Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2022.024098
Article

Tech Science Press

Bat-Inspired Optimization for Intrusion Detection Using an Ensemble Forecasting Method

R. Anand Babu[1,*] and S. Kannan[2]

[1]Department of Information Technology, E.G.S. Pillay Engineering College, Nagapattinam, Tamilnadu, India
[2]Department of Computer Science and Engineering, E.G.S. Pillay Engineering College, Nagapattinam, Tamilnadu, India
*Corresponding Author: R. Anand Babu. Email: ranandbabu215@gmail.com
Received: 04 October 2021; Accepted: 05 November 2021

Abstract: An Intrusion detection system (IDS) is extensively used to identify cyber-attacks preferably in real-time and to achieve integrity, confidentiality, and availability of sensitive information. In this work, we develop a novel IDS using machine learning techniques to increase the performance of the attack detection process. In order to cope with high dimensional feature-rich traffic in large networks, we introduce a Bat-Inspired Optimization and Correlation-based Feature Selection (BIOCFS) algorithm and an ensemble classification approach. The BIOCFS is introduced to estimate the correlation of the identified features and to choose the ideal subset for training and testing phases. The Ensemble Classifier (EC) is used to integrate decisions from three different classifiers including Forest by Penalizing Attributes (FPA), Random Forest (RF), and C4.5 based on the rule of average probabilities. The integration of BIOCFS and EC approaches aids to handle multi-class and unbalanced datasets. The performance of the proposed algorithm is evaluated on a well-known dataset NSL-KDD. The experimental results prove that our combined BIOCFS-EC outdoes other relevant methods in the context of appropriate performance measures. More importantly, the proposed IDS decreases the time complexity of training and testing procedure from 39.43 and 2.25 s to 16.66 and 1.28 s, respectively. Also, the proposed approach achieves the maximum classification accuracy of 0.994, precision of 0.993, F-measure of 0.992, the attack detection ratio of 0.992 and the minimum false alarm ratio of 0.008% on the given dataset.

Keywords: Attacks; bat algorithm; ensemble classifier; feature selection; intrusion detection

1 Introduction

The proliferation of network devices, the rapid development of hacking tools and intrusive activities make computer networks more and more vulnerable. Mostly, an intrusion would lead to loss of integrity, loss of confidentiality, unapproved utilization of resources or denial of network services. Hence, the necessity of network security has gained significant attention from academia and industries globally. The objective of IDS is to detect unapproved use, misuse, and abuse of network resources in real-time of both

insiders (i.e., authorized users who attempt to misuse their privileges) and outside intruders. Handling the problem related to IDS is a difficult endeavor due to the massive expansion of heterogeneous communicating devices, the complexity of the fast-growing number of malware, and the difficulty of mining abnormal patterns from big volumes of large-dimensional data contaminated with attacks. The security system must offer data integrity, confidentiality, and availability. Also, it guarantees improved resilience against denial-of-service (DoS) attacks [1]. Data and communication integrity services are related to the accuracy, authenticity, non-corruptibility, and credibility of communication between nodes. Data confidentiality service protects information against an illegal release. IDSs are developed to react appropriately to intrusive activities. Based on the detection mechanisms used, IDSs are divided into two types: (i) anomaly detection and (ii) misuse detection [2]. The anomaly detection methods are developed to sense the anomalous behavior of the intruder in the data traffic by finding significant deviations from a profile of a user's normal behavior. Though this type of IDSs works better in identifying new threats, they frequently hampered from a high false-positive rate [3]. In case of misuse detection, attack identification depends on traffic patterns or signatures. This process attempts to differentiate genuine samples and intrusive behaviors [4].

Machine learning techniques can be used in both anomaly and misuse detection systems. By considering a packet flow over the central node of the network, an IDS not only requires to differentiate genuine and malicious packet but also detect the particular type of attack arising in the traffic being monitored. Furthermore, only a part of the data flow may exhibit abnormal behaviors when a communication system is overwhelming with genuine data packets, which pose significant challenges in finding threats with the maximum Attack Detection Rate (ADR) and minimum False Alarm Rate (FAR). In this work, we developed a new IDS to identify different attacks with higher accuracy and efficiency. The main contribution of this paper is three-fold:

1. In order to achieve effective and accurate IDS, we develop an approach that integrates the advantages of feature selection and ensemble classification. In the context of dimensionality reduction, we implement the BIOCFS algorithm to evaluate the correlation between pairs of features and used for enhancing the performance of the selection of features. Then, the identified subset that encompasses a reduced dimension is employed in training and testing phases.
2. We present an ensemble classification method by coalescing decisions from different classifiers including FPA, RF, and C4.5 into one to improve the classification efficiency. Furthermore, we use a voting mechanism using the average of probabilities (AOP) rule to handle the multi-class problem in the classification process.
3. The combined BIOCFS and EC algorithm (BIOCFS-EC) is implemented and the results are compared with other feature selection approaches on a testbed containing dataset, viz., NSL-KDD.

The subsequent sections of this article are arranged as follows. We explore substantial relevant feature selection and ensemble classification methods in the context of IDS in Section 2. Section 3 describes the proposed work. Sections 4 and 5 describe the experimental setup and evaluation results obtained from real traffic traces. The evaluation is carried out and the performance of the proposed approach is related to other relevant approaches. Finally, we conclude this paper in Section 7.

2 Related Works

The feature selection is a preprocessing phase of the machine learning techniques to remove unrelated features and select the most relevant one to preserve or improve the performance of the system being monitored. The selection of features is used to choose a subset from the original dataset without any modification. Feature selection algorithms are pigeonholed into three categories as filter, wrapper, and

embedded approaches. Besides, ensemble approaches integrate many fundamental frameworks to generate better results (e.g., Stacking, Boosting, Bagging, etc.). Of late, feature selection and ensemble classification approaches are used in several IDSs to increase the efficiency of the system being monitored. To achieve more reliable and efficient classification, Hota et al. introduced a feature selection method to eliminate the unrelated features from the database [5]. The authors proved that the proposed method using C4.5 with a parameter called information gain (InfoGain) achieves the maximum accuracy with only 17 features. Malik et al. introduced a feature selection approach using Particle Swarm Optimization (PSO) with a RF classifier [6]. In this hybrid model, more suitable features for each class are selected to realize a low false-positive rate with higher classification accuracy related to other approaches.

