Building an Efficient Intrusion Detection System Based on Feature Selection and Ensemble Classifier

Yuyang Zhou[a,b,c], Guang Cheng[a,b,c,*], Shanqing Jiang[a,d] and Mian Dai[a,b,c]

[a]School of Cyber Science and Engineering, Southeast University, Nanjing, China
[b]Key Laboratory of Computer Network and Information Integration, Ministry of Education, Nanjing, China
[c]Jiangsu Provincial Key Laboratory of Computer Network Technology, Southeast University, Nanjing, China
[d]National Key Laboratory of Science and Technology on Information System Security, Beijing, China

ARTICLE INFO

Keywords:
Cyber Security
Intrusion Detection System
Data Mining
Feature Selection
Ensemble Classifier

ABSTRACT

Intrusion detection system (IDS) is one of extensively used techniques in a network topology to safeguard the integrity and availability of sensitive assets in the protected systems. Although many supervised and unsupervised learning approaches from the field of machine learning have been used to increase the efficacy of IDSs, it is still a problem for existing intrusion detection algorithms to achieve good performance. First, lots of redundant and irrelevant data in high-dimensional datasets interfere with the classification process of an IDS. Second, an individual classifier may not perform well in the detection of each type of attacks. Third, many models are built for stale datasets, making them less adaptable for novel attacks. Thus, we propose a new intrusion detection framework in this paper, and this framework is based on the feature selection and ensemble learning techniques. In the first step, a heuristic algorithm called CFS-BA is proposed for dimensionality reduction, which selects the optimal subset based on the correlation between features. Then, we introduce an ensemble approach that combines C4.5, Random Forest (RF), and Forest by Penalizing Attributes (Forest PA) algorithms. Finally, voting technique is used to combine the probability distributions of the base learners for attack recognition. The experimental results, using NSL-KDD, AWID, and CIC-IDS2017 datasets, reveal that the proposed CFS-BA-Ensemble method is able to exhibit better performance than other related and state of the art approaches under several metrics.

1. Introduction

Nowadays, the applications of the Internet help society in many areas such as electronic communication, teaching, commerce, and entertainment, it has become a part of daily life of the people. However, cyber security has become vulnerable due to the massive expansion of the computer networks and rapid emergence of the intrusion incidents. The necessity of developing cyber security has attracted considerable attention from industry and academia around the world. Despite the use of different security applications, such as firewalls, malware prevention, data encryption, and user authentication, many organizations and enterprises fall victims to contemporary cyber-attacks [4]. In order to sneak into the system, attackers might deliberately exploit the vulnerabilities of the target system and launch different types of attacks, which may lead to the leakage of private information.

As technology is rolling out, these attacks threaten the confidentiality, integrity, and availability of cyber systems all the time. Therefore, it is necessary to introduce intrusion detection systems (IDSs) [26, 89, 90, 91] to protect systems from a variety of attacks. To be more specific, IDSs are widely deployed in various distributed systems, perceiving the malicious intrusions and then taking rapid countermeasures to prevent further infections and spread. In general, IDSs can be classified into two major categories based on detection mechanisms: anomaly and misuse detection [42]. In detail, anomaly detection is designed to detect malicious actions through identifying deviations from a normal profile behavior. Such IDSs perform better at detecting novel types of attacks, however, they could not avoid a high false positive (FP) rate [68]. On the other hand, based on known patterns, misuse detection can effectively distinguish legitimate instances from the malicious ones [46]. Although this kind of IDSs is reliable for detecting known attacks, it cannot identify unknown attacks or variations of known ones.

Unfortunately, as the attackers become more sophisticated, new threats and vulnerabilities emerge rapidly. On the one hand, the risk for critical infrastructures to be compromised significantly increases in short order. On the other hand, in order to detect and deal with novel attacks, a higher requirement for IDS has also been brought forward. Hence, many approaches have been researched and developed to improve the detection rate and performance of IDSs. One of them is Machine learning (ML) [23, 24, 61], which can be applied for both anomaly and misuse detection models. By analyzing network traffic passing through central network nodes, an IDS not only needs to distinguish between benign and malicious traffic, but also infers the specific class of an attack occurring in the protected system.

