An Enhanced Intrusion Detection System Based on Multi-Layer Feature Reduction for Probe and DoS Attacks

M. El-Shrkawey[1], Marwa Alalfi[1], and Hassan Al-Mahdi[2*]
[1]Suez Canal University, Information System Dept, Ismailia 41522, Egypt
melshrkawey@ci.suez.edu.eg, marwa.alalfi@gmail.com
[2]Suez Canal University, Computer Science Dept., Ismailia 41522, Egypt
drhassanwesf@ci.suez.edu.eg

Received: July 10, 2021; Accepted: October 27, 2021; Published: November 30, 2021

Abstract

Wireless network has an exponential increase in various aspects of the human community. Accordingly, transmitting a vast volume of sensitive and non-sensitive data over the network puts them at risk of being attacked. To avoid this, Intrusion Detection System (IDS) security is intended to detect threats and protect devices from attacks. IDS usually uses one of the following alternative approaches: signature-based, anomaly-based, or hybrid of the two. In spite of the IDS has been the focus of much research in recent years, there is still space for improvement. Based on the anomaly-based approach, this paper proposes a modified algorithm called a Multi-layer Feature Selection and Reduction IDS (MFSR-IDS) for providing high-level protection against Denial-of-Service (DoS) and Probe attacks. The MFSR-IDS framework makes three major contributions. First, it reduces the feature dimensionality of the network dataset across three layers. Second, it has a fast and accurate detection system. Third, it provides a mathematical model of the framework under consideration. The MFSR-IDS algorithm selects optimal number of features from KDDCUP'99 dataset which used to train the predictive model based on different learning classifiers and ensemble methodology. The performance of MFSR-IDS is evaluated in terms of Detection Rate (DR), False Positive Rate (FPR), FScore, ROC area, Accuracy (Acc) and Processing time. The experiments indicate that, the proposed MFSR-IDS outperforms some existing IDS frameworks in terms of DR, FPR, Acc and Processing time in detecting DoS and Probe attacks.

Keywords: Intrusion Detection System, Anomaly Based Detection, KDDCUP'99 Dataset, Feature Selection

1 Introduction

Recently, the massive revolution in smart technologies causes explosive growth in the transmission of digital data through the internet [1, 2]. As a result, data flow through diverse device technologies has become vulnerable to various types of malicious intrusions [3]. These types of intrusions are classified into four categories of attacks. Namely, Denial of Service (DoS), User to Root (U2R), Remote to Local (R2L), and Probe [4, 5].

In a DoS attack, the attacker consumes computing resources by causing an unwanted overload, preventing legitimate users from completing their duties on time [6]. For the U2R attack, the attacker detects the weaknesses of the victim's device to have local access to his machine and obtain his administration

Journal of Internet Services and Information Security, volume: 11, number: 4 (November), pp. 61-78
DOI:*10.22667/JISIS.2021.11.30.061*
*Corresponding author: Department of Computer Sciences, Suez Canal University, Ismailia 41522, Egypt, Tel: +20-01025373771

privileges. In the R2L, an attacker tries to acquire access right to a remote system without having its account. In Probe attack, the attacker aims to scan networking devices to gather the required information about the vulnerabilities in the system. Later, this information may be exploited in order to compromise the system.

These types of intrusions caused increases in the risks and vulnerabilities of the networks. So, the privacy and security services of the saved and transmitted data became more susceptible to be stolen or destroyed [7]. Therefore, intrusion detection systems (IDS) are offered as efficient security measures employed to improve the security of the information during its transmission. Furthermore, IDS seeks to prevent or at least react to potential intrusions by hovering an alarm when the network is exposed to attacked [8].

The IDS aims to monitor and analyze the statistics of networks' activities in order to detect the intrusion's events obscured among the normal activities of the network. So, the feature extraction from the whole features of the network traffics is defined as a vital challenge in IDS [9]. The main objective of the features extraction is to minimize the dimensionality of the selected feature without losing the information in the feature space [10]. Hence, feature extraction is an imperative process in data mining and machine learning to decrease the numbers of the selected feature [11].

The IDSs are divided into two main categories: Signature-based and anomaly-based detection systems [12]. The signature-based detection classifies malevolent intrusions based on a pattern for certain activities that has been determined in advance. These intrusion detection systems need an updated database to store the innovative intrusion signatures. In anomaly-Based Detection, on the other side, the normal activities of the traffic system are identified as the normal behaviour of the traffic. So, the occurrences of any abnormal activities motivate AIDS to employ the proper countermeasures for reacting to these attacks. In addition, it submits reports to an administrator or security team [13].

To improve the accuracy rate of intrusion detection, data mining was used to implement the IDS. In addition, it was exploited to evolve the efficient classification and clustering methods for discriminating between any abnormal threats or intrusions and the normal behaviour of the network activities [14].

However, the different data mining operations can be managed through two main stages, namely, data preprocessing and data mining [15]. In the data preprocessing stage, the data set is prepared through different methods which include data transferring, data resampling, and various filtration methods [12]. Consequently, the data set becomes applicable to the next step which deals with the features' selection. The feature minimization enables the model to be trained efficiently and leads to better accuracy as well as fast processing time [16].

In data mining, various algorithms are implemented to filter and extract the most significant features from the huge dataset. It applies various classification techniques of data mining that belong to the category of supervised learning. In this category, the input data provided for the training dataset is labelled to utilizes the reduced features in order to understand the behaviour of the normal data. Hence, as the classifier was trained accurately, it can be used to detect abnormal behaviour efficiently [16].

The available IDS techniques applied for detecting unauthorized attacks do not provide 100% of accuracy. As a result, there is still room for improvement. So, this paper proposes a modified framework called a Multi-layer Feature Selection and Reduction IDS (MFSR-IDS). The MFSR-IDS is performed based on the Anomaly Based Detection to detect both types of the attacks mentioned before, namely DoS and Probe [17]. The MFSR-IDS uses the KDDCUP'99 dataset which is extensively used for the evaluating IDS systems [18]. The main contributions of the proposed MFSR-IDS are as follows:

- Minimizing the number of effective and operative features during the data pre-processing, ranking and classifying stages which represents the three reduction layers.
- Increasing the IDS system accuracy and reducing processing time by using classification techniques namely C4.5, Naïve Bayes (NB), RepTree (RT) and Random Forest (RF) as well as inte-

grating different classifiers using ensemble-based algorithms such as Bagging (Bootstrap aggregation).

- Introducing a general mathematical framework to describe the different phases of the IDS.

In order to accomplish contribution's objective, the MFSR-IDS operation is carried out in a three consecutive phases. Namely, preprocessing, feature selection and feature classifier. The preprocessing phase aims to drop the content features that are not highly ranked when employed with the multiple ranking filters. So, the preprocessing phase initiates its operation by dividing Dataset's features into different groups. Next, the features are resampled and labeled features are replaced by numeric values. Finally, the redundant and inappropriate features are removed for efficient classifications that are performed to extract the most effective features.