Paulauskas et al. described an ensemble classifier to integrate four different classification means including J48, C5.0, Naive Bayes and PART [7]. This approach is proposed to integrate multiple weaker learners to form a robust learner. The experimental results confirmed that the ensemble approach achieves higher classification accuracy. In addition, few researchers implemented various approaches to decrease the size of the datasets. Khammassi et al. developed a genetic algorithm (GA) based wrapper technique along with a logistic regression-based learning approach for IDS to select the optimal feature subset [8]. The experiments reveal that their approach achieves higher ADR with a subset of only 18 features in the KDD99 database and 20 features in the UNSW-NB15 database. Jayakumar et al. proposed Multiple IDS Units (MIU) to identify attacks using various algorithms. This approach uses GA for feature selection [9]. The input traffic with selected features is sent to the MIU for processing. The fusion unit collects all the local decisions using majority voting rule and makes the final decision. Selvakumar et al. proposed an IDS using filter and wrapper based methods [10]. This model uses firefly algorithm in selecting the features.

Abdullah et al. developed an IDS with the features selection approach. This model splits input samples into various subsets based on the type of attack [11]. The ideal features are selected by merging the list of subsets that are gained from the InfoGain filter. The simulation results reveal that the maximum accuracy is gained by PART and RF classification methods using product probability rule. Aryeh et al. proposed a multi-layer stack ensemble method to enhance classification accuracy of IDS [12]. The developed model takes the advantages of several rudiment classifiers to form a more robust and efficient meta-classifier. Seth et al. developed an IDS framework by utilizing hybrid feature selection and the light gradient boosting machine learning algorithm to provide better results with superior recall rate, accuracy, and low classification latency [13]. Sahu et al. proposed an ensemble-based approach using K-Means, decision tree, and RUSBoost methods to alleviate the class imbalance issue [14]. The authors analyze the effect of class imbalance on the performance of classification methods and compare the result with other approaches using effective performance measures including accuracy, F-measure, and correlation coefficient.

From the literature, most of the IDS frameworks disclose that dataset-oriented challenges are the principal reason for implementing optimization methods seems unfeasible. To circumvent overfitting the system to the data, small databases need frameworks that have low complexity or high bias. Hence, there is an urgent need to interpret the context prior to selecting a performance measure in light of the fact that each model attempts to solve an issue with a diverse objective function through various datasets. For example, most of the studies considered accuracy, recall and precision. On the other hand, recall and precision are efficient measures mostly in cases where classes are not uniformly dispersed. Motivated by the above-mentioned works, we develop an efficient and accurate IDS by integrating feature selection with an ensemble classification.

3 Description of Proposed Method

To improve the efficiency of the detection process and thwart the service providers from threats, we developed a IDS using BIOCFS and EC algorithm. Fig. 1 illustrates the architecture of the proposed IDS

framework which comprises three modules namely (i) feature selection, (ii) construct and train the ensemble classifier, and (iii) detection. For handling the problem of class imbalance, BIOCFS is applied to find the optimum subset from the dataset containing a very large number of features and remove unrelated features from the dataset. EC is selected after extensive analysis of assembling different learning approaches such as FPA, RF, and C4.5 to find both known and anonymous intrusive actions. The collaboration of FPA, RF, and C4.5 classifiers realizes better accuracy. Furthermore, Voting Mechanism is implemented to integrate the distribution of the probability of the simple learning algorithms for higher classification accuracy.

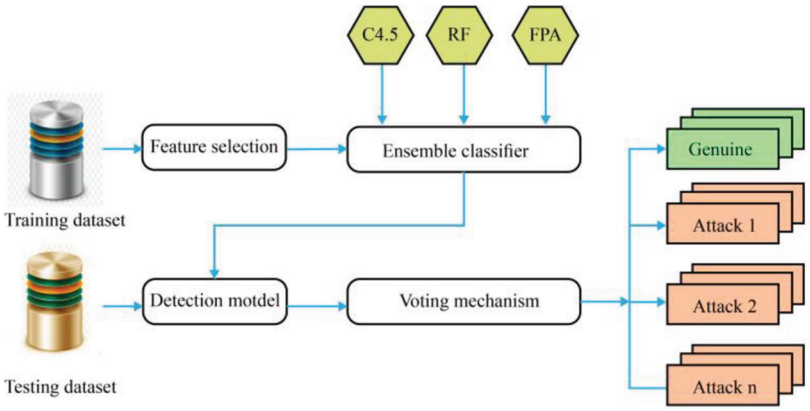

Figure 1: The architecture of the proposed BIOCFS-EC framework

3.1 Feature Selection

Contemporary datasets inevitably comprise of several redundant and unrelated features [15]. Redundant and unrelated features may decrease the efficiency of the IDS, producing uninterpretable results [16]. Hence, we need to decrease the size of the datasets and find the appropriate feature subset. This work proposes a hybrid method to enhance the performance of the feature selection and classification accuracy. The key idea of this method is to assess the redundancy and the relevance of the features and select the ideal subset in the particular search space.

3.1.1 Correlation-Based Feature Selection (CFS)

CFS selects subset based on correlations among pairs of features by means of assessment function (AF) [17]. The purpose of *AF* is to find subsets whose features are extremely correlated with the class but unrelated to each other. The features which indicate a feeble relationship with the class label have to be neglected and recurrent features are selected owing to a strong relationship with at least one of the rest. The selection of a feature will depend on the number of envisaged classes in sample space not up till now expected by various features. The assessment function (AF_{ss}) for a subset *ss* with *m* features is given in Eq. (1):

$$AF_{ss} = \frac{m\overline{d_{cm}}}{\sqrt{\overline{d_{mm}} + m + m(m-1)}} \tag{1}$$

where $\overline{d_{cm}}$ is the degree of average correlation between features and the class, and $\overline{d_{mm}}$ is the degree of the mean inter-correlation between features. A higher $\overline{d_{cm}}$ or lower $\overline{d_{mm}}$ in the result yields a higher assessment value, and the set of features with the maximum *AF* is employed to decrease the dimensionality of the dataset used for training and testing.

