ISSN 2007-9737

PSO and Random Forest Techniques to Improve IDS Performance for Multi-class Classification

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

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

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

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

Abstract. With the increasing digitization of the world, the risk of attacks also increases, creating a need to develop effective network intrusion detection techniques. In this research, the authors proposed a two-phase approach to improve IDS performance in the multi-class classification case. In the first phase, only the relevant features are identified and conserved using an evaluator based on Particle Swarm Optimization. In the second phase, network attacks are classified using the Random Forest classifier. Furthermore, a comparative study is conducted, involving other classifiers such as Naïve Bayes, Stochastic Gradient Descent, Deep Learning, etc. For multi-class classification, the NSL-KDD data set was used to conduct experiments, and the obtained results showed an accuracy of 99.40%. The performance results of our technique are presented and compared with other competing techniques. The obtained results clearly indicate that our technique outperforms the others.

Keywords. Feature selection, intrusion detection system, machine learning, multi-class classification, particle swarm optimization, random forest, NSL-KDD dataset.

1 Introduction

Today, the internet has become indispensable to our daily life, this interpreted by the exponentially increasing volume of information exchanged each day information exchanged every day. Unfortunately, the reliance on the Internet has given rise to instances where certain individuals use it unlawfully, engaging in activities unwanted such as hacking, extortion, data theft, espionage and other malicious acts. Consequently, this stark reality presents a substantial security risk to both individuals and corporations alike.

Faced with this unstable and growing situation, which has become a major challenge for researchers and developers in the field of cybersecurity, it is imperative to find and implement a robust security policy that effectively protects private and corporate data against unexpected threats. Various solutions are available to guarantee the privacy and protection of data and personal information. The objective of this protection is to minimize the dangers related to the availability, confidentiality, and integrity of data.

The most crucial tool for ensuring the performance of computer security systems is an intrusion detection system (IDS). An IDS serves as the first line of defense, ensuring the stability of the system and detecting the majority of attacks that occur after an intrusion or due to the introduction of a malicious application. It is the first responsible for stop or continuity strategies, as well as reaction in the case of an attack [7].

In the existing literature, two primary categories of intrusion detection approaches are commonly discussed: scenario-based approaches, which include signature research, pattern matching, etc.; and behavioral approaches, such as Bayesian analysis, statistical analysis, and neural networks. The idea behind the scenario-based approach is that the IDS is based on a pre-existing knowledge base that includes different known attacks expected to be performed in a computer system. Using this knowledge, the IDS can effectively detect and identify events triggered by intrusion attempts on the monitored computer system. This strategy requires frequent updates to the knowledge base, and the IDS is entirely focused on abuse detection. On the other hand, behavioral approaches aim to identify abnormal behavior by comparing it to a predefined definition or model of normal or abnormal behaviors, which is acquired through prior observations of the system. As a result, the learning process seems feasible.

The literature identifies two primary categories of IDS based on their data sources: Host-based IDS (HIDS) and Network-based IDS (NIDS). HIDS based on information collected from host-specific data sources such as operating systems, system files, etc. In contrast, NIDS relies on information constitutes network packets. IDS can also be classified by their response modes. Passive IDS generate alerts and record detected threats. Conversely, active IDS take counter-measures such as interrupting connections or launching counter-attacks [23].

Numerous ideas have been proposed in this field with the aim of improving the efficiency and performance of intrusion detection systems. The authors have been exploring the use and integration of nature-inspired approaches, particularly particle swarm optimization (PSO), to achieve this goal. PSO is a metaheuristic optimization technique introduced by Eberhart and Kennedy in 1995 [12].

Drawing inspiration from biological systems, specifically the collective behavior observed in birds or fish schools. In PSO, a population of particles moves within a search space to find the optimal solution. Each particle represents a bird or a fish and is characterized by its position and speed in this space [8].

After achieving satisfactory and validated results for binary classification of cybercrime tasks to improve the effectiveness of IDS [28]. In this study, the authors explore the feasibility of extending their methodology to multi-class attack categorization. Firstly, the CfsSubset feature evaluator, based on the bio-inspired PSO method is used to select only the most relevant features from the dataset. Next, the Random Forest (RF) algorithm is used to classify network attacks in the multi-class classification case. This algorithm is widely recognized as one of the most popular machine learning techniques.

The sections of this paper are presented as follows: an overview of related works in the field of intrusion detection systems is presented in section 2, covering both deep learning and machine learning techniques. In section 3, the authors present their proposed method along with a brief description of the statistical analysis and pre-processing of the data from the KDDCup'99 dataset, as well as the version used in this study and an exploration of various evaluation measures used. The analysis and discussion of the experimental results are presented in section 4. Section 5 concludes the paper with recommendations for future

2 Related Work

In the field of computer security, intrusion detection systems (IDS) are considered crucial for improving information security. Numerous studies on intrusion detection have been published recently. Some of them focus on machine learning methods, while others on deep learning. This section provides an overview of several works in this field.

In a study published in 2023 by Olivia et al. [25], the authors demonstrated the use of their models (FFNN, LSTM) for intrusions identification. They evaluated their proposed method by attack type using the NSL-KDD and BoT-IoT datasets. The results indicated that their approach achieved a superior accuracy rate in detecting intrusions.

In another paper presented by Han X. et al. [17], the authors proposed an algorithm

for Naive Bayesian network intrusion detection that incorporated principal component analysis (PCA) to reduce the dimensionality of the features space. The authors evaluated the performance results of their experiments on the 10% subset of KDDCup'99 dataset, which contains various types of attacks.

In 2019, Kwon et al. [21] used with their initial experiments a fully connected network (FCN) model, to improve the analysis of network traffic for network anomaly detection. The authors evaluated their approach on the NSL-KDD dataset, achieving an accuracy of approximately 90%.