However, in most instances, only a fraction of the traffic may indicate malicious behaviors while a network is flooded with normal traffic flows, which leads to the difficulty of identifying attacks with high Attack Detection Rate (ADR) while keeping the False Alarm Rate (FAR) low. There was one problem with the initial idea of applying ML in IDS, that is, a single classifier may not be strong enough to build

*Corresponding author
yyzhou@njnet.edu.cn (Y. Zhou); gcheng@njnet.edu.cn (G. Cheng); sqjiang@njnet.edu.cn (S. Jiang); mdai@njnet.edu.cn (M. Dai)
ORCID(s): 0000-0001-8626-0468 (Y. Zhou)

a good IDS. Thus, researchers have come up with the idea of constructing ensemble classifiers for IDSs [28, 77]. In general, the main goal of ensemble learning is to combine a set of individual classifiers and then make a better classification decision about the object submitted at the input [72]. For instance, training a single classifier on different subsets of an IDS dataset could produce different classification performances, however, an ensemble would average the output of multiple classifiers and therefore become a better option.

Moreover, the numerous attack types and network traffic attributes pose another challenge for ML as they expand the search space of the problem and lead to high computational and time complexity [6]. Notably, feature selection has been proven to be a good solution for an IDS, which detects highly relevant features and eliminates useless ones with a minimum degradation of performance [36, 44]. There are three main models that deal with feature selection: wrapper, filter, and embedded approaches. Information gain ratio based feature selection is one of classical filter algorithms, where information gain ratio represents a ratio of information gain to the intrinsic information. Although it solves the drawback of information gain and reduces a bias towards multi-valued attributes, however, it may be biased towards features with fewer values in some instances. Different from information gain ratio, correlation-based feature selection maximizes the relevance between the input features and the output and minimizes the redundancy of the selected features. This algorithm selects one feature at a time according to its strong correlation with outputs, which can be used to perform both attribute selection and tuple reduction flexibly.

In this paper, we propose a novel intrusion detection system to detect various types of attacks with high accuracy and efficiency. First, as a regular means of dimensionality reduction and redundancy elimination, a nature-inspired feature selection algorithm is proposed to retrieve a subset of the original features. Second, the imbalance between normal and malicious traffic has a negative effect on the accuracy of attack detection. To overcome this problem, our solution then utilizes ensemble classifier to reduce the bias among different training datasets. In this way, feature selection and ensemble classifier are combined to improve the stability and accuracy of the IDS with low computational and time complexity. Finally, an unbiased model can be generated to detect both popular and rare intrusive events. The major contributions of our work are summarized as follows:

- We propose a novel methodology that combines the benefits of feature selection and ensemble classifier with the aim of providing efficient and accurate intrusion detection.
- In the context of feature selection, we provide CFS-BA based approach, which is used to assess the correlation of the selected features and beneficial for optimizing the efficiency of the training and testing phase.
- To increase the multi-class classification performance on unbalanced datasets, we introduce an ensemble approach by combining decisions from multiple classifiers (C4.5, RF, and Forest PA) into one by utilizing a vote classifier based on the average of probabilities (AOP) combination rule.
- The proposal is compared with exsiting methods on an extensive testbed comprising of three datasets, namely: NSL-KDD, AWID, and CIC-IDS2017. Experimental results show that the proposed solution surpasses equivalent methods in terms of Accuracy (Acc), F-Measure, and ADR classification metrics, while keeping FAR at acceptable levels.

The rest of the paper is organized as follows. In Section 2, we review the background information concerning IDSs. Then, the proposed methodology is given in Section 3, while in Section 4 we provide the evaluation results through experiments and comparative analysis. Finally, the conclusion is presented in Section 5.

2. Related work

As a significant tool in computer based systems for ensuring cyber security, IDS constantly attracts the research community's attention. Although plenty of solutions have been proposed to improve the performance of IDS, in the context of this section, we only consider related work that falls under the ML based IDS, utilizes feature selection or ensemble classifier, and especially focuses on hybrid approaches.

2.1. On feature selection techniques

For purpose of reducing computational complexity, the technique of feature selection [59, 60], that can be used as a pre-processing step in ML algorithms, aims to eliminate irrelevant features while preserving or even enhancing the performance of the IDS. In order to obtain more robust and effective classifier, Hota and Shrivas [36] proposed a model that used different feature selection techniques to remove irrelevant features. The results indicate that C4.5 with information gain can achieve the highest accuracy with only 17 features for the NSL-KDD dataset. In addition, Khammassi and Krichen [44] have applied as a search strategy and logistic regression as a learning algorithm for network IDSs to choose the best subset. The results demonstrate that their method provides high detection rate with only 18 features for the KDDCup'99 and 20 features for the UNSW-NB15 dataset. Abdullah et al. [1] also proposed a framework of IDS with selection of features within the NSL-KDD dataset that are based on dividing the input dataset into different subsets, and combining them using Information Gain (IG) filter.