3.1.2 Bat-Inspired Optimization Algorithm

Bat Algorithm (BA) or Bat-inspired optimization algorithm was developed by Yang [18]. This algorithm is motivated by the echolocation nature of bats. Since this algorithm exploits frequency tuning, it is, indeed, the first algorithm based around the idea of computational intelligence and optimization. Let each bat arbitrarily flies with a velocity vel_i^t at position pos_i^t and frequency $freq_i^t$ in t time step in a d−dimensional space. The problem solution is symbolized by the bat position that can be defined by a vector. Amongst the bats in the population, for each iteration, the best solution Pos_{best} estimated hitherto can be stored. For every iteration, the value of pos_i^t and vel_i^t are updated using the method followed by Yang et al. [19] as given in Eqs. (2)–(4).

$$freq_i^t = freq_{min}^t + (freq_{max}^t - freq_{min}^t)\gamma \tag{2}$$

$$vel_i^t = vel_i^{t-1} + (pos_i^{t-1} - Pos_{best})freq_i^t \tag{3}$$

$$pos_i^t = pos_i^{t-1} + vel_i^t \tag{4}$$

We refer to $\gamma \in [0, 1]$ as the random vector obtained from a uniform distribution which serves as a parameter for frequency calculation. For each bat, a new solution is selected from the current best results by implementing a random walk approach as given in Eq. (5).

$$Pos_{new} = Pos_{old} + \delta L_i^t \tag{5}$$

We refer to $\delta \in [-1, 1]$ as a random vector derived from a Gaussian distribution or a uniform distribution and L_i^t is the mean loudness of all the bats at this iteration. For every iteration, along with the value of L_i^t the emission rate of pulses E_i^t also updated as given in Eqs. (6) and (7).

$$L_i^{t-1} = \rho L_i^t \tag{6}$$

$$E_i^{t+1} = E_i^0(1 - e^{-\alpha t}) \tag{7}$$

The parameters $\rho \in [0, 1]$ and $\alpha > 0$ are constants.

3.2 Bat-Inspired Optimization and Correlation-Based Feature Selection Algorithm

The proposed BIOCFS algorithm assesses the significance and the relationship between the identified feature subsets.

Algorithm 1: BIOCFS algorithm

Input: Datasets for training and testing phases

Output: Identified feature subset, Pos_{best}

1: Initialize the number of bats n in the population, $Pos_i = (pos_{i1}, pos_{i2,\ldots} pos_{iD,})^T$, $i = 1, 2, 3 \ldots n$ and vel_i^t

2: Initialize $freq_i^t$, E_i^t and L_i^t

3: Initialize $fit(Pos_i)$ and Pos_{best}

4: Initialize $fit_{temp}(i)$ and $Pos_{temp}(i)$ to store the result

5: while $1 \le t \le$ Max number of iterations do

6: for $1 \le t \le n$ do

7: Calculate new $freq_i^t$

(Continued)

Algorithm 1: (continued)

Update vel_i^t and pos_i^t
if $E_i^t <$ rand (0, 1) then
  Select a Pos_i from Pos_{best}
  Calculate Pos_{new}
 end if
 Calculate $fit(Pos_{new})$
 if($fit(Pos_i <(Pos_{new}))$ and N(0, 1) $< L_i^t$ then
  $fit_{temp}(i) \leftarrow fit(Pos_{new})$
  $Pos_{temp}(i) \leftarrow Pos_{new}$
  Increase E_i^t and decrease L_i^t
 end if
 if $fit(Pos_{new}) \geq$ Max ($fit_{temp}(i)$) then
   $Pos_{best} \leftarrow Pos_{new}$
 end if
 end for
 t = t + 1
 end while

To create the fitness functions and to estimate data integrity of the selected subset BIOCFS algorithm exploits correlation-based feature selection. For a given subset $ss = \{s_1, s_2, s_3 \ldots\ldots s_n\}$, correlation-based feature selection approach evaluates inter-correlation among features and the average correlation between class labels and features using Eq. (1). Conversely, this feature subset may not be the optimal solution due to uncorrelated features. Bat algorithm is used to eliminate the uncorrelated features and decrease the size of the datasets. In this algorithm, the position of the bat is considered as the solution to a problem of interest. Bats fly to find the best solution in the search space. During its movement, every bat finds and stores the best solution using Eqs. (2)–(4). The pseudocode for the proposed BIOCFS approach is given in Algorithm 1.

3.3 Ensemble Classifier (EC)

In EC approaches, several different, unbalanced and good classifiers are integrated in a specific way [20]. Ensemble classification approaches are used to handle the classification problem by implementing and integrating many autonomous classifiers [21]. Boosting [22] and Bagging [23] are the two most well-known approaches in collaborative learning, usually generating better classification solutions and being extensively selected to construct several ensemble frameworks. Besides, the other recognized collaborative learning approaches such as Stacking [24], Bayesian parameter averaging [25] and Voting [26] are used for increasing the efficiency of the classification process. This work focuses on an EC model that combines three different classifiers, namely C4.5, RF, and FPA to increase the predictability of IDS. These classifiers are employed to implement voting mechanism using the average of probabilities (AOP) rule.

C4.5 Decision Tree: It is one of the classic algorithms [27] developed to create a decision tree from a dataset using the Iterative Dichotomiser 3 (ID3) algorithm [28]. This algorithm finds the ideal split so as to maximize the gain ratio (GR) by visiting each and every node in the decision tree. The gain ratio is defined in Eq. (8).

$$GR(X) = \frac{G(X)}{Split_{data(X)}} \tag{8}$$

For classification, an attribute with the maximum GR is selected as a dividing attribute for the node. *Infogain* denotes how much indecision in the dataset *ds* is decreased after it is divided based on selected attribute *X*. The indecision in *ds* is calculated by entropy using Eq. (9).

$$Entropy(ds) = -\sum_{c \in C} prop(c) log_2\ prop(c) \tag{9}$$

where c is the set of classes in *ds* and $prop(c)$ is the ratio of the number of instances in class c and the number of instances in *ds*. Similarly, *Split_data* defines in what way the data are uniformly distributed by the attribute as defined in Eq. (10).

$$Split_{data(X)} = -\sum_{k=1}^{n} \frac{|ds_k|}{|ds|} log_2 \left(\frac{|ds_k|}{|ds|} \right) \tag{10}$$

where $\frac{|ds_k|}{|ds|}$ denotes the weight of k^{th} split in *ds*. In addition, the C4.5 algorithm can represent or categorize both continuous and discrete attributes and can neglect missing information.