In their article, Ambusaidi et al. [5] proposed an algorithm for feature selection based on mutual information. The selected features were then utilized to develop an intrusion detection system using the Least Squares Support Vector Machine (LSSVM-IDS) on three different datasets, including KDDCup'99, NSL-KDD, and Kyoto 2006+, the authors demonstrated that their method achieved higher accuracy for each attack type.

A flexible and efficient Network Intrusion Detection System (NIDS) was developed by Javaid A. et al. in 2015 [20], based on a proposed deep learning approach. The authors utilized the Self-taught Learning (STL) technique and evaluated their system on the NSL-KDD dataset. Their proposed model achieved an accuracy of approximately 79.10 percent for 5-class classification.

In their research, Alrawashdeh and Purdy [3] utilized a Restricted Boltzmann Machine (RBM) and a Deep Belief Network (DBN) to implement a deep learning approach for intrusion detection. The first RBM performed a feature reduction. The weights generated by this RBM were used by a second RBM to construct the DBN. The authors evaluated their approach using the KDDcup'99 dataset, achieving an improved accuracy rate of 97.9 percent. In their paper, Yin et al. [35] introduced a deep learning method for intrusion detection, utilizing recurrent neural networks (RNN-IDS). The implemented solution by authors was tested on the NSL-KDD dataset. Their proposed multi-class classification approach was evaluated and achieved an accuracy rate of 81.29 percent and 64.67 percent on the KDDTest+ and KDDTest-21 datasets, respectively.

In 2024, Bakır and Ceviz [6] proposed a hybrid feature selection and genetic algorithm-based hyperparameter tuning method for RF-based IDS. Notably, their approach achieved over a 98% reduction in training time on the CICIDS2017 dataset for both binary and multi-class classification tasks. In 2018, Shone et al. [29] developed an NDAE (Non-symmetric Deep Auto-Encoder) model that used a Deep Auto-Encoder. In their research, they employed the auto-encoder to reduce the total number of features from 41 to only 28. The proposed model consists of an input layer, six hidden layers and an output layer. To evaluate their model, the authors utilized the 10% subset of KDDCup'99 and NSL-KDD datasets. They achieved an accuracy rate of 97.85 percent and 85.42 percent, respectively, for the two datasets by employing a random forest-based classifier for the multi-classification. A set of related works is summarized in Table 1 below.

3 The Proposed Approach

This section presents two proposals, the first one aims to select only relevant features that have a significative influence on the attack detection process while eliminating unnecessary ones. Therefore, it is crucial to identify and eliminate redundant and irrelevant features from a dataset prior to using, as demonstrated by Maniriho and Ahmad in their research [24], because, the complexity of the model can be reduced if there are fewer attributes and consequently the performance of the attack determination task can be increased.

The second proposal focuses on an IDS model based on machine learning techniques for attack detection.

To achieve the goal of the first proposal, several tests were conducted using various feature evaluators to select the most appropriate features for training the model proposed in the second proposal. Three evaluators, such as, Correlation based Features Selection (CfsSubset), Pearson's Correlation (PC) and Gain Ratio (GR), were employed in this initial phase. For each evaluator and out of a total of forty-one features, only

Table 1. Related work summary table (5-class used)

Used algo/model	Dataset	Accuracy (%)	Ref.
FCN	NSL-KDD	89.40	[21]
CNN,DAE	Bot-IoT	98.39	[14]
	CSE-CIC-IDS2018	97.37	
FFNN	NSL-KDD	98.67	[25]
LSTM		96.44	
DM	NSL-KDD	98.27	[10]
SM		96.75	
DBN	40% NSL-KDD	97.45	[2]
DBN	10% KddCup'99	93.49	[15]
DBN – LR	10% KddCup'99	97.90	[3]
DNN	10% KddCup'99	93.50	[32]
	NSL-KDD	78.50	
RNN	NSL-KDD Test+	81.29	[35]
	NSL-KDD Test-21	64.67	
NDAE (DL - AE - RF)	NSL-KDD	85.42	[29]
	10% KddCup'99	97.85	
DNN - AE – SM	NSL-KDD	-	[27]
FL + GA	10% KddCup'99	94.60	[9]
K-Means + NB + BNN	KddCup'99	99.90	[11]
DL - AE – SM (STL)	NSL-KDD	79.10	[20]
DL - AE – SM (SMR)		75.23	
LSTM RNN	KddCup'99	97.54	[22]
GMMs + PSO + SVM	KddCup'99	99.99	[18]
NB + KNN	NSL-KDD	84.86	[26]
KNN + SVM + PSO	KddCup'99	88.72	[1]

twenty-one were considered relevant and selected based on their ranking scores.

Using the newly datasets formed with the relevant features, multiples tests were conducted employing different machine learning classifiers; in this second phase, an IDS model based on the Random Forest algorithm was proposed. This algorithm has demonstrated its efficiency in the field of classification in general and multi-class classification field in particular. The suggested model is presented in the block diagram below (see Figure 1).

Below is a brief description of the three evaluators used in the first phase of this study.