2.2. On ensemble classifiers

Moreover, ensemble methods are machine learning techniques that combine several base models in order to reduce false positive rates and produce more accurate solutions than a single model would. Gaikwad and Thool [33] proposed a bagging ensemble method using REPTree as its base classifier, which takes less time to build the model and provides

highest classification accuracy with lowest false positives on the NSL-KDD dataset. Jabbar et al. [41] proposed a cluster-based ensemble classifier for IDS, which is built with Alternating Decision Tree (ADTree) and k-Nearest Neighbor algorithm (kNN). The experimental results show that the proposed ensemble classifier outperforms other existing techniques in terms of accuracy and detection rate. In order to create a stronger learner, Paulauskas and Auskalnis [70] proposed an ensemble model of four different base classifiers: J48, C5.0, Naive Bayes, and Partial Decision List (PART), which depends on the idea of combining multiple weaker learners. Results prove that their ensemble model produces more accurate results for an IDS. In order to mitigate malicious events, in particular botnet attacks in Internet of Things (IoT) networks, Moustafa et al. [62] proposed new statistical flow features and developed an AdaBoost ensemble learning method to detect attacks effectively.

2.3. On hybrid approaches

Recently, many hybrid approaches using both feature selection and ensemble method have been produced to improve the performance of IDSs. Malik et al. [57] proposed a combination approach of Particle Swarm Optimization (PSO) and Random Forest (RF). More appropriate features for each class help the proposed model produce a higher accuracy along with low false positive rate compared with other algorithms. Pham et al. [72] built a hybrid model, which utilizes gain ratio technique as feature selection and bagging to combine tree-based base classifiers. Experimental results show that the best performance was produced by the bagging model that used J48 as the base classifier and worked on 35-feature subset of the NSL-KDD dataset. Abdullah et al. [1] also built an IDS using IG based feature selection and ensemble learning algorithms. The experiment on NSL-KDD dataset indicates that the highest accuracy obtained when using RF and PART as base classifiers under the product probability rule. In addition, Salo et al. [77] proposed a hybrid IDS which combines the feature selection approaches of IG and Principal Component Analysis (PCA) with an ensemble classifier based on Support Vector Machine (SVM), Instance-Based learning algorithms (IBK), and Multi-Layer Perceptron (MLP). A comparative analysis performed on several IDS datasets has proven that IG-PCA-Ensemble method exhibits better performance than the majority of existing approaches. Due to large-scale data produced from a massive network infrastructure, Khan et al. [45] proposed a scalable and hybrid IDS, which is based on Spark ML and Convolutional-LSTM (Conv-LSTM) network to employ the anomaly and misuse detection separately. Zhong et al. [99] also proposed a new anomaly detection model called HELAD, which is based on the Damped Incremental Statistics algorithm for feature selection and organic integration of multiple deep learning techniques for classification. In [82], a novel IDS based on hybrid feature selection and two-level classifier ensembles has been proposed, and experimental results show that it produces a significant improvement of the detection rate on the NSL-KDD and UNSW-NB15 datasets.

3. Proposed methodology

In order to increase the detection ability of IDS and prevent the service providers from attack, we propose an efficient ML-based IDS using a metaheuristic optimization algorithm based feature selection approach, and a vote classifier which is an ensemble of classifiers method. The AOP combination rule is integrated into the model for the decision step. During the experiments, 10-fold cross-validation (CV) approach is used to validate the performance of the model and classify benign traffic and various types of attacks.

Fig. 1 demonstrates the detection framework of the proposed ML-based IDS, which consists of the following four main phases:

- Datasets preprocessing: The first phase is to transform raw data into a format suitable for analysis by applying preprocessing to the original datasets.
- Dimensionality reduction: In order to overcome the problem of high-dimensional datasets, the feature selection approach based on CFS-BA is used to reduce the dimensionality of the datesets and select the most relevant features for each type of attacks.
- Classifiers training: For purpose of improving the accuracy of the IDS, we train three individual classifiers as base learners using C4.5, RF, and Forest PA, and build an ensemble classifier based on them.
- Attack recognition: The detection model is tested using a 10-fold cross-validation approach, and voting technique is used to combine the probability distributions of the base learners with the AOP combination rule to make classification decisions.