The features selection (FS) phase is a vigorous and central phase exploited to improve our learning model to rise accuracy and reducing training time. In addition, FS will lead to robust learning against noise and overfitting. Finally, the dataset in the feature classifier phase is performed through the backward elimination technique which performs its operations through successive iterations. In the first iteration, it initiates with the whole features while the least significant feature will be removed before leaving each iteration. The iterations will proceed until there is no improvement when features have been removed. This technique leads to improve the accuracy of the model.

The remainder of our paper is laid out as follows: The IDS-related work is introduced in Section 2. The suggested MFSR-IDS description, research methodology are described in detail in section 3. Besides, the dataset processing, feature selection and classifier are depicted. In section 4, we give a brief description of KDDCUP'99 dataset, conduct experiments using Weka software and finally discuss the obtained results. The conclusion and future work are illustrated in the last section.

2 Related Work

A great IDS research has been carried in recent years to solve the intrusion's detections problems in the different security systems. Actually, feature selection has a significant effect on the classification and training phase of machine learning IDS. To increase the efficacy of IDS models in detecting various types of attacks, many strategies and models have been investigated and developed [19].

In [20], tree-based algorithms were employed to create a combining classifier model to detect networks' intrusion. This model was implemented through the NSL-KDD dataset. The experiments show that the combination of random tree and Naive Bayes Tree are providing higher detection accuracy rates than the individual random tree algorithm results. In [21], information gain and correlation-based are used as filters to select the most significant features among the 41 features in the KDD'99 dataset. In addition, the wrapper methods (RF, C4.5, NB, and RepTree are implemented with all 41 features for classification purposes. The supervised machine learning algorithms C4.5, Naïve Bayes, Random Forest and RepTree are used as wrappers to build the prediction model. The model was built to effectively detect Probe and DoS attacks.

The authors in [22] presented an adaptive ensemble learning algorithm based on the classifiers decision tree, support vector machines, logical regression, k-nearest neighbors, adaboost, random forest, and deep neural network classifiers. By regulating the percentage of training data, and initiating multiple decision trees, a multi-tree algorithm was applied to acquire the optimal detection effect.

The authors in [23] proposed an ensemble IDS based on six rankers for feature selection. Namely, gain ratio, information gain, symmetrical uncertainty, Relief-F, One-R, and chi-square. Each ranker determines its feature subset through four different classification algorithms. They are decision tree:

J48, SOM, Bayesian network, and naïve bayes. The main objective of this ensemble IDS is to select the most important features to increase the ability of attack detection.

A model of information gain ranker with feature reduction was suggested by the authors in [24]. In addition, the operations of the classifications are performed based on a hybrid approach of K-nearest neighbour, random tree, Rep tree, J48, and random forest classifiers. The implementation of this model offered an enhancement in the accuracy and false-positive rate detection than the other conventional classification techniques. A hybrid approach of the IG-PCA method was proposed in [30]. In this

Reference	Attacks	Classifiers	Dataset	Pros	Cons
Ref. [25]	• DoS, Probe • R2L, U2R	J48, NB	KDD'99	• Fast detection	• NB is impractical with large datasets.
Ref. [26]	• DoS, Probe • R2L, U2R	Ensemble (CVM)	NSL-KDD	• High detection rate • Less false positive • Low computation overhead	• Poor results for R2L attacks • Problematic of determining optimal kernels • Problematic parameters of kernel function
Ref. [16]	• Binary • Multiple class	enhanced J48	NSL-KDD	• Improve detection • Improve accuracy	• The processing time is not considered
Ref. [27]	• DoS, Probe • R2L, U2R	random forest	NSL-KDD	• Improve accuracy • Improve detection rate	• IG Limitations for feature selection. • Hard to analyse RF output. • Implementation of RF is ineffective for real-time predictions
Ref. [21]	• DoS • Probe	• C4.5, NB • RepTree, RF	KDD'99	• false positive rate • Improve detection rate • Processing time is optimized	• IG threshold value is not determined. • No reduction algorithm applied for feature selection. • k-folds cross validation is not applied which causes over-fitting problem
Ref. [24]	• Binary • Multiple class	• Ensemble (IBk, RandomTree, REPTree, j48graft, RF)	NSL-KDD	• Minimize time complexity. • Minimize computational cost. • Improve accuracy. • Lowering false positive rate	• Challenging of obtaining the effective features due to difficulty of choosing the optimal K-value. • sensitivity of determining IBK similarity function.
Ref. [20]	• DoS, Probe • R2L, U2R	• RT • NBTree	NSL-KDD	• Better detection accuracy.	• Overhead processing due to the excessive permutation operation causes a huge time and storage consumptions which is impractical for real time network traffic.
Ref. [22]	• DoS, Probe • R2L, U2R	• Ensemble (DT, SVM, logical regression, k-NN, adaboost, RF, deep neural network)	NSL-KDD	• Detection accuracy is improved.	• Deep neural network consumes a long time which is impractical for real network attacks and affects the response time of attack detection.
Ref. [28]	• DoS, Probe • R2L, U2R	• Ant Colony • Ensemble (DT)	KDD'99	• High model accuracy.	• Multiple of attack classes and large data classification increase the time complexity.
Ref. [29]	• DoS, Probe • R2L, U2R	• NB	NSL-KDD	• High detection accuracy. • Low false positive rate.	• Due to lowering number of features, the time complexity badly affects the real-time detection .

Table 1: Summary of relevant work in feature selection of IDS deployed for anomaly based detection.

approach, an ensemble classifier method is performed based on supporting vector machines, K-nearest neighbor, and multi-layer perception to create a voting algorithm for the feature dimensionality reduction. The execution of this model leads to an increase in the rates of accuracy.

An improved intrusion detection J48 algorithm was proposed in [31]. The authors integrated various approaches like the J48, naive Bayes, random tree, and NB-Tree. The model was tested over the NSL

KDD intrusion data set. The result depicted that, the improved J48 algorithm enhanced the ability of attack detections.

In [26], an ensemble core vector machine (CVM) approach was applied for detecting all four types of attacks. In the preprocessing phase, the chi-square test was performed on twenty-one of the input features. As a result, ten features were selected based on this test. Next, the CVM classifier is modelled for each type of attack. The execution of this methodology shows a high detection rate and decrease in false positive rate.

The authors in [27] aimed to present a model that is able to reduce the number of the applied features. This model was conducted using random forest through the information gain method. The significant features were tested over the NSL-KDD standard dataset. By comparison, the model results outperform other existing algorithms.

In [28], Mousavi proposed an intelligent IDS framework combining artificial intelligence, data-mining and machine learning techniques. He used MCC-based GFR as a feature removal method. Both of ACO and ensemble of decision trees methods are implemented as a classifier for feature selection. Finally, the authors presented 2 phases framework defined as WBNAD in [29]. In Phase I: GA is exploited for Feature selection as a wrapper approach. Then, Test instances are classified using Bayesian Network. Table 1 illustrates a summary of anomaly based detection research works reviewed in this section.