Random Forest classifier: It is also a decision tree-based classification approach developed by Breiman [29]. It accepts a large number of input parameters without variable exclusion and categorizes them according to their reputation. RF classifier considers only fewer parameters as compared with the other machine learning methods (e.g., artificial neural network, support vector machine, etc.). In this classifier, a group of single tree classifiers can be denoted by Eq. (11).

$$\{r(x,\ \varphi_k) \qquad k = 1,\ 2,\ 3 \ldots . i \ldots\} \tag{11}$$

We refer to r as a function of RF classification. $\{\varphi_k\}$ represents random vectors and each tree has a prediction (vote) for the most popular class at input variable x. The characteristic and size of φ depend on its utilization. In order to train each tree in the forest, the RF classifier built a bootstrapped subset of the training process. Hence, each tree utilizes approximately 2/3 of the training dataset. The idle (out-of-bag) instances employ inner cross-validation process to calculate the classification accuracy. Furthermore, RF has the minimum computational overhead, and it is oblivious to the outliers and parameters. Also, the over-fitting issue is less as related to single decision tree-based approaches and it is not required to prune the tree which is a difficult and time-consuming process [30].

Forest by Penalizing Attributes: FPA exploits a subset of the non-class features and forms a group of decision trees with higher accuracy based on the strength of all non-class features existing in a dataset [31]. The weight allocation and weight augmentation policies are considered to preserve the accuracy and strong diversity. FPA will arbitrarily calculate the weights for features that present in the most recent tree. The weight range (W) can be calculated by Eq. (12).

$$W^{\omega} = \left\{ \begin{array}{ll} \left[0.00,\ \ e^{-\frac{1}{\omega}}\right], & \omega = 1 \\ \left[e^{-\frac{1}{\omega - 1} + \tau},\ e^{-\frac{1}{\omega}}\right], & \omega > 1 \end{array} \right\} \tag{12}$$

where ω denotes the attribute level and the parameter τ is employed to guarantee the weight range for various levels be non-overlying. When the feature presents in the root node, we select the value for ω equal to 1.

When the feature appears at a child, we select $\omega = 2$. Likewise, to describe the adverse impact of holding weights that does not exist in the most recent tree, FPA has a technique to progressively improve the attribute weights. Consider an attribute X_i is verified at level τ of the T_{k-1} th tree with height h and its weight is W_i. The increment of weight ∂_i is estimated by Eq. (13).

$$\partial_i = \frac{1 - W_i}{(h+1) - \omega} \tag{13}$$

Voting Mechanism: The estimation of each partaking classifier in the ensemble approach may be treated as a vote for a specific class, i.e., genuine or malicious [32]. It takes the advantages of many single classifier approaches and exploits a combination rule for making decisions. For example, maximum probability, minimum probability, average of probabilities, product of probabilities, and majority voting are used as combination rules. In this work, we apply an average of probabilities method to make a decision. The class label is designated according to the maximum value of AOP. Let m denotes the number of classifiers $C = \{C_1, C_2, \ldots C_m\}$ with c classes $\Omega = \{\Omega_1, \Omega_2, \ldots \Omega_c\}$. In our experiment, we choose $c = 15$ and $m = 3$. A classifier $C_i{:}R^n \rightarrow [0, 1]^c$ gets an input instance $x_i \in R^n$ and gives the output as a vector $\rho_{ci}(W_1|x)$, $\rho_{ci}(W_2|x)$, $\ldots \rho_{ci}(W_c|x)$, where $\rho_{ci}(W_k|x)$ represents the probability allocated by C_i that input instance x fits into class W_k. Let AV_k is the AOP allocated by the classifiers for each class. It can be estimated by Eq. (14).

$$AV_k = \frac{1}{m} \sum_{i=1}^{m} \rho_{ci}(W_k|x) \tag{14}$$

Consider $AV = [av_1, av_2, \ldots av_c]$ is the set of AOPs for c classes and x is allocated to the weight W_c if AV_k is having higher value in AV.

4 Description of Proposed Method

In this work, we integrate bat-inspired optimization and correlation-based feature selection with an ensemble classifier to find an optimal subset. EC is used to integrate FPA, RF and C4.5 classifiers by applying the AOP rule. EC is trained and tested on NSL-KDD traffic traces. Experimental results illustrate that the combined BIOCFS-EC algorithm outdoes every individual classifiers by realizing higher classification efficiency. The experiments are carried out on 3.6 GHz with 16 GB RAM, Intel Core i7-4790 processor using Weka 3.8.3 tool [33].

4.1 Dataset to Model Traffic Flow

Procuring a real-time database that reflects the data transmission over a network without any kind of modification or anonymization is an issue that has been constantly addressed by several researchers [34]. In some applications, the information is permitted to be shared or released for public use. Hence, it will be severely unidentifiable or rigorously modified. Therefore, several investigators preferred simulated datasets such as NSL-KDD [35]. Motivated by this, we plan to use samples taken from the NSL-KDD dataset. Every connection records consist of 41 features as shown in Tab. 1.

The features in this dataset are categorized into three types as fundamental, content, and traffic features calculated with two seconds time windows [36]. Although this database is slightly older and there have been few studies pointing out its flaws [37], it is still considered as benchmark dataset and employed by topical research in various field of engineering domains [8,38]. The attack instances in the dataset are classified into four groups based on their probability distributions as follows.

Table 1: Features of TCP connection records in NSL-KDD dataset

Type	Features
Discrete	Protocol_type, service, flag , land, logged_in, is_host_login, is_guest_login
Continuous	Duration, src_bytes, dst_bytes, wrong_fragment, urgent, hot, num_failed_login, num_compromised, root_shell, su_atempted, num_root, num_file_creations, num_shells, num_access_files, num_outbound_cmds, count, srv_count, serror_rate, srv_serror_rate, rerror_rate, srv_rerror_rate, same_srv_rate, diff_srv_rate, srv_diff_host_rate, dst_host_count, dst_host_srv_count, dst_host_diff_src_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_srv_serror_rate, dst_host_rerror_rate, dst_host_srv_rerror_rate

- DoS: It is a threat wherein the intruder makes memory or processing resource excessively full or very busy to accept genuine requests, or rejects approved users entry to a system.
- Probing: It is an effort to gather details about nodes in a system in order to change the state of the network security.
- Remote to Local (R2L)–In this attack, weaknesses of a system permit an intruder to gain access locally an approved user account without having their account.
- User to Root (U2R)-It is an attack where the intruder accesses the network as an authorized user using a legitimate user account (possibly achieved by social engineering, a dictionary attack, or sniffing passwords). The attacker is competent to use some weaknesses to achieve root access to the system.