Finally, according to the results of the ensemble classifier, benign traffic and various intrusive events can be detected and classified with high classification accuracy. Detailed information about the framework is provided in Sections 3.1–3.2.

3.1. Feature selection

The aim of feature selection is to find a subset of the attributes from the original set which are representative enough for the data, and the attributions in the subset are highly relevant to the prediction. Feature selection approaches can be mainly categorized into wrapper, filter, and embedded approaches [35]. While filter approaches assess the relevance of the features from the dataset and the selection of the features is based on the statistics, the classification performance is used in wrapper approaches as a part of the feature subsets evaluation and selection processes. In contrast to wrapper approaches, embedded approaches are computationally less intensive because they incorporate an interaction between feature selection and learning process. Although embedded approaches integrate a regularised risk function to optimize the features designating parameters and the predictor parameters [14], it is not easy to make a modification in the classification model to get higher performance [56].

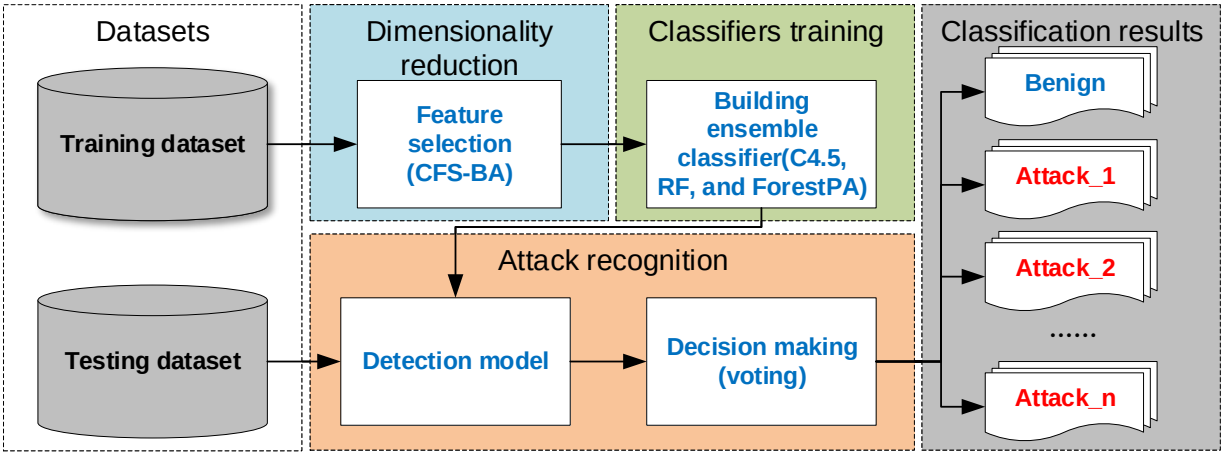


Figure 1: The framework of the proposed Feature selection-Ensemble model.

Modern intrusion detection datasets inevitably contain plenty of redundant and irrelevant attributes [2], which lower the efficacy of data mining algorithms and cause uninterpretable results [21]. Therefore, the first step in this study is to reduce the dimensionality and select the feature subset of the utilized dataset [77]. In this paper, a hybrid approach by combining CFS with BA is proposed to optimize the efficiency of the feature selection process and enhance the accuracy of the classification. The main concept of this approach is to evaluate the relevance and the redundancy of the selected feature subset which is searched in the given search space for the optimal solution.

3.1.1. Correlation-based feature selection (CFS)

CFS [80] is one of classical filter algorithms that choose features according to the result of the heuristic (correlation-based) assessment function. The preference of this function is to select subsets whose features are extraordinarily related with the class but uncorrelated with each other. While insignificant features that show low association with the class ought to be ignored on the grounds, repetitive features are chosen due to high relation with at least one of the rest of features. The acknowledgment of a feature will rely upon the degree to which it predicts classes in territories of the instance space not as of now anticipated by different features. The feature subset assessment function [81] in CFS is as:

$$M_s = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1) + \overline{r_{ff}}}} \tag{1}$$

In Eq. 1, M_s is the heuristic evaluation for a feature subset s including k features, $\overline{r_{cf}}$ is the mean correlation degree between features and the category label, and $\overline{r_{ff}}$ is the average inter-correlation degree among features. The evaluation of CFS is a method of correlation based on feature subsets. A bigger $\overline{r_{cf}}$ or smaller $\overline{r_{ff}}$ in acquired subsets by the method produce a higher evaluation value, and the set of features with the highest value found during the process is utilised to reduce the size of both the training and testing set.