— Correlation based Feature Selection (CfsSubset)

The Correlation based Feature Selection (CfsSubset) technique combines a feature evaluation method with a suitable correlation measurement and heuristic searching algorithm. This technique is employed to select the most important features from a collection of features. The preferred feature subsets exhibit

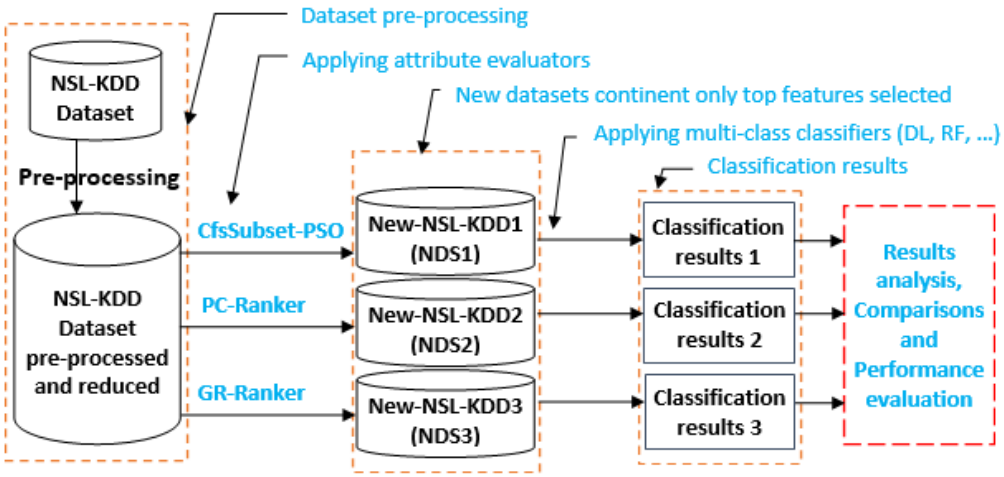


Fig. 1. Proposed research methodology

low intercorrelation and high correlation with the class [16].

The fundamental idea of the Correlation based Features Selection (CfsSubset) is to calculate the attribute value by measuring the Pearson's correlation between it and the class. For the feature evaluator based on Correlation based Features Selection, the particle swarm optimization (PSO) approach is chosen as the search strategy. This method was developed by Eberhart and Kennedy in 1995 [12].

The basic idea of the PSO algorithm is based on a populational approach to determine a sub-optimal solution within the search space. Each particle X_i is modified and updated based on the two best values at each iteration of the PSO algorithm: the best local position, which depends on the speed that the particle X_i has already reached, and the best global position [13].

— Pearson's correlation

Pearson's correlation coefficient, denoted r, is a statistical measure utilized to determine the presence and strength of a linear relationship between two variables. The value of this coefficient varies between -1 to $+1$. A value of $+1$ indicates a strong positive correlation, while a value close to -1 indicates a strong negative correlation. A value of 0 indicates no correlation between the two variables. the Pearson correlation coefficient is calculated using Equation (1) below.

$$r = \frac{Cov(X,Y)}{\sigma_X . \sigma_Y}. \tag{1}$$

Were, $Cov(X,Y)$ is the covariance between variables X and Y; σ_X and σ_Y are the standard deviations of X and Yrespectively.

The Pearson correlation coefficient can also be calculated by the formula in Equation (2) below.

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

Were, x_i and y_i are the individual sample points of the two variables X and Y respectively; $\bar{x}$ and $\bar{y}$ are the means of X and Y respectively.

— Gain ratio

Gain ratio is a metric derived from information gain that is used to select the optimal features for dataset splitting. The value of the gain ratio is determined based on the information gain normalization using division information.

This metric is calculated using the following formula (3).

$$GR(S, f_j) = \frac{IG(S,f_j)}{SI(S,f_j)}. \quad (3)$$

Where, S is the training dataset; f_j is the j-th feature.

Note that a feature has a more values, it will be prioritized by the information gain.

$IG(S, f_j)$ the information gain by splitting the dataset S with the feature f_j. This metric can be calculated using Equation (4).

$$IG(S, f_j) = H(S) - H(S/f_j). \quad (4)$$

Were, $H(S/f_j)$ is the conditional entropy and $H(S)$ represents the entropy of S, can be calculated as follows:

$$H(S) = -\sum_{i=1}^{m} p_i log(p_i). \quad (5)$$

p_i is the probability that an element of S belongs to subset S_i.

The split information (SI) can be calculated using Equation (6).

$$SI(S, f_j) = -\sum_{S_{jk} \in S_j} (\frac{|S_{jk}|}{|S|} * log_2(\frac{|S_{jk}|}{|S|})). \quad (6)$$

Here, S is the training dataset; S_j a hyper-set containing sets with the same values of the feature f_j;

3.1 Dataset Description

In the field of cybersecurity, the assessment and evaluation of researchers' work and their proposed solutions often rely on the use of datasets. In this context, the KDDCup'99 dataset has been widely utilized in research studies to evaluate the performance of intrusion detection systems based on machine learning techniques, deep learning and others algorithms. This dataset is freely accessible and available on the network. The dataset in question was constructed and modified from the original network traffic data collected by the DARPA 1998 evaluation program, and prepared by [30] under the supervision of MIT's Lincoln Laboratory. The KDD Cup'99 training set contains approximately 4.9 million connection records. Each record consists of 41 values of distinct features. A normal value or a specific attack type labels each record [31]. The labels in the KDDCup'99 dataset can be categorized into five classes, as shown below:

— Normal: This class represents any normal or benign activity of the network traffic.

— Denial of service (DoS): Attacks in this class aim to disrupt network or system availability by rendering specific computing resources inaccessible to clients, for example, overloading memory resources to exhaust victim resources.

— Probing (Probe): This class encompasses various malicious activities in which the attacker attempts to gather detailed information about a target network or system, such as security configuration, in order to identify vulnerabilities, bypass firewalls, and prepare for future attacks.

— Remote to local (R2L): This category includes intruders who do not belong to the network but attempts to gain unauthorized access to the target system or network, by exploiting vulnerabilities in network protocols or services.

— User to root (U2R): Intruders have limited access to a system and make multiple attempts to obtain root or superuser privileges, in order to gain complete administrative control by exploiting vulnerabilities in the user's privileges or system configuration.