3 The proposed MFSR-IDS framework description

As shown in Figure 1, the conceptual structure of the proposed MFSR-IDS framework consists of three phases: data preprocessing, feature selection and classification. The data resampling, moving, balancing, categorizing, and reducing are executed within the preprocessing phase. Dataset features are selected using some rankers and its dimension is reduced in the feature selection phase. In the classification phase, the optimum features are selected by deleting the irrelevant or useless features using a set of wrappers. In addition, during this phase, the feature dimension is reduced once more. As previously stated, the feature dimensionality is reduced in three layers distributed across the MFSR-IDS framework.

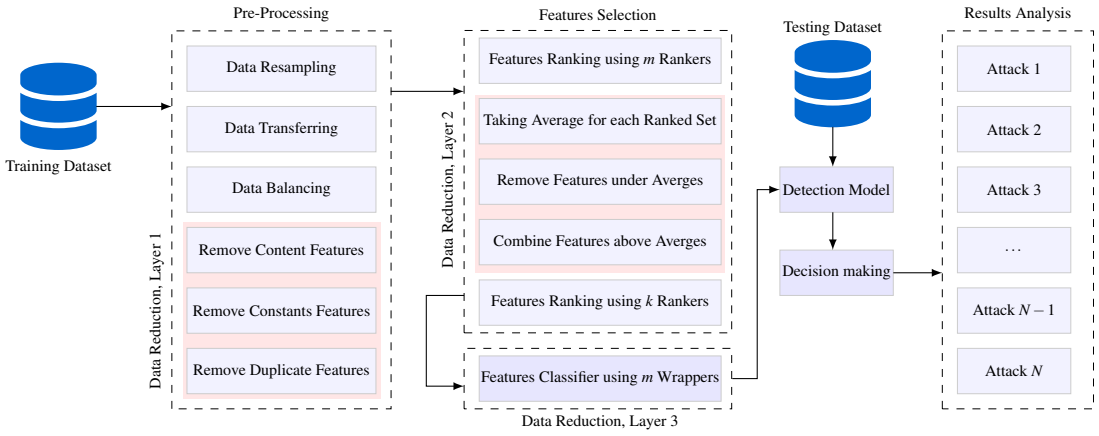

Figure 1: Conceptual structure of the MFSR-IDS framework.

3.1 Preprocessing Phase

Preprocessing phase is essential for data reduction because processing large amounts of network traffic with a huge number of features is very difficult. Furthermore, network traffic patterns come in a variety of

formats and dimensionalities. In order to obtain reliable performance, preprocessing is used to transform real-world datasets into understandable formats and behaviours.The preprocessing phase of the dataset fearures includes the following four processes:

1. **Features grouping:** Dataset's features are divided into four types: basic, content, host, and traffic.

2. **Features resampling**: Asymmetrical distributions is resampled to eliminate the effects of imbalanced normal and effected data.

3. **Features transferring:** The labelled features are converted into numeric values using the Hot Encoding algorithm for computing and testing purposes [32]. For example, the labelled protocol types TCP and UDP, are replaced with the numeric values 15 and 17.

4. **Feature cleaning:** The redundant and irrelevant features are eliminated for efficient classification tasks. Content, constant, quasi-constant and duplicate features are eliminated. Content features are not highly ranked when using multiple rankers and are rarely chosen in the final subset features. Constant and quasi-constant features provide no useful information for training. Finally, duplicate features add overhead and unnecessary delay to algorithm training.

5. **Feature ranking and reduction:** The main goal of this step is to extract the most influence κ features from n original features, where $n \gg \kappa$. This can be done through two sub-steps: 1) ranking the dataset features using m different rankers. 2) The combination of the selected features form the first sub-step is then ranked again using another p rankers, where each sub-step employs a unique set of rankers. Let $F = \{f_1, f_2, f_3, \ldots, f_n\}$ be the set of features of the dataset Ω with dimension n. Let $\Re = \{r_1, r_2, r_3, \ldots, r_m\}$ be set of rankers used for ranking the set F. Let $F'_l(r_i)$ be the ranked features of class $l \in L$ when applying the ranker r_i on the set F, where L denoting the set of classes (labels). For $i = 1,2,3,\ldots,m$ and $j = 1,2,3,\ldots,n$, the output ranked set $F'_l(r_i)$ is comprised of descending ordered pairs $\left(f'_j, d_{i,j}\right)$ based on the value of $d_{i,j}$. That is:

$$F'_l(r_i) = \left\{\left(f'_1, d_{i,1}\right), \left(f'_2, d_{i,2}\right), \ldots, \left(f'_n, d_{i,n}\right)\right\} \tag{1}$$

where $d_{i,j}$ denotes the numeric rank value of the feature f'_j within the ranked set $F'_l(r_i)$ and $d_{i,1} \geq d_{i,2} \geq \ldots \geq d_{i,n}$. For each F'_{r_i}, we calculate the average a_i of the rank values $d_{i,j}$ as follows:

$$a_i = \frac{\sum_{j=1}^{n} d_{i,j}}{n} \tag{2}$$

Based on the values of a_i, we remove all features from $F'_l(r_i)$ where $d_{i,j} \leq a_i$ to obtain the reduced feature set $F''_l(r_i)$. That is:

$$F''_l(r_i) = \left\{\left(f'_1, d_{i,1}\right), \left(f'_2, d_{i,2}\right), \ldots, \left(f'_w, d_{i,w}\right)\right\}, \qquad w \leq n \tag{3}$$

where $d_{i,w} \geq a_i$. Let's define the new dataset Π_l, which are made up of the concatenation of $F''_l(r_i)$, $i = 1,2,3,\ldots.m$, without duplication of features. That is:

$$\Pi_l = \left\{F''_l(r_1), F''_l(r_2), F''_l(r_3) \ldots, F''_l(r_m)\right\} \tag{4}$$

Let $\Re' = \left\{r'_1, r'_2, r'_3, \ldots, r'_p\right\}$ be a new set of rankers. The dataset Π_l is ranked using one ranker from $\Re'$ to obtain the new dataset Π'_l. The resulted dataset Π'_l is then employed as an input to the selection and classifier phase. Algorithm 1 illustrates the feature ranking and reduction process. .