3.1.2. Bat algorithm (BA)

The original bat algorithm was developed by Xin-She Yang in 2010 [94, 95]. The main inspirations for these works were the echolocation behavior of microbats. As BA uses frequency tuning, it is, in fact, the first algorithm of its kind in the context of optimization and computational intelligence. Each bat flies randomly with a velocity v_i^t, a location x_i^t, and a frequency f_i at iteration t, in a $d-$dimensional search or solution space. The location can be considered as a solution vector to a problem of interest. Among the n bats in the population, the current best solution x_* found so far can be archived during the iterative search process.

Defined by Yang [96], the updating rules for location x_i^t and velocity v_i^t at time step t are given by

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{2}$$
$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_*)f_i \tag{3}$$
$$x_i^t = x_i^{t-1} + v_i^t \tag{4}$$

where $\beta \in [0,1]$ is a random vector drawn from a uniform distribution.

For the local search part, once a solution is selected among the current best solutions, a new solution for each bat is generated locally using random walk

$$x_{new} = x_{old} + \varepsilon A^t \tag{5}$$

where ε is a random vector drawn from a uniform distribution in [-1,1] or a Gaussian distribution, while A^t is the average loudness of all the bats at this time step.

In addition, the loudness A_i^t and the rate r_i^t of pulse emission have to be updated accordingly as the iterations proceed. The updating rules for them can be written as

$$A_i^{t+1} = \alpha A_i^t \tag{6}$$
$$r_i^{t+1} = r_i^0(1 - e^{-\gamma t}) \tag{7}$$

where $0 < \alpha < 1$ and $\gamma > 0$ are constants.

3.1.3. CFS-BA approach for feature selection

In this section, we proposed CFS-BA based feature selection approach, which is used to assess the importance and the correlation of the selected feature subset. CFS-BA approach utilises correlation based feature technique to form the fitness functions and evaluation of integrity of the reduced feature subset.

For a feature subset S with k features, $S = (s_1, s_2, ...s_k)$, CFS assesses the mean feature-class correlation and average inter-correlation among features by using Eq. 1. As one of classical filter algorithms, CFS can easily select the subset of independently good features according to the result of correlation-based evaluation function. However, this feature subset may not be the best combination because of redundancy between features.

In order to remove the redundant features and reduce the dimensionality, BA, which inspired by the echolocation behavior of microbats, is introduced. In BA, every solution of the problem is denoted by the location of a bat, which can be represented by a vector. Bats fly in the search space to search for the best solutions and during this movement, the current best solution found so far can be archived. The population scans for the ideal arrangement by refreshing and updating the position of every bat based on Eq. 2−Eq. 4 during the iterative search process.

The feature selection process of the CFS-BA approach is presented in Algorithm 1. The main parts of the CFS-BA algorithm can be summarized as follows:

• Initialization (lines 1-4). The parameters of algorithm, generation and evaluation of the initial population are initialized here.

• New solution generation (lines 7-8). Here, bats in the population are moved in the search space according to updating rules of Eq. 2−Eq. 4.

• Local search process (lines 9-11). We select a solution among the best solutions, then generate a local solution around the selected one by random walks.

• Evaluation of the new solution (line 13). The feature subset assessment function in CFS is utilized here to evaluate the new solution.

• Archive of the new solution (line 14-17). The new solution which meets our requirement needs to be archived here. After that, the loudness A_i^t and the rate r_i^t of pulse emission have to be updated using Eq. 6−Eq. 7.

• Update of the best solution (line 19-20). We compare the evaluation result of the archived solution and find the current best X_{best} until the iterations end.