As mentioned above, the KDDCup'99 dataset is extensively utilized by computer security researchers. NSL-KDD is a new refined and improved version of the original corpus (KDDCup'99). It was developed and provided by Tavallaee et al. [31] to address some limitations of the original dataset. Both datasets are widely used and accessible to researchers in the security field to evaluate the efficiency and effectiveness of various existing and recently studied machine learning techniques [33].

Table 2. Composition of NSL-KDD datasets [31]

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%

The NSL-KDD dataset is a reduced and enhanced version, it consists of 125,973 instances. In the literature, the different types of attack connections of the training set for this dataset version are categorized as follows [19]:

— Denial of Service: Neptune, Smurf, Tear drop, Back;

— Probing: Ip sweep, Nmap, Port sweep, Satan;

— User to Root: Buffer overflow, Load module, Rootkit;

— Remote to Local: Guess passwd, Warezmaster, Imap, Multihop.

In this research, the proposed method for intrusion detection was trained and tested using the NSL-KDD dataset. Table 2 presents a brief description of this dataset.

The NSL-KDD test set used in this paper illustrated in Figure (2) below.

As mentioned previously, it is important to note that each connection instance in the KDDCup'99 dataset or its variants is described by 41 features (see Table 3), and each feature is associated with only one type of variable (continuous, discrete or symbolic), as stated by Wang et al. [34].

3.2 Data Preprocessing

The KDDCup'99 dataset and all its versions contain 41 various features (see Table 3), which can be categorized as follows: three nominal features, namely Protocol type, Service and Flag; four binary features, and the remaining features are continuous. Since most algorithms and techniques only operate on numerical data, preprocessing is necessary to obtain more accurate experimental results. Firstly, the One-hot-encoding is used [36].

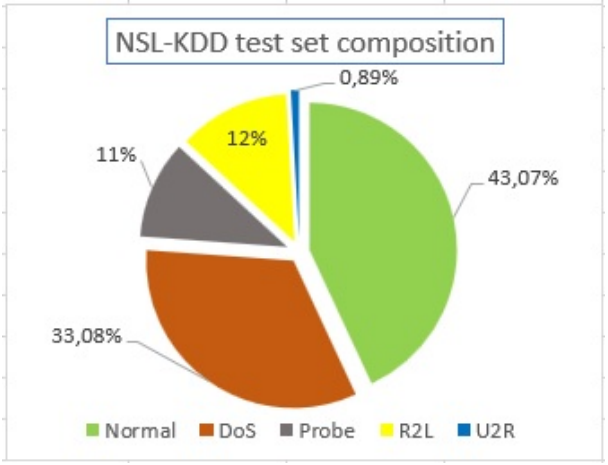


Fig. 2. NSL-KDD Test set composition

One-hot encoding is a commonly employed technique in machine learning to convert categorical variables into numerical variables that can be utilized in various techniques. In the case of NSL-KDD dataset, there are three nominal features: protocol type, service, and flag. One-hot encoding can be applied to these features to convert them into binary or discrete variables.

For example, the 'Protocol Type' feature has three different values: TCP, UDP and ICMP. One-hot encoding would create three new binary variables: 'Protocol Type_TCP', 'Protocol Type_UDP', and 'Protocol Type_ICMP'. Each of these new variables would have a value of 0 or 1, depending on whether the original 'Protocol Type' feature had that value for a particular instance in the dataset. It should be noted that the nominal features 'Protocol Type', 'Service' and 'Flag' in the dataset in question, have 3, 66 and 11 variables, respectively. The same process would be applied to the service and flag variables, creating new binary variables for each possible value of these two variables.

Secondly, normalization standard, also known as normalization or normalization z-score, is the next important step to complete. The purpose of this step is to scale all the features to ensure that all predicted values are on a uniform scale. The

Table 3. Feature set of NSL-KDD dataset

No. f	Feature label	No. f	Feature label	No. f	Feature label
1	duration	15	su_attempted	29	same_srv_rate
2	protocol_type	16	num_root	30	diff_srv_rate
3	service	17	num_file_creations	31	srv_diff_host_rate
4	flag	18	num_shells	32	dst_host_count
5	src_bytes	19	num_access_files	33	dst_host_srv_count
6	dst_bytes	20	num_outbound_cmds	34	dst_host_same_srv_rate
7	land	21	is_host_login	35	dst_host_diff_srv_rate
8	wrong_fragment	22	is_guest_login	36	dst_host_same_src_port_rate
9	urgent	23	count	37	dst_host_srv_diff_host_rate
10	hot	24	srv_count	38	dst_host_serror_rate
11	num_failed_logins	25	serror_rate	39	dst_host_srv_serror_rate
12	logged_in	26	srv_serror_rate	40	dst_host_rerror_rate
13	num_compromised	27	rerror_rate	41	dst_host_srv_rerror_rate
14	root_shell	28	srv_rerror_rate		

main concept behind z-score normalization is to compute the empirical mean μ and the standard deviation σ of the data, and then subtract the mean from each data point and divide the result by the standard deviation. The following Equation (7) can be used in this case. For each feature j, the values X_i^j of the data vector X^j are transformed using the Equation (7), where $\mu(j)$ and $\sigma(j)$ represent the mean and standard deviation of X^j respectively.

$$X_i^j = \frac{X_i^j - \mu(j)}{\sigma(j)}. \tag{7}$$

In the data pre-processing stage, it is important to note that the training and testing datasets of KDDCup'99 dataset had approximately 78.05% and 80.68% duplicate instances, respectively [31]. This duplication is a significant drawback of this dataset and has a negative impact on experimental results. Therefore, they need to be removed. Additionally, it is crucial to delete any instances with incorrect values in their fields, such as strings in numeric fields or vice versa, empty fields, etc.

3.3 Evaluation Criteria

In general, the following metrics are frequently used to evaluate the precision of intrusion detection.