Though NSL-KDD preserved the useful and stimulating characteristics, it addressed some downsides including preservation of the diversity of identified samples, addition of a more rational amount of samples, and the removal of redundant records. Particularly, the main physiognomy of NSL-KDD is that it was assembled to increase the difficulty level of prediction. Several standard classifiers are used to assess the initial dataset. Every record in the dataset is marked with the number of its fruitful predictions and difficulty levels of predictions [39]. The following Tab. 2 gives some examples of the attacks found in the NSL-KDD dataset.

Table 2: Example of attacks in NSL-KDD

Class	Example attacks
DoS	Neptune, smurf, back, teardrop, pod, land
Probing	Ipsweep, satan, portsweep, nmap,
R2L	Warezclient, guess_passwd, warezmaster, multihop, spy, phf, ftp_write, imap
U2R	Buffer_overflow, rootkit, loadmodule, perl,

4.2 Data Preprocessing

Real-world dataset often incomplete, redundant, inconsistent, noisy, and/or missing certain trends or behaviors [40]. Therefore, it is essential to convert raw data into an appropriate format suitable for knowledge discovery and evaluation. In this work, we employ preprocessing for eliminating redundant features and outliers (filtration), transforming and data normalization.

- **Filtration:** The real-world raw data from heterogeneous platforms unsurprisingly comprises abnormal and redundant features which may have an adverse impact on the accuracy of the classifier. To combat this challenge, abnormal and redundant records must be excluded from the dataset.

- **Transforming and normalization:** NSL-KDD dataset encompasses discrete, continuous, and symbolic values. Since most of the classifiers recognize only numerical values, the transformation process is essential and this process has a considerable effect on the performance of IDS. In our work, we assign numerical values for every single symbolic feature. Furthermore, different scales among features can reduce the performance of the classifier, for instance, features with large numeric values. Therefore, normalization is considered as a 'scaling down' transformation process. It relates every feature to a standardized range. In this work, we use a simple and fast technique known as a min-max technique [41] for normalization. The min-max technique is defined by Eq. (15).

$$\overline{\mu} = \frac{\mu - \mu_{min}}{\mu_{max} - \mu_{min}} \tag{15}$$

where μ_{min} and μ_{max} are the minimum and maximum values of feature μ.

Indeed, the intrusions do not typically befall as frequent as genuine traffic [42,43]. Conversely, the proportion of abnormal to genuine instances is a major problem that can considerably impact the performance of the training and learning phases of our experiments. To derive a balanced dataset with genuine/malicious instances, we retrieve some instances and confirm that the amount of anomalous instances is approximately equal to that of genuine instances for training and testing processes. In the chorus, we ensure that a single instance does not repeat in both subsets, which assures the correctness of the training process and causes the maximum accuracy in the testing phase.

The description of the records used in the training and testing processes of our study is given in Tabs. 3 and 4. The number of records for each class in the dataset is given in Fig. 2.

Table 3: Distribution of attacks in the dataset with 72900 records

Class	Attack	No. of records	Class	Attack	No. of records	Class	Attack	No. of records	Class	Attack	No. of records
DOS	Neptune	20760	Probing	Ipsweep	1812	Remote to local	Warezclient	473	User to root	Buffer_overflow	18
	Smurf	1350		Satan	1746		Guess_passwd	32		Rootkit	8
	Back	504		Portsweep	1399		Warezmaster	12		Loadmodule	2
	Teardrop	448		Nmap	756		Multihop	6		Perl	1
	Pod	82					Spy	2			
	Land	16					Phf	4			
							Ftp_write	5			
							Imap	7			

Table 4: Statistics of subsets NSL-KDD with 72900 records

Class	NSL-KDD	
	Training	Testing
Normal	43457	43457
DoS	22907	23160
PRB	5713	5713
R2L	797	541
U2R	26	29
Total attack	29443	29443
Total	72900	72900

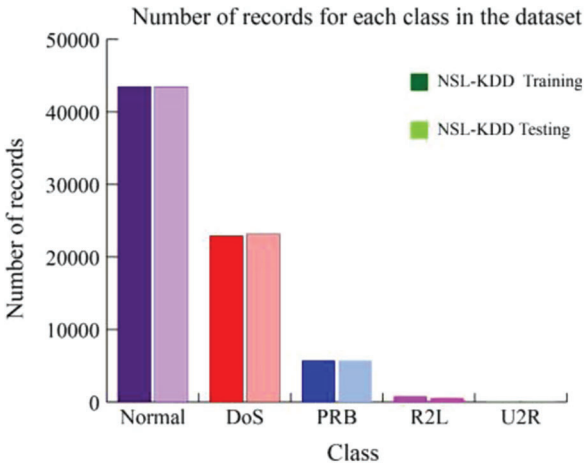

Figure 2: Number of records for each class in the dataset

5 Results and Discussion

The efficiency of the proposed IDS is assessed according to its competence in categorizing input samples into an appropriate class. The proposed IDS has been appraised by training and testing subsets of the NSL-KDD database. More precisely, for given dataset, we derive the confusion matrix during the testing phase of BIOCFS-EC approach, and relate the efficiency of our approach without implementing any feature selection process and also some related feature selection approaches with respect to performance measures such as classification accuracy (CA), precision (PR), F-measure (FM), false alarm rate (FAR), and attack detection rate (ADR). The methods of estimation of performance measures are derived from [43]. First, the important features are identified using our BIOCFS algorithm. Then, the representative features are retrieved from the intact dataset for the subsequent phases. Tab. 5 illustrates the name of the identified candidate features. It is easily observed that the size of the dataset is decreased considerably by applying the proposed BIOCFS approach. Finally, in order to increase the classification accuracy of IDS, an efficient ensemble classifier is used with a voting mechanism.

Table 5: Selected features

Name of the feature	Service, flag, src bytes, dst bytes, root shell, is host login, serror rate, same srv rate, diff srv rate, dst host srv diff host rate.

Fig. 3 shows the results obtained from the classifier of our IDS. It is found that the enactment of the IDS is acceptable; however, a small number of attacks cannot be identified well, for example, 'R2L' and 'U2R'. Furthermore, the BIOCFS approach is not considered the records of a particular class; it is projected for identifying features from the entire dataset, which could not assure the enactment of each type of intrusion. Conversely, since the results obtained for genuine records are very good, the proposed approach can be employed to detect intrusion.