Algorithm 1 CFS-BA approach for feature selection

Input: Training Dataset and Testing Dataset
Output: Selected Feature Subset X_{best}

1: Initialize a population of n bats $X_i = (x_{i1}, ...x_{iD})^T (i = 1, 2, ..., n)$ and v_i
2: Initialize frequency f_i, pulse emission rate r_i^t, and loudness A_i^t
3: Initialize $fit(X_i)$ (cf. Eq.1) and X_{best}
4: Initialize $fit_{temp}(i)$ and $X_{temp}(i)$ for solution storage
5: **while** $1 \leqslant t \leqslant$ Max no. of iterations **do**
6: **for** $i = 1$ to n **do**
7: Generate new f_i (cf. Eq.2)
8: Update X_i and v_i (cf. Eq.3 and Eq.4)
9: **if** $r_i^t <$ rand(0,1) **then**
10: Select a X_i from X_{best}
11: Generate a new X_{new} (cf. Eq.5)
12: **end if**
13: Calculate $fit(X_{new})$ (cf. Eq.1)
14: **if** $fit(X_i) \leqslant fit(X_{new})$ and N(0,1) $< A_i^t$ **then**
15: $fit_{temp}(i) \leftarrow fit(X_{new})$
16: $X_{temp}(i) \leftarrow X_{new}$
17: Decrease A_i^t and Increase r_i^t (cf. Eq.6 and Eq.7)
18: **end if**
19: **if** $fit(X_{new}) \geq$ Max of fit_{temp} **then**
20: $X_{best} \leftarrow X_{new}$
21: **end if**
22: **end for**
23: $t = t + 1$
24: **end while**

3.2. Ensemble classification

For ensemble learning, the classification methods usually combine multiple base classifiers in some way to produce better accuracy [28]. These classifiers are powerful to solve the same problem and collectively achieve a forecasting result with higher stability and accuracy by creating multiple independent models and combining them [53]. The classical reasons for employing ensemble classifiers to improve the effectiveness are representational issue, statistical reason, and computational reason. First, sometimes a single classifier is not qualified to obtain the best representation in the hypothesis space, therefore, it is necessary to combine independent classifiers to improve the predictive performance. Second, if the input dataset is not sufficient to train the learning algorithm, the result may lead to a weak or false hypothesis. In the last case, in order to produce a suitable hypothesis, an individual classifier could spend a significant amount of computing time, in which the procedure will be more likely to cause problems.

Bagging [15] and Boosting [29] are the two most popular algorithms in ensemble learning, usually producing good results in classification and being widely chosen to build many ensemble models. Moreover, the other well-known ensemble learning methods for improving the performance of classification are Voting [38] , Bayesian parameter averaging [32], and Stacking [40]. Likewise, ensemble methods have been shown to improve accuracy in many use cases, including intrusion detection. For example, the results in [1, 72, 77] proved that their proposed ensemble models produce better performance of IDS than the one using a single classifier. For security professionals, ensemble classifiers provide mechanisms that aid in analysis such as similarity to existing known malicious or benign samples.

Among decision tree algorithms, C4.5 has been widely used in the field of anomaly detection due to its high effi-

ciency and its simple characteristics. Meanwhile, random forest is the most representative algorithm among ensemble learning methods, and it is generally more robust and can achieve better performances than single decision trees. Moreover, Forest PA can use the strength of the entire feature space to generate trees with high accuracy. With its novel weight assignment strategy and bootstrap sampling, Forest PA generates highly diverse trees while retaining their higher individual accuracy. Therefore, C4.5, random forest, and Forest PA are selected to construct the ensemble for multi-class intrusion detection in this paper.

For bagging algorithm, the base classifiers are generated in parallel by bootstrap sampling. Boosting works by training a set of classifiers sequentially and combining them for prediction, where the later classifiers focus more on the mistakes of the earlier classifiers. However, sensitivity to noise leads to performance degradation when appearing wrong labels. Moreover, base classifiers usually are homogeneous in bagging and boosting, which will be not suitable for three different base classifiers (C4.5, random forest, and Forest PA) in this paper. Although stacking generates an ensemble of heterogeneous learners, it will bring enormous computational complexity when generating different level models. Compared to the above algorithms, in this paper, voting is more suitable for heterogeneous learners ensemble with lower computational complexity and less time overhead.

3.2.1. C4.5

C4.5 [73] is a typical decision tree algorithm which is developed based on the ID3 [37] algorithm. This algorithm passes through decision tree, visits each node and select optimal split based on the maximisation of the gain ratio, which is represented by the following formula:

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)} \tag{8}$$

In the process, an attribute with the highest information gain is chosen as splitting attribute for the node N. Information gain represents how much uncertainty in the set D is reduced after it is partitioned on attribute A, where the uncertainty can be calculated by entropy as:

$$Entropy(D) = -\sum_{x \in X} p(x) log_2 p(x) \tag{9}$$

where X is the set of classes in D and $p(x)$ is the proportion of number of elements in class x to the number of elements in set D.