Algorithm 1: Ranking and reduction process

Input: Dataset F, Ranker sets $\Re$, $\Re'$, m, n
Output: Reduced and ranked datasets Π'_l
for $i \in m$ **do**
  $F'_l(r_i) \leftarrow Ranking(F, r_i)$
  $sum \leftarrow 0$
  for $j \in n$ **do**
    $sum \leftarrow sum + r_i.d_{i,j}$
  end for
  $a_i \leftarrow sum/n$
end for
for $i \in m$ **do**
  for $j \in n$ **do**
    if $F''_l(r_i).d_{i,j} \geq a_i$ **then**
      Add $F''_l(r_i).f_j$ to Π_l
    end if
  end for
end for
Select ranker r'_i from $\Re'$, $i \in p$
$\Pi'_l \leftarrow Ranking(\Pi_l, r'_i)$

3.2 Feature selection

Feature selection (FS) is a vital and crucial phase. Using FS, we can improve our learning model in a number of ways. i) Avoid learning from noise and overfitting. ii) Improved accuracy. iii) Reducing training time. FS is process of reducing dataset dimension by selecting only relevant data from the original features and removing the non-essential data [33]. As a result, a robust model which satisfies the classification performance criteria is developed. The FS can be supervised or unsupervised. The methods that use the output label class for FS are referred to as supervised FS. Under supervised, there are four common FS approaches, namely filter-based, wrapper-based, embedded-based and hybrid-based algorithms [34, 35, 33].

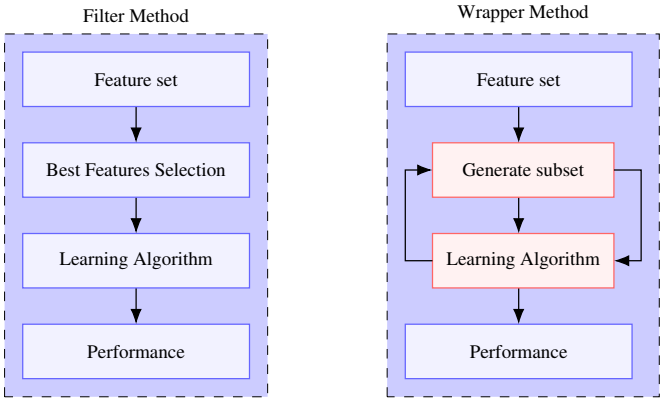

Figure 2: Features Selection Methods.

- The filter-based approach: In this method, features are eliminated based on their relationship to the output or how they correlate with the output. In other words, it focuses on statistical properties of the features (i.e., correlation) in order to eliminate features that aren't quite insightful [36]. As shown in Figure 2-a, the filter-based approach employs a ranking search method that is independent of any machine learning algorithms, as well as being computationally simple, fast, and scalable. Filters may be univariate, in which the important features are measured individually, or multivariate, in which the important features are calculated in a group.

- The wrapper-based approach: This approach measures the value of features set using the classification methods. As a result, the features are chosen by the classifier model itself. However, as shown in Figure 2-b, it uses a learning algorithm to evaluate which features are useful [37]. Despite the fact that this approach is more computationally complex, it provides better performance than the filter approach.

- The embedded-based approach performs feature selection within the training phase (i.e., during the classifier construction).

- Hybrid-based approach: This approach uses filter to select the most relevant features and then refining them using wrapper approach.

In this paper, we will use the embedded-based approach for FS, in which FS is implemented within the classifier phase, as shown in the next section.

3.3 Feature classification

Classification is a technique for categorizing instances in a dataset into a set of labels L using supervised machine learning techniques. In other words, classification is the process of deciding which label or class a new data belongs to. The dataset classification conducted in two subsequent phases namely, training phase and test data phase [38]. Typically, during the training phase, a classifier is generated and a target is learned. The classes are predicted during the testing phase [39]. There are a variety of different classification techniques for classifying intrusion detection datasets. The wrapper approach combines a feature search algorithm with a classifier to evaluate all possible feature combinations and choose the best feature subset with the lowest classification error [40]. Wrapper can be either forward selection, backward elimination or random scheme. In this paper, we consider the backward elimination technique, which starts with all of the features and eliminates the least significant feature at each iteration, improving the accuracy of the model. We keep doing this until there is no improvement when features are removed.

Let $C = \{c_1,\ c_2, \ldots, c_M\}$ be a set of M classifiers and $\Pi'_l = \{f'_1,\ f'_2,\ f'_3, \ldots, f'_q\}$ be the initial feature set with length $|\Pi'_l| = q$, $l \in L$. For each classifier $c_i \in C$ $(1 \leq i \leq M)$ the initial accuracy a_i^0 corresponding to c_i is calculated for the entire subset Π'_l. Exclude the feature f_j $(1 \leq j \leq q)$ from Π'_l and calculate the accuracy a_i^j using the performance measure $a_i^j = \Upsilon\left(c_i, \Pi'_c - \{f_j\}\right)$. If $a_i^j > a_i^0$ then $a_i^0 = a_i^j$. In such case the new subset Π'_l is given as

$$\Pi'_l = \Pi'_l - \left\{\ f_j \in \Pi'_l \mid \Upsilon\left(c_i, \Pi'_l - \{f_j\} > a_i^0\right)\right\}, \qquad 1 \leq j \leq q \tag{5}$$

This implies that there is gain in both classification accuracy and feature elimination. For each classifier $c_i \in C$, this process continues for the dataset Π'_l as illustrated in algorithm 2 .

Algorithm 2: Classification Phase

Input: Π'_l, $l \in L$, C
Output: Accuracy a_i^0 for $i = 1,2,3\ldots,\left|\Pi'_l\right|$.
for $l \in L$ **do**
  for $i \in |C|$ **do**
    $a_i^0 = \Upsilon\left(c_i, \Pi'_l\right)$
  end for
  for $i \in |C|$ **do**
    for $j \in \Pi'_l$ **do**
      $a_i^j = \Upsilon\left(c_i, \Pi'_l - \left\{f_j\right\}\right)$
      if $a_i^j > a_i^0$ **then**
        $\Pi'_l = \Pi'_l - \left\{f_j\right\}$
        $a_i^0 = a_i^j$
      end if
    end for
  end for
end for

4 Experimental work

Experiments in this paper are conducted based on the well-known KDDCUP'99 which commonly and widely used in the validation of most IDS model [41]. This dataset compromises of approximately 4GB of network traffic in tcpdump format. As shown in Table 2, the KDDCUP'99 dataset has one normal data class and four attack classes, namely Probe, DoS, R2L and U2R attacks. As illustrated in Table 3, the

Dataset	DoS	Probe	U2R	R2L	Normal	Total
10% KDD	391,458	4,107	52	1,126	97,277	494,020
Corrected KDD	229,853	4,166	70	16,347	60,593	311,029
Whole KDD	3,883,370	41,102	52	1,126	972,780	4,898,430

Table 2: Number of samples in KDDCUP'99 Dataset.

KDDCUP'99 dataset Ω contains 41 of numeric and symbolic features $F = \{f_1,\ f_2,\ f_3,\ldots,f_{41}\}$ which can be logically divided into four categories namely, basic, content, host, and traffic features.

4.1 Preprocessing phase

The distribution of the normal and attack classes in the training and testing dataset are shown in Table 4. This distribution values are highly different which tend to the class imbalance problem where there is skewness towards the majority class.

