrusion detection using deep belief networks. 2015 National Aerospace and Electronics Conference (NAECON), pp. 339–344. DOI: 10.1109/NAECON.2015.7443094.
4. **Alrawashdeh, K., Purdy, C. (2016).** Toward an online anomaly intrusion detection system based on deep learning. 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 195–200. DOI: 10.1109/ICMLA.2016.0040.
5. **Altwaijry, H., Algarny, S. (2011).** Bayesian based intrusion detection system. IAENG Transactions on Engineering Technologies: Special Edition of the World Congress on Engineering and Computer Science, Vol. 1, pp. 29–44. DOI: 10.1016/j.jksuci.2011.10.001.
6. **Ambusaidi, M. A., He, X., Nanda, P., Tan, Z. (2016).** Building an intrusion detection system using a filter-based feature selection algorithm. IEEE Transactions on Computers, Vol. 65, No. 10, pp. 2986–2998. DOI: 10.1109/TC.2016.2519914.
7. **Bostani, H., Sheikhan, M. (2017).** Hybrid of binary gravitational search algorithm and mutual information for feature selection in intrusion detection systems. Soft Computing, Vol. 21, No. 9, pp. 2307–2324. DOI: 10.1007/s00500-015-1942-8.
8. **Boudia, A., Hamou, R. M., Amine, A. (2017).** A new meta-heuristics for intrusion detection system inspired from

the protection system of social bees. International Journal of Information Security and Privacy, Vol. 11, No. 1, pp. 18–34. DOI: 10.4018/IJISP.2017010102.

9. **Bousmaha, R., Hamou, R. M., Amine, A. (2022).** Optimizing connection weights in neural networks using hybrid metaheuristics algorithms. International Journal of Information Retrieval Research, Vol. 12, No. 1, pp. 1–21. DOI: 10.4018/ijirr.289569.
10. **Chadha, K., Jain, S. (2015).** Hybrid genetic fuzzy rule based inference engine to detect intrusion in networks. Intelligent Systems and Computing, Vol. 321, pp. 185–198. DOI: 10.1007/978-3-319-11227-5_17.
11. **Dubey, S., Dubey, J. (2015).** KBB: A hybrid method for intrusion detection. 2015 International Conference on Computer, Communication and Control (IC4), pp. 1–6. DOI: 10.1109/IC4.2015.7375704.
12. **Eberhart, R., Kennedy, J. (1995).** A new optimizer using particle swarm theory. MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39–43. DOI: 10.1109/MHS.1995.494215.
13. **Elngar, A., Mohamed, D. A., Ghaleb, F. F. (2013).** A real-time anomaly network intrusion detection system with high accuracy. Information Sciences Letters, Vol. 2, No. 2, pp. 49–56. DOI: 10.12785/isl/020201.
14. **Fiore, U., Palmieri, F., Castiglione, A., De-Santis, A. (2013).** Network anomaly detection with the restricted boltzmann machine. Neurocomputing, Vol. 122, pp. 13–23. DOI: 10.1016/j.neucom.2012.11.050.
15. **Gao, N., Gao, L., Gao, Q., Wang, H. (2014).** An intrusion detection model based on deep belief networks. 2014 Second International Conference on Advanced Cloud and Big Data, pp. 247–252. DOI: 10.1109/CBD.2014.41.
16. **Han, X., Xu, L., Ren, M., Gu, W. (2015).** A naive bayesian network intrusion detection algorithm based on principal component analysis. 2015 7th International Conference on Information Technology in Medicine and Education (ITME), pp. 325–328. DOI: 10.1109/ITME.2015.29.
17. **Horng, S. J., Su, M. Y., Chen, Y. H., Kao, T. W., Chen, R. J., Lai, J. L., Perkasa, C. D. (2011).** A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert Systems with Applications, Vol. 38, No. 1, pp. 306–313. DOI: 10.1016/j.eswa.2010.06.066.
18. **Hu, W., Gao, J., Wang, Y., Wu, O., Maybank, S. (2014).** Online adaboost-based parameterized methods for dynamic distributed network intrusion detection. IEEE Transactions on Cybernetics, Vol. 44, No. 1, pp. 66–82. DOI: 10.1109/TCYB.2013.2247592.
19. **Javaid, A., Niyaz, Q., Sun, W., Alam, M. (2016).** A deep learning approach for network intrusion detection system. Proceedings of the 9th EAI International Conference on Bio-Inspired Information and Communications Technologies (Formerly BIONETICS), pp. 21–26. DOI: 10.4108/eai.3-12-2015.2262516.
20. **Khan, L., Awad, M., Thuraisingham, B. (2007).** A new intrusion detection system using support vector machines and hierarchical clustering. The VLDB Journal, Vol. 16, pp. 507–521. DOI: 10.1007/s00778-006-0002-5.
21. **Kim, J., Kim, H. (2017).** An effective intrusion detection classifier using long short-term memory with gradient descent optimization. 2017 International Conference on Platform Technology and Service (PlatCon), pp. 1–6. DOI: 10.1109/PlatCon.2017.7883684.
22. **Lin, W. C., Ke, S. W., Tsai, C. F. (2015).** CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. Knowledge-Based Systems, Vol. 78, pp. 13–21. DOI: 10.1016/j.knosys.2015.01.009.