Likewise, $SplitInfo$ is the term which describes how equally the attribute splits the data and can be calculated as:

$$SplitInfo(A) = -\sum_{j=1}^{n} \frac{|D_j|}{|D|} log_2 \left(\frac{|D_j|}{|D|} \right) \tag{10}$$

where $\frac{|D_j|}{|D|}$ represents the weight of the $j-$th partition in the set D.

Moreover, as an improvement of ID3 algorithm, C4.5 has the capability to model or classify both discrete and continuous attributes, and can ignore missing attribute values in a dataset.

3.2.2. Random Forest (RF)

Random Forest, proposed by Breimanis in [16], is another decision tree technique that operates by constructing multiple decision trees. It takes thousands of input variables without variable deletion and classifies them based on their significance. RF can be described as an ensemble of classification trees where every tree contributes with a single vote for the task of the most frequent class to the input data. Compared to other machine learning methods (e.g., support vector machine, artificial neural network), there are fewer parameters to be specified when running RF. In RF, a collection of individual tree structured classifiers can be defined as:

$$\left\{ h\left(x, \theta_k\right), k = 1, 2, ...i... \right\} \tag{11}$$

where h represents RF classifier, $\left\{\theta_k\right\}$ stands for random vectors distributed independently identical, and each tree has a vote for the most famous class at input variable x. The nature and dimensionality of θ depends on its use in tree construction.

The key to the success of RF is the creation of each decision tree that makes up the forest. A bootstrapped subset of the training dataset is created to train each tree in the forest. Due to this fact, on average, each tree makes use of around two-thirds of the training dataset. The unused elements are called by the Out Of Bag (OOB) samples, which are used for inner cross-validation to evaluate the classification accuracy of RF.

Significantly, RF has a low computational burden, and it is insensitive to the parameters and outliers. Besides, overfitting is less of an issue compared to individual decision tree, and there is no need to prune the trees which is a cumbersome task [27].

3.2.3. Forest by Penalizing Attributes (Forest PA)

Unlike some existing algorithms that use a subset of the non-class attributes, Forest PA [3] is an algorithm that builds a set of highly accurate decision trees by exploiting the strength of all non-class attributes available in a data set. At the same time, some weight-related concerns, such as weight assignment strategy and weight increment strategy, are taken into account in order to retain individually accurate and promote strong diversity.

For the weights of the attributes that appear in the latest tree, Forest PA will randomly update the weights for those attributes within a Weight-Range (WR), which can be defined as follows:

$$WR^{\lambda} = \begin{cases} [0.0000, e^{-\frac{1}{\lambda}}], & \lambda = 1 \\ [e^{-\frac{1}{\lambda - 1}} + \rho, e^{-\frac{1}{\lambda}}], & \lambda > 1 \end{cases} \tag{12}$$

where λ represents the level of the attribute and ρ is used to ensure the WR for different levels be non-overlapping. For example, if an attribute appears in the root node then its $\lambda = 1$. In the same way, if an attribute is tested at a child node of the root node then its $\lambda = 2$.

Moreover, in order to address the negative effect of retaining weights which are not present in the latest tree, Forest PA has a mechanism to gradually increase weights of the attributes that have not been tested in the subsequent trees. Let an attribute A_i is tested at Level ρ of the $T_{j-1}-$th tree with η height and its weight is ω_i. Then, the weight increment value σ_i of A_i is calculated as:

$$\sigma_i = \frac{1.0 - \omega_i}{(\eta + 1) - \lambda} \tag{13}$$

3.2.4. Vote

Vote is a meta algorithm which performs the decision process by applying several classifiers [18]. It uses the power of several individual classifiers and applies a combination rule for the decision. For example, minimum probability, maximum probability, majority voting, product of probabilities, and average of probabilities are different algorithms for combination rules. In order to deal with the multi-class classification, majority voting could not be chosen because the number of classes is more than that of base classifiers. In this paper, average of probabilities approach is used to make decision, where the class label is determined based on the maximum value of the average of predicted probabilities.