Table 4. Confusion matrix for classification problem

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

— True Positive (TP): This metric indicates the number of attacks that the model correctly recognized and classified;

— True Negative (TN): This metric represents the number of normal occurrences that the model detected and correctly classified as normal instances;

— False Positive (FP): This metric indicates the number of normal occurrences that the model predicted and incorrectly classified as attacks;

— False Negative (FN): This metric represents the number of attacks that the model identified and incorrectly classified as normal instances.

These measurements are discrete values and are typically used to construct a confusion matrix for a specific categorization problem, as shown in Table 4 below.

In order to measure the precision of intrusion detection performed by the IDS, several other performance metrics including Accuracy rate, Precision, True Positive Rate and False Positive Rate, were utilized to evaluate the performance of our proposed model. These metrics can be calculated using the values presented in Table 4, which represents the confusion matrix as follows:

Accuracy rate is the proportion of the number of correctly classified occurrences to the total number of occurrences, as shown in below (see Equation (8)):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}. \quad (8)$$

Precision is a proportion defined as the number of correctly identified occurrences divided by the total number of predicted abnormal occurrences, as shown in Equation (9) below:

$$Precision = \frac{TP}{TP+FP}. \quad (9)$$

True Positive Rate (TPR) that corresponds to Detection Rate (DR), is a proportion determined by dividing the number of correctly identified instances by the total number of abnormal instances, as shown in Equation (10) below:

$$TPR = \frac{TP}{TP+FN}. \quad (10)$$

False Positive Rate (FPR) that corresponds to False Alarm Rate (FAR) is a proportion calculated by dividing the number of incorrectly rejected instances by the total number of normal instances, as shown in below (see Equation (11)):

$$FPR = \frac{FP}{TN+FP}. \quad (11)$$

4 Experiment Results and Discussion

In this study, we applied three feature evaluation metrics: CfsSubset-PSO, PC-Ranker and GR-Ranker. The experiments were conducted using the NSL-KDD dataset to assess the performance of the CfsSubset-PSO evaluator for five-categories classification, including Normal, DoS, R2L, U2R and Probe. The results obtained are presented in Table 5.

To optimize the performance of intrusion detection systems (IDS), it's crucial to select only the most relevant features that significantly influence the detection of intrusions. Feature selection techniques can be employed to identify and retain these key features. Feature merit order based selection is used in this research. Following a series of experiments, twenty-one pertinent features were selected for each feature evaluator. Table 5 lists the selected features by their corresponding positions (indices).

4.1 Analysis of Experimental Results

In the context of multi-class classification, and for each attribute evaluator applied to the full NSL-KDD dataset, we obtain a new reduced dataset containing only the top twenty-one features.

Various machine learning algorithms, including Naive Bayes, Random Forest, Stochastic Gradient Descent, Deep Learning, K-Nearest Neighbors and Support Vector Machine, can be utilized for training and testing the resulting datasets. Consequently, different performance metrics, including TPR, FPR, system precision and accuracy rate, can be calculated based on the results of the confusion matrix from the experiments conducted on each new dataset. The obtained results are presented as following.

The performance measurements for the true positive rate and false positive rate are calculated for each machine learning classifier and feature selection method. These results are presented in Table 6 and also illustrated in Figures (3a) and (3b).

Similarly, precision and accuracy measurements can be calculated for five-category classification.

The performance measurements for precision and accuracy rate are calculated for each machine learning classifier and feature selection method. These results are presented in Table 7 and also illustrated in Figures (4a) and (4b).

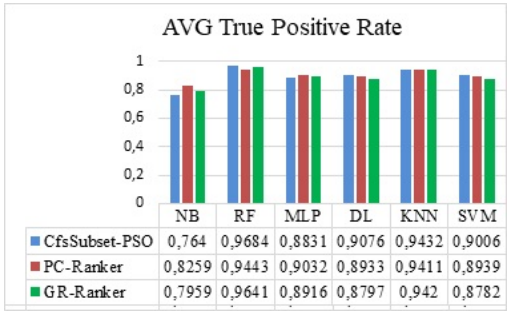
We continued our experiments in the context of multi-class categorization. Using the same dataset, we compared the performance of the proposed technique to some recent methods in terms of accuracy. The results of this comparison are presented in Table 8 below.

Table 5. Positions of 21 top features selected by three attribute evaluators for 5-class classification

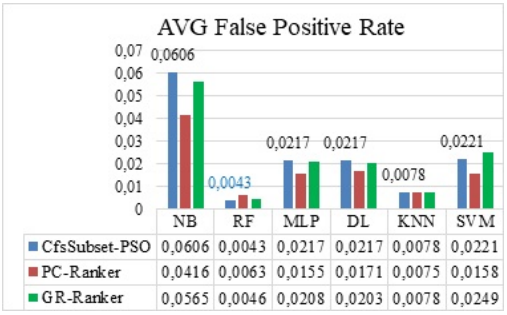
Attribute Evaluator: Search Method:	CfsSubset PSO	Pearson's Correlation Ranker	Gain Ratio Ranker
	1, 3, 4, 5, 6, 8, 9	3, 4, 12, 23, 25, 26	3, 4, 5, 6, 8, 10
	12, 14, 16, 18, 22	27, 28, 29, 30, 31	11, 12, 14, 22, 25
	23, 25, 26, 30, 31	32, 33, 34, 35, 36	26, 29, 30, 31, 33
	35, 37, 38, 39	37, 38, 39, 40, 41	34, 35, 37, 38, 39

Table 6. Performance evaluation of different classifiers using different attribute evaluators in terms of TPR and FPR (case of 5-class)