23. **Lokbani, A. C., Lehireche, A., Hamou, R. M., Boudia, M. A. (2015).** An approach based on social bees for an intrusion detection system by scenario. Securing the Internet of Things: Concepts, Methodologies, Tools, and Applications, pp. 914–938. DOI: 10.4018/978-1-5225-9866-4.ch040.

24. **Maniriho, P., Ahmad, T. (2018).** Analyzing the performance of machine learning algorithms in anomaly network intrusion detection systems. 2018 4th International Conference on Science and Technology (ICST), pp. 1–6. DOI: 10.1109/ICSTC.2018.8528645.

25. **Pajouh, H. H., Javidan, R., Khayami, R., Dehghantanha, A., Choo, K. K. R. (2019).** A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks. IEEE Transactions on Emerging Topics in Computing, Vol. 7, No. 2, pp. 314–323. DOI: 10.1109/TETC.2016.2633228.

26. **Potluri, S., Diedrich, C. (2016).** Accelerated deep neural networks for enhanced intrusion detection system. 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), Vol. 2016-November, pp. 1–8. DOI: 10.1109/ETFA.2016.7733515.

27. **Shone, N., Ngoc, T. N., Phai, V. D., Shi, Q. (2018).** A deep learning approach to network intrusion detection. IEEE Transactions on Emerging Topics in Computational Intelligence, Vol. 2, No. 1, pp. 41–50. DOI: 10.1109/TETCI.2017.2772792.

28. **Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A. R., Ghogho, M. (2016).** Deep learning approach for network intrusion detection in software defined networking. 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM), pp. 258–263. DOI: 10.1109/WINCOM.2016.7777224.

29. **Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A. A. (2009).** A detailed analysis of the KDD CUP 99 data set. 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1–6. DOI: 10.1109/CISDA.2009.5356528.

30. **Tsai, C. F., Lin, C. Y. (2010).** A triangle area based nearest neighbors approach to intrusion detection. Pattern Recognition, Vol. 43, No. 1, pp. 222–229. DOI: 10.1016/j.patcog.2009.05.017.

31. **Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., Venkatraman, S. (2019).** Deep learning approach for intelligent intrusion detection system. IEEE Access, Vol. 7, pp. 41525–41550. DOI: 10.1109/ACCESS.2019.2895334.

32. **Vinayakumar, R., Soman, K. P., Poornachandran, P. (2017).** Evaluating effectiveness of shallow and deep networks to intrusion detection system. 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1282–1289. DOI: 10.1109/ICACCI.2017.8126018.

33. **Wang, G., Hao, J., Mab, J., Huang, L. (2010).** A new approach to intrusion detection using artificial neural networks and fuzzy clustering. Expert Systems with Applications, Vol. 37, No. 9, pp. 6225–6232. DOI: 10.1016/j.eswa.2010.02.102.

34. **Xiao, Y., Xing, C., Zhang, T., Zhao, Z. (2019).** An intrusion detection model based on feature reduction and convolutional neural networks. IEEE Access, Vol. 7, pp. 42210–42219. DOI: 10.1109/ACCESS.2019.2904620.

35. **Yin, C., Zhu, Y., Fei, J., He, X. (2017).** A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access, Vol. 5, pp. 21954–21961. DOI: 10.1109/ACCESS.2017.2762418.

36. **Zhang, Q., Bao, H., You, Y., Lee, K., Guo, D. (2018).** Category coding with neural network application. arXiv. DOI: 10.48550/arXiv.1805.07927.

ISSN 2007-9737

Article received on 18/04/2023; accepted on 23/04/2024.
** Corresponding author is Benaissa Safa.*