Table 4 illustrates imbalanced distribution of the training dataset where DoS and Probe attacks forms (resp.) 79.24% and 0.83 % of the total size of the KDDCUP'99 instances. To overcome this problem, we first resample the dataset Ω into four balanced data subsets $\Omega_1 = \{normal, DoS\}$, $\Omega_2 = \{normal, Probe\}$, $\Omega_3 = \{normal, R2L\}$ and $\Omega_4 = \{normal, U2R\}$, where each data subset consists of a normal class and one attack class. In this paper, we only focus on detecting DoS and Probe attacks (i.e. Ω_1 and Ω_2). In such case, $L = \{N = normal,\ D = DoS,\ P = Probe\}$. Hot encoding (either manually or randomly) is

f_i	Basic Features	f_i	Content Features	f_i	Traffic Features	f_i	Host Features
1	Duration	10	hot	23	Count	32	Dst_host_count
2	Protocol_type	11	Num_failed_logins	24	Srv_count	33	Dst_host_srv_count
3	Service	12	Logged_in	25	Serror_rate	34	Dst_host_same _srv_rate
4	Flag	13	Num_compromised	26	Srv_serror_rate	35	Dst_host_diff_srv_rate
5	src_bytes	14	Root_shell	27	Rerror_rate	36	Dst_host_same_src_port_rate
6	dst_bytes	15	Su_attempted	28	Srv_rerror_rate	37	Dst_host_srv_diff_host_rate
7	Land	16	Num_root	29	Same_srv_rate	38	Dst_host_serror _rate
8	Wrong_fragment	17	Num_file_creations	30	Diff_srv_rate	39	Dst_host_srv_serror_rate
9	Urgent	18	Num_shells	31	Srv_diff_host_rate	40	Dst_host_rerror_rate
		19	Num_access_files			41	Dst_host_srv_rerror_rate
		20	Num_outbound_cmds				
		21	Is_hot_login				
		22	Is_guest_login				

Table 3: Different categories of KDDCUP'99 Features

Dataset Label	DoS	PRobe	U2R	R2L	Total attacks	Total Normal
Training Data	79.24 %	0.83 %	0.01 %	0.23 %	80.31 %	19.69%
Testing Data	73.90 %	1.34 %	0.07 %	5.20 %	81.51 %	19.49 %

Table 4: Normal and attacks classles distribution within KDDCUP'99 dataset.

used to convert each feature within Ω_1 and Ω_2 from text or symbolic into numerical form. For instance, in the case of protocol type feature, 1 is assigned to TCP, 2 to ICMP ,and 3 to UDP and so on. For effective training and classification, any redundant details and meaningless features within Ω_1 and Ω_2 should be removed. This is the first layer of features dimensionality reduction which conducted using Python language.This is done using the following steps:

1. Removing duplicate features: Duplicate values are removed because it adds overhead and unnecessary delay to the training time. This is done using WEKA preprocessing filter.

2. Remove content features: After conducting many experiments and in the phase of feature selection, the content features from f_{10} to f_{22} in Table 3 are not highly ranked and most of the time they are below the rankers' mean value. They are rarely being selected in the final features' subset. As a result, the sizes of Ω_1 and Ω_2 are reduced to 28 features.

3. Removing constant and quasi- constant features: Both constant and quasi-constant features don't provide useful information for making predictions and they are not helping in the classification of the training dataset .Therefore, it must be removed from the dataset. Table 5 illustrates the constant and quasi-constant features for both DoS and Probe attacks. By the end of this step, the sizes of Ω_1 and Ω_2 are reduced to 24 and 20 respectively.

4. **Class Balancing:** Employing the class balancer to re-weight instances in Ω_1 and Ω_2 such that their total weight is equivalent as shown in Table 6. .

5. **Feature ranking and reduction:** This is the second layer for feature reduction. Now we have a training dataset Ω_1 for DoS attack with feature set $F_D = \{f_1,\ f_2,\ f_3, \ldots, f_{24}\}$ and a training dataset Ω_2 for Probe attack with feature set $F_P = \{f_1,\ f_2,\ f_3, \ldots, f_{20}\}$. The training dataset Ω_1 and Ω_2 are still considered too big to be used in model training. Therefore, using algorithm 1, ranking-based

Constant Feature		Quasi-Constant Feature	
DoS	**Probe**	**DoS**	**Probe**
Land	Land	Wrong_fragment	serror_rate
	Wrong_fragment	Dst_host_srv_diff_host_rate	srv_serror_rate
	Urgent	Urgent	dst_host_srv_diff_host_rate
			dst_host_serror_rate
			dst_host_srv_serror_rate

Table 5: Constant and Quasi-Constant features for DoS and Probe attacks

Ω_1			Ω_2		
Class	**Count**	**Weight**	**Class**	**Count**	**Weight**
DoS	572	6703.0	Probe	509	8460.5
Normal	13014	6703.0	Normal	16412	8460.5

Table 6: Classes balancer

feature selection technique is applied on F_D and F_P for more features reduction. To do this, we select the set of rankers as $\Re = \{r_1 = Chi\ Square,\ r_2 = OneR,\ r_3 = Correlation\}$. Chi-Square and Correlation are scoring (i.e. statistical test) methods, while OneR is a rule-based method. Each ranker r_i $(i = 1,\ 2,\ 3)$ ranks F_D and F_P according to its merit to generate $F'_D(r_i)$ and $F'_P(r_i)$ and then calculates the metric average a_i using (2).

All features of $F'_D(r_i)$ and $F'_P(r_i)$ whose ranking value is less than the average a_i are dropped to generate $F''_D(r_i)$ and $F''_P(r_i)$. For all $i = 1,\ 2,\ 3$, combine all the features of $F''_D(r_i)$ without duplicates (i.e. removing all the repeated one) to produce the updated training datasets Ω_1 and Ω_2 with size 14 features. Finally, gain ratio (GR) ranker is applied on the reduced training datasets Ω_1 and Ω_2,where their 14 features are ranked and sorted ascendingly according to its merit to produce the training datasets Ω'_1 and Ω'_2

4.2 Training phase

First, the machine learning algorithms C4.5, Naïve Bayes (NB), RepTree and Random Forest (RF) are used as wrappers for training datasets Ω'_1 and Ω'_2 with 10-fold cross validation to avoid the effect of data sampling when evaluating the IDS. In such case, $C = \{c_1 = C4.5,\ c_2 = NB,\ c_3 = RF,\ c_4 = RepTree\}$. Second, we used the homogenous ensemble method [42]. In this method, number of learners are constructed from training data using C4.5, NB, RepTree, RF and Random Tree (RT) algorithms. Each base learner has gained the ability to classify unseen arriving instances after the training step is done. Following that, each base learner delivers their own judgement for new input during the testing phase, and the final outcome is formed by integrating their outputs.[19].