		TPR						FPR					
F. Selector	ML	Normal	Dos	Probe	R2L	U2R	AVG	Normal	Dos	Probe	R2L	U2R	AVG
CfsSubset-PSO	NB	0.8204	0.7675	0.9364	0.3707	0.925	0.7640	0.1253	0.0117	0.0417	0.0097	0.1147	0.0606
	RF	0.9848	0.9977	0.9727	0.9670	0.920	0.9684	0.0117	0.0023	0.0017	0.0053	0.0007	0.0043
	MLP	0.9366	0.9322	0.9608	0.8711	0.715	0.8831	0.0701	0.0087	0.0058	0.0223	0.0017	0.0217
	DL	0.9388	0.9103	0.9413	0.9325	0.815	0.9076	0.0664	0.0143	0.0085	0.0179	0.0015	0.0217
	KNN	0.9689	0.9916	0.9645	0.9510	0.840	0.9432	0.0189	0.0066	0.0038	0.0087	0.0013	0.0078
	SVM	0.9138	0.9116	0.9620	0.9608	0.755	0.9006	0.0573	0.0103	0.0093	0.0327	0.0009	0.0221
PC-Ranker	NB	0.8838	0.7998	0.9162	0.8348	0.695	0.8259	0.1114	0.0101	0.0210	0.0487	0.0167	0.0416
	RF	0.9736	0.9964	0.9827	0.9488	0.820	0.9443	0.0157	0.0025	0.0007	0.0115	0.0010	0.0063
	MLP	0.9253	0.9764	0.9694	0.9299	0.715	0.9032	0.0318	0.0203	0.0046	0.0191	0.0018	0.0155
	DL	0.9422	0.9399	0.9496	0.9397	0.695	0.8933	0.0428	0.0101	0.0080	0.0210	0.0036	0.0171
	KNN	0.9666	0.9945	0.9760	0.9481	0.820	0.9411	0.0180	0.0032	0.0023	0.0125	0.0013	0.0075
	SVM	0.9433	0.9460	0.9781	0.9423	0.660	0.8939	0.0425	0.0086	0.0084	0.0191	0.0005	0.0158
GR-Ranker	NB	0.9031	0.7255	0.9393	0.5817	0.830	0.7959	0.1926	0.0115	0.0260	0.0060	0.0463	0.0565
	RF	0.9870	0.9975	0.9814	0.9495	0.905	0.9641	0.0138	0.0024	0.0006	0.0055	0.0006	0.0046
	MLP	0.9370	0.9464	0.9707	0.8489	0.755	0.8916	0.0655	0.0119	0.0058	0.0193	0.0014	0.0208
	DL	0.9514	0.9612	0.9599	0.7959	0.730	0.8797	0.0672	0.0166	0.0082	0.0070	0.0026	0.0203
	KNN	0.9727	0.9925	0.9748	0.9299	0.840	0.9420	0.0220	0.0040	0.0026	0.0089	0.0016	0.0078
	SVM	0.9612	0.9134	0.9760	0.7752	0.765	0.8782	0.0983	0.0091	0.0097	0.0062	0.0014	0.0249



(a) True Positive Rate



(b) False Positive Rate

Fig. 3. Comparison of feature selection techniques for different classifiers in terms of TPR and FPR

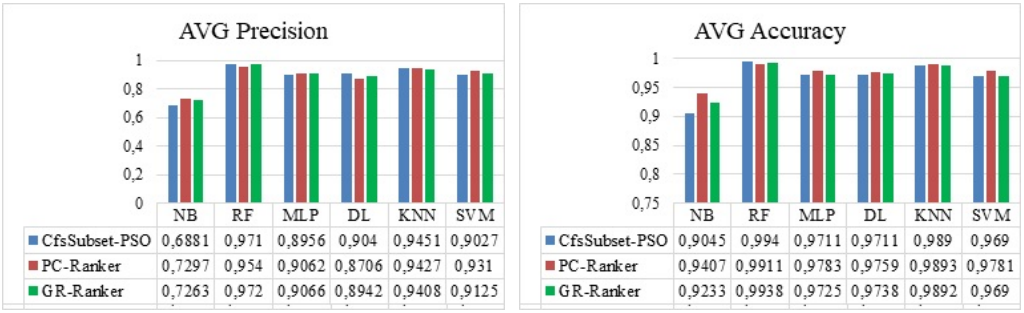
The results of the comparison in Table 8 are illustrated graphically in Figure 5 above. As can be clearly seen, the proposed method outperforms the other competing techniques.

4.2 Discussion of Experimental Results

The application of a technique to remove unused features from a data set is a crucial step as

Table 7. Performance evaluation of different classifiers using different attribute evaluators in terms of precision and accuracy rate (case of 5-class)