5.1 Performance Evaluation Related to No Feature Selection

To assess the effectiveness of the proposed IDS, we compare BIOCFS approach to other approaches without using feature selection. It is observed that the performance measures of the proposed IDS including CA, PR, FM, FAR and ADR are increased considerably as compared with other approaches. Tab. 6 summarizes the results obtained from different classifiers on the identified dataset, However, the

performance of the EC is not enhanced in terms of some measures (e.g., FAR) without applying the feature selection process. Figs. 4–6 illustrate the performance of our EC approach related to the other classifiers in terms of CA, PR, FM ADR, FAR and time required for training and testing.

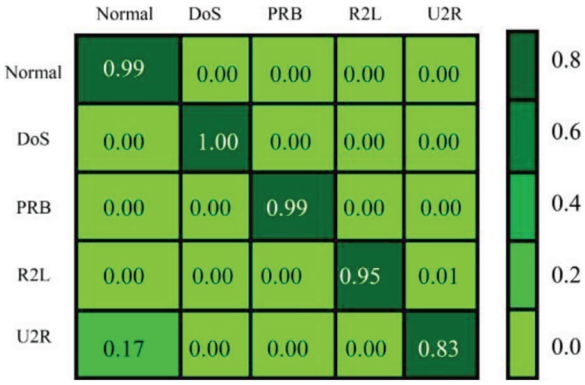

Figure 3: Confusion matrix

Table 6: Performance measures for original features in the dataset (41 features)

Classifier	C 4.5	RF	FPA	EC
CA	0.941	0.951	0.951	**0.962**
PR	0.971	0.932	0.962	**0.992**
FM	0.950	0.963	0.963	**0.983**
ADR	0.951	0.922	0.931	**0.954**
FAR	0.070	**0.011**	0.031	0.055
Time for training (s)	1.56	10.65	35.57	39.43
Time for testing (s)	0.19	2.15	0.23	2.25

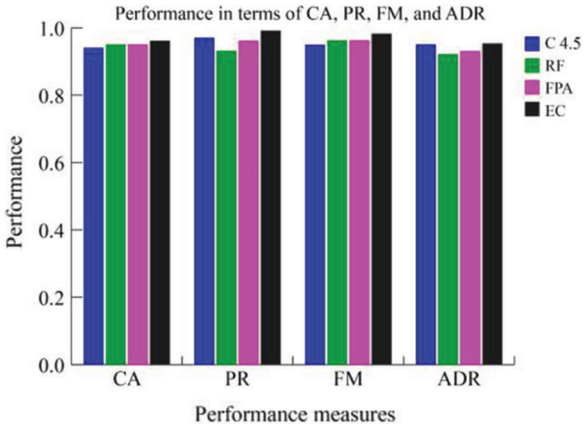

Figure 4: A comparison of the performance measures CA, PR, FM, and ADR

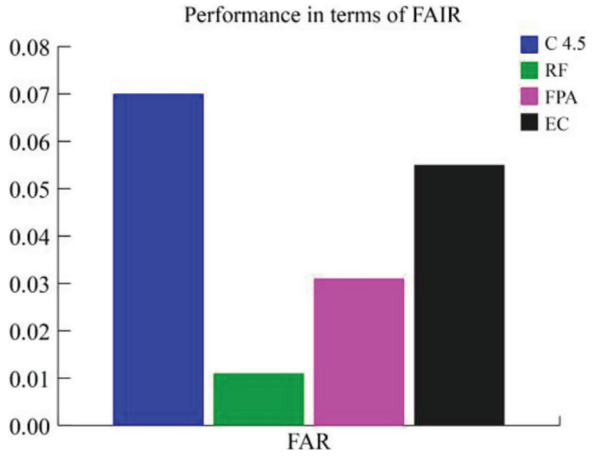

Figure 5: A comparison of the false alarm rate for different classifiers

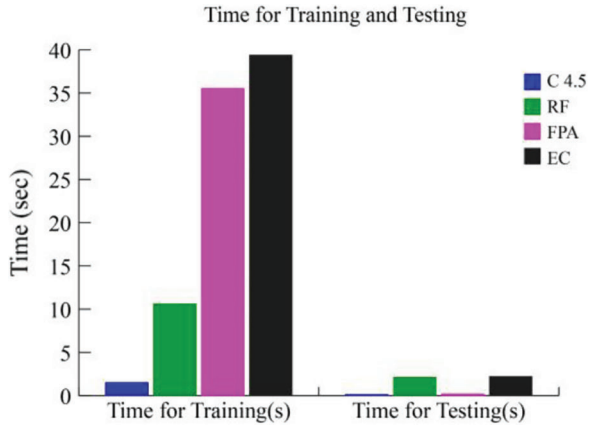

Figure 6: A comparison of running time required for training and testing

From the results shown in this table, our EC approach reveals a greater accuracy (96.2%), precision (99.2%), FM (98.3%), ADR (95.4%), FAR (5.5%) and time required for training and testing (39.43 and 2.25 sec), respectively as compared to the other classification systems. The proposed algorithm consumes more time for training and testing. Hence, we need to reduce this factor by implementing proper feature selection methods. Fig. 5 also illustrates the effectiveness EC approach in detecting intrusion.

5.2 Performance Evaluation Related to Feature Selection

As discussed earlier, the standard datasets reflect an existing and multifaceted attack scenario. Handling several classes of attacks and extremely unbalanced instances is a difficult task for any machine learning technique. In order to further appraise our approach, we relate its performance to some distinguished feature selection approaches found in the literature. In this comparative analysis also, we utilize CA, PR, FM, FAR and ADR as the performance metrics. Tab. 7 shows the obtained evaluation metrics for the reduced set of features (10 features). The proposed BIOCFS-EC approach decreases the time consumption of training and testing procedures from 39.43 and 2.25 s to 16.66 and 1.28 s, respectively. Also, the proposed approach achieves the highest classification accuracy of 0.994, precision of 0.993, F-measure of 0.992, the attack detection rate of 0.992 and lowest false alarm rate of 0.008% on the dataset.

Table 7: Estimated performance measures for the selected features (10 features)

Classifier	C 4.5	RF	FPA	BIOCFS-EC
CA	0.987	0.988	0.987	**0.994**
PR	0.988	0.991	0.989	**0.993**
FM	0.988	0.989	0.989	**0.992**
ADR	0.985	0.988	0.987	**0.992**
FAR	0.012	0.011	0.013	**0.008**
Time for training (s)	0.27	4.70	14.28	16.66
Time for testing (s)	0.09	1.20	0.16	1.28

Figs. 7–9 illustrate the performance of BIOCFS–EC model as compared to the other approaches in terms of CA, PR, FM ADR, FAR and time required for training and testing. Moreover, due to the reduced dimension of the subsets, our BIOCFS-EC approach decreases the time complexity when it is used for feature selection. Based on Tab. 7, the proposed feature selection approach BIOCFS with ensemble classifier has reduced the training and testing time significantly related to other approaches.