For generating an ensemble model, there are three prominent ensemble-based algorithms. Namely, Bagging (Bootstrap aggregation), Boosting, and Stacking [22]. In this paper we will use Bagging algorithm.Bagging is a parallel ensemble meta technique that generates and integrates several models to form a predictive model [43]. Using the bootstrap sampling approach, sub samples are generated at random from the original dataset. The voting procedure is then used to all ensembles in order to choose the best subset that performed the best. This strategy is utilised to improve the accuracy of the classifier as well

Classifier	DoS Attack	Probe Attack
C4.5	7	6
NB	9	9
RepTree	3	3
RF	9	5
Bagging +C4.5	6	10
Bagging+NB	11	7
Bagging+RepTree	4	3
Bagging+RF	6	7
Bagging+RT	11	9

Table 7: The number of the best features generated by each classifier for DoS and Probe attacks.

as the prediction performance.

By starting the classification step using " classify" tab in weka, the model accuracy for each classifier is calculated and considered as an initial threshold for the removal process. Using algorithm 2, one feature at a time is removed from the training datasets Ω_1' and Ω_2', and then calculate the accuracy of a certain classifier. If the accuracy of that classifier is below the initial threshold then this feature is considered important and re-included in the dataset. Otherwise, the feature is dropped and the initial accuracy is updated and considered as the new threshold. The process is repeated until the final subset for each classifier is reached. Table 7 reveals the number of features generated by each classifier.

4.3 Testing and analysis

Each best selected number of features that are obtained in the training phase will be used to classify attacks via five classification algorithms and one homogenous ensemble method. The testing and analysis are carried out in the WEKA environment, which enables us to combine ranker and wrapper techniques. The intrusion detection is done over DoS and Probe attacks. The performance of the proposed framework is evaluated in terms of DR, FPR, FScore, ROC area, Acc and processing time.

Table 8 indicates that when using C4.5, RepTree, and ensemble with C4.5, RepTree, RF and RT algorithms, the proposed model outperforms the model introduced in [21] in terms of DR and FScore, ROC Area, and Accuracy. Table 9 illustrates that the proposed model outperform the model introduced in [21] in terms of the building time for C4.5, NB, RepTree, RF and RT algorithms. Even though ensemble technique has certain advantages in terms of detection effect, it takes a bit longer time in our comparative experiment. As a result, it will cause a considerable detection delay in real applications and will affect the attack detection response time.

Table 10 shows the Probe attack's experimental outcomes for several evaluation metrics. As per the results obtained, it was seen that both DR and FScore achieved satisfactory results in all classifications algorithms compared to [21] in the literature. In addition, our accuracy is higher in all classifications algorithms except ensemble with NB algorithm. The FPR of our proposed model doing well with all classifier except C4.5 and ensemble with NB algorithms. Table 11 illustrates the building time of our model compared to [21] in the literature. From result, we note that our model enhances the building in all classifiers. Table 12 illustrates the evaluation results of the proposed framework and some existing anomaly-based IDS approaches in terms of detection rate, false positive rate, FScore and accuracy. From the table, we note that, the proposed MFSR-IDS framework presents the best outcome of accuracy in the case of C4.5, RepTree, and ensemble classifier algorithms. In addition, the performance of MFSR-IDS framework in terms of false positive rate (FPR) outperforms most of the some existing approaches

Author	Algorithm	FPR	DR	FScore	ROC Area	Acc	Features #
Our Model	C4.5	0.001	0.992	0.995	0.998	99.86	7
	NB	0.015	0.908	0.915	0.988	97.67	9
	RepTree	0.001	1	0.999	0.999	99.89	3
	RF	0.003	0.98	0.988	1	99.67	8
	Bagging+C4.5	0.001	0.999	1	1	99.97	6
	Bagging+NB	0.008	0.992	0.989	0.987	98.89	11
	Bagging+RT	0.001	0.999	0.999	1	99.94	4
	Bagging+RF	0	1	0.999	1	99.91	6
	Bagging+RepTree	0	1	1	1	99.996	6
Taha [21]	C4.5	0.002	0.989	0.993	0.998	99.82	6
	NB	0.019	0.995	0.994	0.989	99.04	12
	RepTree	0.001	1	0.999	0.999	99.84	3
	RF	0.001	0.991	0.994	1	99.84	25

Table 8: The performance results of the proposed model for DoS attack detection compared to Ref. [21]

Author	Classifier				Ensemble (Bagging)				
	C4.5	NB	RepTree	RF	C4.5	NB	RepTree	RF	RT
Our Model	0.71	1.1	0.35	6.34	2.31	1.11	0.85	53.6	1.17
Taha [21]	1.75	3.83	0.56	15.75	-	-	-	-	-

Table 9: Building and testing time for DoS attack in seconds

Author	Algorithm	FPR	DR	FScore	ROC Area	Acc	Features #
Our Model	C4.5	0.006	0.893	0.885	0.936	98.8	6
	BN	0.022	0.704	0.794	0.979	97.7	9
	RepTree	0.004	0.921	0.887	0.954	98.8	3
	RF	0.004	0.925	0.906	0.986	99.00	5
	Bagging+C4.5	0	1	0.998	0.998	99.78	10
	Bagging+NB	0.193	0.821	0.852	0.936	84.64	7
	Bagging+RT	0.001	0.999	0.998	0.999	99.76	3
	Bagging+RF	0	1	0.998	1	99.78	7
	Bagging+RepTree	0	1	0.998	0.999	99.79	9
Taha [21]	C4.5	0.005	0.891	0.806	0.984	98.08	6
	BN	0.023	0.69	0.784	0.985	97.2	13
	RepTree	0.005	0.844	0.794	0.983	97.9	5
	RF	0.004	0.916	0.816	0.996	98.2	20

Table 10: The performance results of the proposed model for Probe attack

Author	Classifier				Ensemble (Bagging)				
	C4.5	BN	RepTree	RF	C4.5	NB	RepTree	RF	RT
Our Model	1.02	1.02	0.46	4.82	4.27	0. 87	0.78	46.2	1.11
Taha [21]	17.77	17.77	0.6	15.94	-	-	-	-	-

Table 11: Building and testing time for Probe attack in seconds

Algorithm	Author	DR (%)	FPR	FScore	ACC
C4.5	Proposed Model	99.2	0.001	0.995	99.86
	Taha et al, [21]	98.9	0.002	0.993	99.82
	Wang et al, [25]	99.87	0.14	0.997	x
	Sainis et al, [44]	x	x	x	99.94
NB	Proposed Model	90.8	0.015	0.915	97.67
	Taha et al, [21]	99.5	0.019	0.994	99.04
	Wang et al, [25]	99.88	0	0.999	x
	Radoglou et al, [39]	x	0.049	0.751	91.7
	Sainis et al, [44]	x	x	x	96.16%
REPTree	Proposed Model	100	0.001	0.999	99.89
	Taha et al, [21]	100	0.001	0.999	99.84
	Pham et al, [45]	x	x	x	83.22
RF	Proposed Model	98	0.003	0.988	99.67
	Taha et al, [21]	99.1	0.001	0.994	99.84
	Radoglou et al, [39]	x	0.005	0.97	99
	Anusha et al, [46]	74	0.544	x	x
	Pham et al, [45]	x	x	x	80.58
	Sainis et al, [44]	x	x	x	99.94%
Ensemble	Proposed Model	100	0	1	99.996
	Zhou et al [34]	95.3	0.016	95.2	95.3
	Divyasree et al, [26]	99.12	0.4714	x	99.05

Table 12: Performance comparisons of the proposed algorithm and IDS techniques in terms of detection accuracy, false positive rate, Fscore and accuracy.