		Precision						Accuracy					
F. Selector	ML	Normal	Dos	Probe	R2L	U2R	AVG	Normal	Dos	Probe	R2L	U2R	AVG
CfsSubset-PSO	NB	0.9700	0.8321	0.7296	0.8417	0.0673	0.6881	0.9152	0.8513	0.9559	0.9146	0.8856	0.9045
	RF	0.9953	0.9846	0.9858	0.9624	0.9246	0.971	0.9977	0.9868	0.9956	0.9914	0.9986	0.9940
	MLP	0.9814	0.9100	0.9521	0.8444	0.7901	0.8956	0.9717	0.9328	0.9906	0.9646	0.9958	0.9711
	DL	0.9692	0.9145	0.9298	0.8789	0.8274	0.9040	0.9607	0.9359	0.9861	0.976	0.9969	0.9711
	KNN	0.9868	0.9749	0.9685	0.9380	0.8571	0.9451	0.9928	0.9759	0.9928	0.9863	0.9973	0.9890
	SVM	0.9776	0.9235	0.9257	0.8035	0.8830	0.9027	0.9638	0.9303	0.9876	0.9665	0.9969	0.9690
PC-Ranker	NB	0.9750	0.8573	0.8402	0.7048	0.2715	0.7297	0.9270	0.8866	0.9723	0.9371	0.9807	0.9407
	RF	0.9949	0.9792	0.9941	0.9201	0.8817	0.954	0.9971	0.9797	0.9975	0.9837	0.9974	0.9911
	MLP	0.9595	0.9566	0.9619	0.8717	0.7814	0.9062	0.9786	0.9497	0.9926	0.9747	0.9957	0.9783
	DL	0.9788	0.9434	0.9346	0.8615	0.6347	0.8706	0.9734	0.9508	0.9874	0.9742	0.9937	0.9759
	KNN	0.9934	0.9760	0.9809	0.9136	0.8497	0.9427	0.9960	0.9754	0.9954	0.9827	0.9971	0.9893
	SVM	0.9819	0.9437	0.9334	0.8729	0.9231	0.931	0.9764	0.9513	0.9902	0.9762	0.9965	0.9781
GR-Ranker	NB	0.9690	0.7802	0.8130	0.9309	0.1382	0.7263	0.9015	0.8487	0.9703	0.9436	0.9526	0.9233
	RF	0.9952	0.9819	0.9946	0.9603	0.9282	0.972	0.9976	0.9866	0.9974	0.9890	0.9985	0.9938
	MLP	0.9753	0.9155	0.9526	0.8599	0.8297	0.9066	0.9743	0.9356	0.9917	0.9646	0.9965	0.9725
	DL	0.9663	0.9146	0.9337	0.9408	0.7157	0.8942	0.9761	0.9408	0.9884	0.9689	0.9950	0.9738
	KNN	0.9920	0.9710	0.9780	0.9354	0.8276	0.9408	0.9949	0.9757	0.9949	0.9836	0.9970	0.9892
	SVM	0.9801	0.8810	0.9238	0.9459	0.8315	0.9125	0.9652	0.9273	0.9888	0.9671	0.9965	0.9690



(a) Precision **(b)** Accuracy

Fig. 4. Comparison of feature selection techniques of different classifiers in terms of Precision and Accuracy

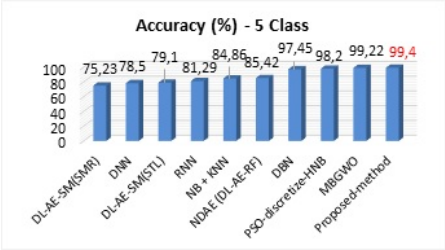


Fig. 5. Comparison of accuracy rate with other techniques (5-class) which NSL-KDD used

these additional features reduce the accuracy and efficiency of the prediction algorithms. Therefore, the search space expands when the number of features in dataset increases. In this study, three attribute evaluation metrics, namely CfsSubset-PSO, PC-Ranker and GR-Ranker, were applied to the dataset in the context of multi-class classification, to perform feature selection and reduction by retaining only the most relevant features. After conducting several experiments, twenty-one pertinent features were selected, which is the same number of pertinent features for each of these three metrics, aiming to enhance

Table 8. Comparison of the results with other techniques (NSL-KDD dataset used)

Method	Accuracy (%)	Ref.
DL-AE-SM(SMR)	75.23	[20]
DL-AE-SM(STL)	79.10	
DNN	78.50	[32]
RNN	81.29	[35]
NB + KNN	84.86	[26]
NDAE (DL-AE-RF)	85.42	[29]
DBN	97.45	[2]
PSO-discretize-HNB	98.20	[13]
MBGWO	99.22	[4]
Proposed-method	99.40	

the intrusion detection rate and consequently, optimize the performance of IDS. Table 5 shows the results of this phase. Based on the outcomes of the confusion matrix, various performance measures, including TPR, FPR, precision, and system accuracy, were determined. The obtained results are presented in detail as follows.

In this multi-class classification case, Table 6 represented by Figures (3a) and (3b) demonstrated that the proposed technique (CfsSubset-PSO attribute evaluation metric combined with RF classifier) achieves a higher average TPR of 96.84%, and a lower FPR average of 0.43% compared to other competing machine learning algorithms. Similarly, Table 7 presents a performance comparison in terms of precision and accuracy among the competing techniques used in this classification case. The results are also interpreted in Figures (4a) and (4b), respectively.

The CfsSubset-PSO attribute evaluation technique shows promising results for some classifiers, but the best result was shown by the GR-Ranker technique for the RF classifier. It achieves the highest average precision rate approximately 97.2% with a slight increase of 0.1% compared to the proposed method. However, the proposed method demonstrates the lowest FPR rate.

Additionally, Figure (4b) illustrates that the proposed method achieves the highest average accuracy of 99.40%, which is considered as the optimized accuracy compared to other machine learning algorithms such as NB, SGD, DL, KNN and SVM. Finally, Table 8 presents a performance comparison of the proposed method with other recent methods using the NSL-KDDTest dataset in terms of accuracy. It is evident from the table that the proposed method (CfsSubset-PSO + RF) ranks first in terms of accuracy in this multi-class classification case (see Figure 5). Therefore, the proposed CfsSubset-PSO attribute evaluator-based RF classifier performs better than all other competitive techniques.

5 Conclusion and Future Work

This study introduces a proposed method to enhance network intrusion detection, which involves two distinct phases. Relevant features were retained in the first phase by eliminating those with minimal or no impact on the intrusion detection process. To accomplish this, the feature selection technique based on correlation (CfsSubset) was applied to the NSL-KDD dataset, utilizing the Particle Swarm Optimization (PSO) method as a search method, a reduction of the feature space of about 50% was obtained. In the second phase, tests were conducted on the new NSL-KDD dataset, containing only twenty-one selected features, to evaluate the performance of the proposed technique. Alongside the proposed Random Forest (RF) algorithm, several other machine learning algorithms were examined.