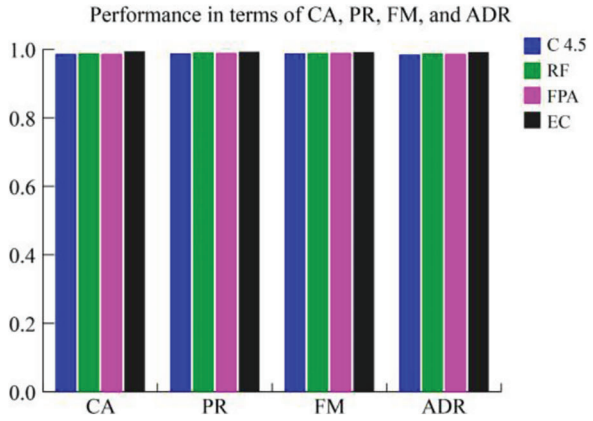

Figure 7: A comparison of the performance measures CA, PR, FM, and ADR

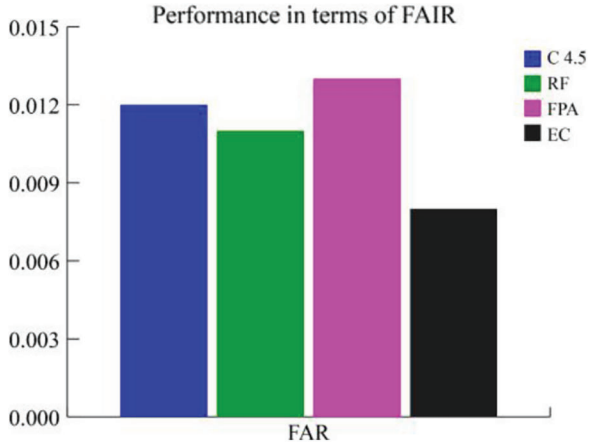

Figure 8: A comparison of the false alarm rate for different classifiers

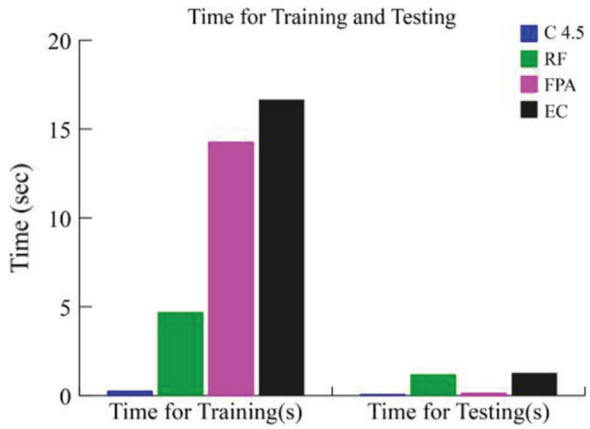

Figure 9: A comparison of running time required for training and testing

6 Conclusions

An IDS is extensively used to identify attacks and to achieve integrity, confidentiality, and availability of sensitive information. Though several unsupervised and supervised machine learning methods have been employed to improve the efficiency of the IDS, it is still a challenge to handle several redundant and unrelated information in big data scenarios. In this work, we develop an IDS using machine learning techniques to improve the performance of attack detection. In order to cope with high dimensional feature-rich traffic in large networks, we introduce a Bat-Inspired Optimization and Correlation-based Feature Selection algorithm and an ensemble classification approach. The integration of these approaches aid to handle multi-class and unbalanced datasets. The competence of proposed IDS is evaluated on a well-known dataset NSL-KDD. The experimental results reveal that our combined approach outperforms other state-of-the-art approaches in terms of classification accuracy, precision, F-measure, false alarm rate, and attack detection rate. One of the key issues of the proposed IDS is the relatively high computational overhead due to the incomplete information, isolated features and redundant contents in the IDS datasets. To handle such problems and ensure creating effective and more precise IDS frameworks, we plan to apply appropriate preprocessing technique with our proposed algorithm for developing the IDS with high predictive ability.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Qadir and S. M. K. Quadri, "Information availability: An insight into the most important attribute of information security," *Journal of Information Security*, vol. 7, no. 3, pp. 185–194, 2016.

[2] O. Joldzic, Z. Djuric and P. Vuletic, "A transparent and scalable anomaly-based DoS detection method," *Computer Networks*, vol. 104, pp. 27–42, 2016.

[3] D. Papamartzivanos, F. G. M´armol and G. Kambourakis, "Dendron: Genetic trees driven rule induction for network intrusion detection systems," *Future Generation Computer Systems*, vol. 79, no. 2, pp. 558–574, 2018.

[4] J. Kim, J. Kim, H. L. T. Thu and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in *Proc. 2016 Int. Conf. on Platform Technology and Service (PlatCon)*, Jeju, Korea (South), pp. 1–5, 2016.

[5] H. Hota and A. K. Shrivas, "Decision tree techniques applied on NSL-KDD data and its comparison with various feature selection techniques," *Advanced Computing, Networking and Informatics*, vol. 1, pp. 205–211, 2014.

[6] A. J. Malik, W. Shahzad and F. A. Khan, "Network intrusion detection using hybrid binary PSO and random forests algorithm," *Security and Communication Networks*, vol. 8, no. 16, pp. 2646–2660, 2015.

[7] N. Paulauskas and J. Auskalnis, "Analysis of data pre-processing influence on intrusion detection using NSL-KDD dataset," in *Proc. 2017 Open Conf. of Electrical, Electronic and Information Sciences (eStream)*, IEEE, Vilnius, Lithuania, pp. 1–5, 2017.

[8] C. Khammassi and S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," *Computers & Security*, vol. 70, pp. 255–277, 2017.

[9] K. Jayakumar, T. Revathi and S. Karpagam, "Fusion of heterogeneous intrusion detection systems for network attack detection," *The Scientific World Journal*, vol. 2015, pp. 314601, 2015.

[10] B. Selvakumar and K. Muneeswaran, "Firefly algorithm based feature selection for network intrusion detection," *Computers & Security*, vol. 81, pp. 148–155, 2019.