Bagging+RF and Bagging+RT.

5 Conclusion and Future Work

In this paper, we proposed a multi-layers feature selection and reduction IDS framework for DoS and Probe attacks. In this framework, the input features are reduced in three layers. First, the irrelevant and redundant features are removed during cleaning dataset. Second, we developed an algorithm that ranks features using Chi-Square, OneR, and correlation rankers. The ranked features are reduced using the average metric of each ranker. Finally, the features are ranked using Gain Ratio and reduced again using different classifier algorithms. To improve the detection accuracy, the advantages of different classifier algorithms are integrated by deploying the methodology of ensemble learning to build the predictive model. The experimental results show that: 1) The efficiency of our proposed framework based on C4.5, RepTree and RF is enhanced compared to the previous works in literature. 2) Ensemble learning based

on the C4.5, RepTree and RF algorithms improves the IDS efficiency in terms of DR, FPR, FScore, and Accuracy. 3) The proposed framework is superior to the performance of all approaches mentioned in section 4 in the case of using Bagging+RT. 4) The building time is improved when the classifiers C4.5, NB, RT, RF and Random Tree are deployed to build the predictive model.

In our future work, we plan to enhance the performance of our MFSR-IDS approach via employing optimization approaches, detecting significant threats and testing using real data set.

References

[1] Puneet Kumar, Sahil Garg, Amritpal Singh, Shalini Batra, Neeraj Kumar, and Ilsun You. Mvo-based 2-d path planning scheme for providing quality of service in uav environment. *IEEE Internet of Things Journal*, 5(3):1698–1707, June 2018.

[2] Vishal Sharma, Ilsun You, Kangbin Yim, Ing-Ray Chen, and Jin-Hee Cho. Briot: Behavior rule specification-based misbehavior detection for iot-embedded cyber-physical systems. *IEEE Access*, 7:118556–118580, May 2019.

[3] Gunupudi Rajesh Kumar, Nimmala Mangathayaru, and Gugulothu Narsimha. An approach for intrusion detection using novel gaussian based kernel function. *J. Univers. Comput. Sci.*, 22(4):589–604, 2016.

[4] Manu Bijone. A survey on secure network: intrusion detection & prevention approaches. *American Journal of Information Systems*, 4(3):69–88, 2016.

[5] Sungkwan Kim, Junyoung Park, Kyungroul Lee, Ilsun You, and Kangbin Yim. A brief survey on rootkit techniques in malicious codes. *Journal of Internet Services and Information Security.*, 2(3/4):134–147, December 2012.

[6] Shalki Sharma, Anshul Gupta, and Sanjay Agrawal. A survey of intrusion detection system for denial of service attack in cloud. *International Journal of Computer Applications*, 120(19), June 2015.

[7] Mohammad M Shurman, Rami M Khrais, and Abdulrahman A Yateem. Iot denial-of-service attack detection and prevention using hybrid ids. In *Proc. of the 2019 International Arab Conference on Information Technology (ACIT'19), Al Ain, UAE*, pages 252–254. IEEE, December 2019.

[8] Alaa F Sheta and Amneh Alamleh. A professional comparison of c4. 5, mlp, svm for network intrusion detection based feature analysis. *Computer Networks and Internet Research CNIR*, 47:15–30, December 2015.

[9] Aparna U.R. and Shaiju Paul. Feature selection and extraction in data mining. In *Proc. of the 2016 Online International Conference on Green Engineering and Technologies (IC-GET'16), Coimbatore, India*, pages 1–3. IEEE, November 2016.

[10] Samina Khalid, Tehmina Khalil, and Shamila Nasreen. A survey of feature selection and feature extraction techniques in machine learning. In *Proc. of the 2014 Science and Information Conference (SAI'14), London, UK*, pages 372–378. IEEE, August 2014.

[11] Kaiser Nahiyan, Samilat Kaiser, Ken Ferens, and Robert McLeod. A multi-agent based cognitive approach to unsupervised feature extraction and classification for network intrusion detection. In *Proc. of the 1st International Conference on Applied Cognitive Computing (ACC'17). Las Vegas, Nevada, USA*, pages 25–30. CSREA, July 2017.

[12] Abdur Rahman Onik, Nutan Farah Haq, Lamia Alam, and Tauseef Ibne Mamun. An analytical comparison on filter feature extraction method in data mining using j48 classifier. *International Journal of Computer Applications*, 124(13), August 2015.

[13] Rebecca Gurley Bace and Peter Mell. *Intrusion detection systems*. U.S. Department of Commerce, Technology Administration, National Institute of Standards and Technology, 2001.

[14] Solane Duque and Mohd Nizam bin Omar. Using data mining algorithms for developing a model for intrusion detection system (ids). *Procedia Computer Science*, 61:46–51, October 2015.

[15] Jaina Patel and Krunal Panchal. Effective intrusion detection system using data mining technique. *Journal of Emerging Technologies and Innovative Research*, 2(6):1869–1878, June 2015.

[16] Shadi Aljawarneh, Monther Aldwairi, and Muneer Bani Yassein. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *Journal of Computational Science*, 25:152–160, March 2018.

[17] Noureldien A Noureldien and Izzedin M Yousif. Accuracy of machine learning algorithms in detecting dos attacks types. *Science and Technology*, 6(4):89–92, 2016.

[18] Rashmi Ravindra Chaudhari and Sonal Pramod Patil. Intrusion detection system: classification, techniques and datasets to implement. *International Research Journal of Engineering and Technology (IRJET)*, 4(2):1860–1866, February 2017.

[19] Nabeel H. Al-A'araji, Safaa O. Al-Mamory, and Ali H. Al-Shakarchi. Classification and clustering based ensemble techniques for intrusion detection systems: A survey. *Journal of Physics: Conference Series*, 1818(1):012106, March 2021.

[20] Jasmin Kevric, Samed Jukic, and Abdulhamit Subasi. An effective combining classifier approach using tree algorithms for network intrusion detection. *Neural Computing and Applications*, 28(1):1051–1058, June 2017.