Three categories of experiments can be distinguished for this study. The first category involves comparing), the selected attribute evaluator (CfsSubset-PSO), with two other evaluators, namely, PC-Ranker and GR-Ranker. In the second category, a comparison is performed between the suggested classifier (RF) and other machine learning classifiers, including NB, SGD, DL, KNN and SVM. The experimental results on the NSL-KDD dataset show that the suggested method performs significantly better in terms of accuracy and true positive rate compared to competing approaches. The third set of experiments compares the suggested method with various existing techniques. The performance results obtained demonstrate that the suggested

method outperforms other competing techniques in multiclass classification. This method also gives acceptable results in the case of binary classification [28]. It is recommended that this research also be extended by developing more robust models that can detect a wider range of attacks in real time, using different and larger datasets.

References

1. **Aburomman, A. A., Reaz, M. B. I. (2016).** A novel svm-knn-pso ensemble method for intrusion detection system. Applied Soft Computing Journal, Vol. 38, pp. 360–372. DOI: 10.1016/j.asoc.2015.10.011.
2. **Alom, M. Z., Bontupalli, V., Taha, T. M. (2016).** Intrusion detection using deep belief networks. 2015 National Aerospace and Electronics Conference (NAECON), Vol. 2016-March, pp. 339–344. DOI: 10.1109/NAECON.2015.7443094.
3. **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.
4. **Alzubi, Q. M., Anbar, M., Alqattan, Z. N., Al-Betar, M. A., Abdullah, R. (2020).** Intrusion detection system based on a modified binary grey wolf optimisation. Neural Computing and Applications, Vol. 32, No. 10, pp. 6125–6137. DOI: 10.1007/s00521-019-04103-1.
5. **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.
6. **Bakır, H., Ceviz, O. (2024).** Empirical enhancement of intrusion detection systems: A comprehensive approach with genetic algorithm-based hyperparameter tuning and hybrid feature selection. Arabian Journal for Science and Engineering. DOI: https://doi.org/10.1007/s13369-024-08949-z.
7. **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.
8. **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.
9. **Chadha, K., Jain, S. (2015).** Hybrid genetic fuzzy rule based inference engine to detect intrusion in networks. Advances in Intelligent Systems and Computing, Vol. 321, pp. 185–198. DOI: 10.1007/978-3-319-11227-5_17.
10. **Diro, A. A., Chilamkurti, N. (2018).** Distributed attack detection scheme using deep learning approach for internet of things. Future Generation Computer Systems, Vol. 82, pp. 761–768. DOI: https://doi.org/10.1016/j.future.2017.08.043.
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. A., El, D. A., Mohamed, A., Ghaleb, F. F. M. (2013).** A real-time anomaly network intrusion detection system with high accuracy. Information Sciences Letters An International Journal, Vol. 2, No. 2, pp. 49–56. DOI: 10.12785/isl/020201.
14. **Ferrag, M. A., Maglaras, L., Moschoyiannis, S., Janicke, H. (2020).** Deep learning for cyber

security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, Vol. 50, pp. 102419. DOI: https://doi.org/10.1016/j.jisa.2019.102419.

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. **Hall, M. A. (1999).** Correlation-based feature selection for machine learning. In Proceedings of the 17th international conference on machine learning (ICML), pp. 359–366.
17. **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.
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. **Iliyasu, U., A.S.and Abdurrahman, Zheng, L. (2022).** Few-shot network intrusion detection using discriminative representation learning with supervised autoencoder. Applied Sciences, Vol. 12, No. 5, pp. 2351. DOI: https://doi.org/10.3390/app12052351.
20. **Javaid, A., Niyaz, Q., Sun, W., Alam, M. (2015).** 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), ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, BEL, pp. 21–26. DOI: 10.4108/eai.3-12-2015.2262516.
21. **Kwon, D., Kim, H., Kim, J., Suh, S. C., Kim, I., Kim, K. J. (2019).** A survey of deep learning-based network anomaly detection. Cluster Computing, Vol. 22, No. 1, pp. 949–961. DOI: 10.1007/s10586-017-1117-8.
22. **Le, T. T. H., 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.
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. International Journal of Organizational and Collective Intelligence, Vol. 5, No. 3, pp. 44–67. DOI: 10.4018/ijoci.2015070104.
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. **Olivia, J., Beatriz, O., Eva, R., Norma, G., Héctor, A., Ramon, C. (2023).** Deep-learning based detection for cyber-attacks in iot networks: A distributed attack detection framework. Journal of Network and Systems Management, Vol. 31, No. 33, pp. 1–24. DOI: https://doi.org/10.1007/s10922-023-09722-7.
26. **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.
27. **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.

28. Safa, B., Hamou, R. M., Toumouh, A. (2024). Optimizing the performance of the IDS through feature-relevant selection using PSO and random forest techniques. Computación y Sistemas (CyS), Vol. 28, No. 2, pp. 473–488. DOI: 10.13053/CyS-28-2-4579.

29. 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.

30. Stolfo, S., Fan, W., Lee, W., Prodromidis, A., Chan, P. (2000). Cost-based modeling for fraud and intrusion detection: results from the jam project. Proceedings DARPA Information Survivability Conference and Exposition, IEEE, Vol. 2, pp. 130–144. DOI: 10.1109/DISCEX.2000.821515.

31. 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.

32. 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.

33. 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), Vol. 2017-January, pp. 1282–1289. DOI: 10.1109/ICACCI.2017.8126018.

34. 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.

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. eprint arXiv:1805.07927. DOI: https://doi.org/10.48550/arXiv.1805.07927.

Article received on 30/07/2024; accepted on 13/01/2025.
**Corresponding author is Benaissa Safa.*