[11] M. Abdullah, A. Balamash, A. Alshannaq and S. Almabdy, "Enhanced intrusion detection system using feature selection method and ensemble learning algorithms," *International Journal of Computer Science and Information Security*, vol. 16, no. 2, pp. 48–55, 2018.

[12] F. L. Aryeh and B. K. Alese, "A Multi-layer stack ensemble approach to improve intrusion detection system's prediction accuracy," in *Proc. 2020 15th Int. Conf. for Internet Technology and Secured Transactions (ICITST)*, London, United Kingdom, pp. 1–6, 2020.

[13] S. Seth, G. Singh and K. K. Chahal, "A novel time efficient learning-based approach for smart intrusion detection system," *Journal of Big Data*, vol. 8, no. 111, pp. 1–28, 2021.

[14] S. K. Sahu, D. P. Mohapatra, J. K. Rout, K. S. Sahoo and A. K. Luhach, "An ensemble-based scalable approach for intrusion detection using big data framework," *Big Data*, vol. 9, no. 4, pp. 303–321, 2021.

[15] N. Acharya and S. Singh, "An IWD-based feature selection method for intrusion detection system," *Soft Computing*, vol. 22, no. 13, pp. 4407–4416, 2018.

[16] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Applied Sciences*, vol. 9, no. 20, 4396, 2019.

[17] S. Singh and A. K. Singh, "Detection of spam using particle swarm optimisation in feature selection." *Pertanika Journal of Science & Technology*, vol. 26, no. 3, pp. 1355–1372, 2018.

[18] X. -S. Yang, "Nature-inspired optimization algorithms: Challenges and open problems," *Journal of Computational Science*, vol. 46, pp. 101104, 2020.

[19] X. -S. Yang and X. He, "Bat algorithm: Literature review and applications," *International Journal of Bio-Inspired Computation*, vol. 5, no. 3, pp. 141–149, 2013.

[20] X. Feng, Z. Xiao, B. Zhong, J. Qiu and Y. Dong, "Dynamic ensemble classification for credit scoring using soft probability," *Applied Soft Computing*, vol. 65, pp. 139–151, 2018.

[21] H. Li and J. Sun, "Predicting business failure using an RSF-based CASE-based reasoning ensemble forecasting method," *Journal of Forecasting*, vol. 32, no. 2, pp. 180–192, 2013.

[22] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. Int. Conf. on Machine Learning*, Bari, Italy, vol. 96. pp. 148–156, 1996.

[23] B. Pes, "Ensemble feature selection for high-dimensional data: A stability analysis across multiple domains," *Neural Computing and Applications*, vol. 32, pp. 5951–5973, 2020.

[24] C. Hung and J. -H. Chen, "A selective ensemble based on expected probabilities for bankruptcy prediction," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5297–5303, 2009.

[25] K. Friston, K. Stephan, B. Li and J. Daunizeau, "Generalised filtering," *Mathematical Problems in Engineering*, vol. 2010, Article ID 621670, 2010.

[26] J. Hu, "An approach to EEG-based gender recognition using entropy measurement methods," *Knowledge-Based Systems*, vol. 140, pp. 134–141, 2018.

[27] J. R. Quinlan, "C4. 5: Programs for machine learning," in *Morgan Kaufmann Series in Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2014.

[28] B. Hssina, A. Merbouha, H. Ezzikouri and M. Erritali, "A comparative study of decision tree ID3 and C4. 5," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 2, pp. 13–19, 2014.

[29] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[30] Q. Feng, J. Liu and J. Gong, "UAV remote sensing for urban vegetation mapping using random forest and texture analysis," *Remote Sensing*, vol. 7, no. 1, pp. 1074–1094, 2015.

[31] M. N. Adnan and M. Z. Islam, "Forest pa: Constructing a decision forest by penalizing attributes used in previous trees," *Expert Systems with Applications*, vol. 89, pp. 389–403, 2017.

[32] C. Catal and M. Nangir, "A sentiment classification model based on multiple classifiers," *Applied Soft Computing*, vol. 50, pp. 135–141, 2017.

[33] I. H. Witten, E. Frank, M. A. Hall and C. J. Pal, "Data mining: Practical machine learning tools and techniques," in *The Morgan Kaufmann Series in Data Management Systems*, 3rd ed., Morgan Kaufmann Inc., Burlington, Massachusetts, 2011.

[34] T. Aldwairi, D. Perera and M. A. Novotny, "An evaluation of the performance of restricted boltzmann machines as a model for anomaly network intrusion detection," *Computer Networks*, vol. 144, pp. 111–119, 2018.

[35] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. 2009 IEEE Symp. on Computational Intelligence for Security and Defense Applications*, Ottawa, ON, Canada, pp. 1–6, 2009.

[36] W. Lee, S. J. Stolfo and K. W. Mok, "A data mining framework for building intrusion detection models," in *Proc. 1999 IEEE Symp. on Security and Privacy*, Oakland, CA, USA, pp. 120–132, 1999.

[37] C. Luo, L. Wang and H. Lu, "Analysis of LSTM-RNN based on attack type of KDD-99 dataset," in *Proc. 2018 Int. Conf. on Cloud Computing and Security*, Haikou, China, pp. 326–333, 2018.

[38] R. Bala and R. Nagpal, "A review on KDD cup99 and NSL NSL-KDD dataset." *International Journal of Advanced Research in Computer Science*, vol. 10, no. 2, pp. 64–67, 2019.

[39] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang *et al.,* "Feature selection: A data perspective," *ACM Computing Surveys*, vol. 50, no. 6, pp. 1–45, 2018.

[40] S. Kotsiantis, D. Kanellopoulos and P. Pintelas, "Data preprocessing for supervised leaning," *International Journal of Computer Science*, vol. 1, no. 2, pp. 111–117, 2006.

[41] M. Tavallaee, N. Stakhanova and A. A. Ghorbani, "Toward credible evaluation of anomaly-based intrusion-detection methods," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 5, pp. 516–524, 2010.

[42] H. Kwon, Y. Kim, H. Yoon and D. Choi, "Optimal cluster expansion-based intrusion tolerant system to prevent denial of service attacks," *Applied Sciences*, vol. 7, no. 11, pp. 1186, 2017.

[43] S. Elhag, A. Fern´andez, A. Altalhi, S. Alshomrani and F. Herrera, "A Multi-objective evolutionary fuzzy system to obtain a broad and accurate set of solutions in intrusion detection systems," *Soft Computing*, vol. 23, no. 4, pp. 1321–1336, 2019.