[21] Taha AIT Tchakoucht and Mostafa Ezziyyani. Building a fast intrusion detection system for high-speed-networks: probe and dos attacks detection. *Procedia Computer Science*, 127:521–530, March 2018.

[22] Xianwei Gao, Chun Shan, Changzhen Hu, Zequn Niu, and Zhen Liu. An adaptive ensemble machine learning model for intrusion detection. *IEEE Access*, 7:82512–82521, June 2019.

[23] Deris Stiawan, Ahmad Heryanto, Ali Bardadi, Dian Palupi Rini, Imam Much Ibnu Subroto, Mohd Yazid Bin Idris, Abdul Hanan Abdullah, Bedine Kerim, Rahmat Budiarto, et al. An approach for optimizing ensemble intrusion detection systems. *IEEE Access*, 9:6930–6947, December 2020.

[24] Mohit Dua et al. Attribute selection and ensemble classifier based novel approach to intrusion detection system. *Procedia Computer Science*, 167:2191–2199, April 2020.

[25] Wei Wang, Yongzhong He, Jiqiang Liu, and Sylvain Gombault. Constructing important features from massive network traffic for lightweight intrusion detection. *IET Information Security*, 9(6):374–379, 2015.

[26] TH Divyasree and KK Sherly. A network intrusion detection system based on ensemble cvm using efficient feature selection approach. *Procedia computer science*, 143:442–449, November 2018.

[27] Alaa Abd Ali Hadi and Al-Awsat Al-Furat. Performance analysis of big data intrusion detection system over random forest algorithm. *International Journal of Applied Engineering Research*, 13(2):1520–1527, 2018.

[28] Seyed Morteza Mousavi, Vahid Majidnezhad, and Avaz Naghipour. A new intelligent intrusion detector based on ensemble of decision trees. *Journal of Ambient Intelligence and Humanized Computing*, (29):1–13, November 2019.

[29] Md Reazul Kabir, Abdur Rahman Onik, and Tanvir Samad. A network intrusion detection framework based on bayesian network using wrapper approach. *International Journal of Computer Applications*, 166(4):13–17, May 2017.

[30] Fadi Salo, Ali Bou Nassif, and Aleksander Essex. Dimensionality reduction with ig-pca and ensemble classifier for network intrusion detection. *Computer Networks*, 148:164–175, January 2019.

[31] Shadi Aljawarneh, Muneer Bani Yassein, and Mohammed Aljundi. An enhanced j48 classification algorithm for the anomaly intrusion detection systems. *Cluster Computing*, 22(5):10549–10565, 2019.

[32] Yazan Otoum and Amiya Nayak. As-ids: Anomaly and signature based ids for the internet of things. *Journal of Network and Systems Management*, 29(3):1–26, March 2021.

[33] Nazrul Hoque, Dhruba K Bhattacharyya, and Jugal K Kalita. Mifs-nd: A mutual information-based feature selection method. *Expert Systems with Applications*, 41(14):6371–6385, April 2014.

[34] Yuyang Zhou, Guang Cheng, Shanqing Jiang, and Mian Dai. Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Computer networks*, 174:107247, April 2020.

[35] A. Jović, K. Brkić, and N. Bogunović. A review of feature selection methods with applications. In *Proc. of the 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO'15), Opatija, Croatia*, pages 1200–1205. IEEE, May 2015.

[36] Saurabh Mukherjee and Neelam Sharma. Intrusion detection using naive bayes classifier with feature reduction. *Procedia Technology*, 4:119–128, June 2012.

[37] Sebastián Maldonado, Richard Weber, and Fazel Famili. Feature selection for high-dimensional class-imbalanced data sets using support vector machines. *Information Sciences*, 286:228–246, July 2014.

[38] Azar Abid Salih and Maiwan Bahjat Abdulrazaq. Combining best features selection using three classifiers in intrusion detection system. In *Proc. of the 2019 International Conference on Advanced Science and Engineering (ICOASE'19), Zakho Duhok, Iraq*, pages 94–99. IEEE, June 2019.

[39] Panagiotis Radoglou Grammatikis, Panagiotis Sarigiannidis, Georgios Efstathopoulos, and Emmanouil Panaousis. Aries: a novel multivariate intrusion detection system for smart grid. *Sensors*, 20(18):5305, September 2020.

[40] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, August 2005.

[41] Saharon Rosset and Aron Inger. Kdd-cup 99: knowledge discovery in a charitable organization's donor database. *ACM SIGKDD Explorations Newsletter*, 1(2):85–90, January 2000.

[42] Quang-Vinh Dang. Active learning for intrusion detection systems. In *Proc. of the 2020 RIVF International Conference on Computing and Communication Technologies (RIVF'20), Ho Chi Minh City, Vietnam*, pages 1–3. IEEE, October 2020.

[43] Lior Rokach. Ensemble-based classifiers. *Artificial intelligence review*, 33(1):1–39, November 2009.

[44] Ngoc Tu Pham, Ernest Foo, Suriadi Suriadi, Helen Jeffrey, and Hassan Fareed M Lahza. Improving performance of intrusion detection system using ensemble methods and feature selection. In *Proc. of the 2018 Australasian Computer Science Week Multiconference (ACSW'18), Brisband, Queensland, Australia*, pages 1–6. ACM, January 2018.

[45] Nachiket Sainis, Durgesh Srivastava, and Rajeshwar Singh. Feature classification and outlier detection to increased accuracy in intrusion detection system. *International Journal of Applied Engineering Research*, 13(10):7249–7255, September 2018.

[46] K Anusha and E Sathiyamoorthy. Comparative study for feature selection algorithms in intrusion detection system. *Automatic Control and Computer Sciences*, 50(1):1–9, April 2016.

Author Biography

Mohammed El-Shrkawey received his B.Sc. in Electrical engineering from the Military Technical College, Cairo in 1987, Master in Computer Engineering from the Faculty of Engineering, Al Azhar University, Cairo in 2002 and PHD in Network Security from the Faculty of Computers and Informatics, Cairo University in June 2007. He is currently an Associate professor in the Faculty of Computers and Informatics, Suez Canal University.

Marwa Alalfi received her B.Sc. in Computer Science from the Faculty of Computer and Informatics, Suez Canal University in 2008. Currently, she is pursuing her Master's degree in the Department of the Information System. Her research interests include Computer Security, Data Mining and Machine Learning.

Hassan Al-Mahdi received the BSc in Computing Science, MSc in Computer Science and PHD in Wireless Networks from the Faculty of Science, Suez Canal University, Egypt in 1994, 2001 and 2005, respectively. Currently, he is a Full Professor of Computer Networks at the Faculty of Computer and Informatics, Suez Canal University, Egypt. His researches are in fields of ad hoc networks, mobile cellular communications, cognitive radio networks, IoT, Cloud Computing, WSN, and the performance evaluation of computer networks. He has many international papers mostly in the area of performance evaluation of computer networks, IoT, WSN, Cloud Computing, Queuing System and Cryptography.