Suppose we have l classifiers $C = \{C_1, ..., C_l\}$, and c classes $\Omega = \{\omega_1, ..., \omega_c\}$. For instance, due to the above base classifiers considered in our experiment, l can be set to 3, and the value of c depends on the number of attack types. A classifier $C_i : R^n \rightarrow [0, 1]^c$ accepts an object $x \in R^n$ and outputs a vector $\left[P_{c_i}(\omega_1|x), ...P_{c_i}(\omega_c|x)\right]$, where $P_{c_i}(\omega_j|x)$ denotes the probability assigned by the classifier C_i that object x belongs to class ω_j. For each class ω_j, let m_j represents the mean of the probabilities assigned by the l classifiers, which can be calculated as:

$$m_j = \frac{1}{l} \sum_{i=1}^{l} P_{c_i}(\omega_j|x) \tag{14}$$

let $M = \left[m_1, ...m_c\right]$ be the set of mean probabilities for c classes. Then, x is assigned to the class ω_k if m_k is the maximum in M.

4. Evaluations and results

As stated before, this paper aims to develop an efficient intrusion detection system with high accuracy and low false alarms. For this purpose, a hybrid method, combined CFS and BA named CFS-BA, is performed to determine a subset of the original features in order to eliminate the irrelevant features, and improve the classification efficiency. In the classification step, an ensemble classifier combined three different algorithms, C4.5, RF, and Forest PA based on the AOP combination rule, is trained and tested based on three datasets. The experiments are performed by Weka 3.8.3 [92] on desktop PC with 3.6 GHz Intel Core i7-4790 processor and 16GB RAM.

4.1. Description of the benchmark datasets

During the evaluation of IDS, one of the challenges faced by researchers is finding a suitable dataset. Acquiring a real world dataset that represents the traffic flowing through the network without any sort of anonymization or modification is a problem that has been continuously encountered by the cybersecurity research community [5]. Even in the cases where the data is allowed to be released or shared for public use, it will be heavily anonymized or severely altered. This will cause a lot of the essential data components that are considered critical to the researchers to be lost or no longer reliable.

For this reason, many researchers have decided to use simulated datasets such as the most well-known KDDCup'99 dataset [75], or one of its contemporaries the NSL-KDD dataset [85]. Recently there has been a significant effort to try and develop data sets that are reflective of real world data. In 2015, Kolias et al. [48] published Aegean WiFi Intrusion Dataset (AWID) dataset, which includes real traces of both normal and intrusive 802.11 traffic. In addition, in 2017, the Canadian Institute for Cybersecurity (CIC) published an intrusion detection dataset named CIC-IDS2017 [78], which resembles the true real-world data packet capture (PCAPs). Therefore, in this paper, experiments are conducted based on the NSL-KDD, AWID, and CIC-IDS2017 datasets.

4.1.1. NSL-KDD dataset

The NSL-KDD dataset [85] was proposed in 2009 as a new revised version of the original dataset KDDCup'99 [51]. On the one hand, NSL-KDD retained the advantageous and challenging characteristics of KDDCup'99. On the other hand, it addressed some drawbacks inherited from the original dataset by eliminating redundant records, rationalizing the number of instances, and maintaining the diversity of selected samples. It is worth noting that the NSL-KDD dataset is compiled to maximize the difficulty of prediction, which constitutes its outstanding characteristics. In order to group the records into five difficulty levels, the initial dataset was evaluated using several benchmark classifiers, and each instance was annotated with the number of its successful predictions [12]. For each difficult level group, the amount of selected records is inversely proportional to the record percentages from the original KDDCup'99 dataset.

In this study, KDDTrain+, KDDTest+, and KDDTest-21 sets of the NSL-KDD dataset are used. The KDDTrain+ set contains total 125,973 instances comprising of 58,630 instances of attack traffic and 67,343 instances of normal traffic. Whereas, the KDDTest+ set contains total 22,544 instances, and as a subset of the KDDTest+ set, the KDDTest-21 set includes total 11,850 instances. Cross-validation is done on the the KDDTrain+ set in our experiments, and to extend this benchmark, we also consider a validation test

using simple hold-out (train-test) approach applied on KDDTest+ and KDDTest-21 sets. A detailed overview of the instances is shown in Table 1.

4.1.2. Aegean WiFi Intrusion Dataset (AWID) dataset

AWID was publicly available in 2015 as a collection of sets of WiFi network data, which contain real traces of both normal and intrusive data collected from real network environments [48]. Each record in the dataset is represented as a vector of 155 attributes, and each attribute has numeric or nominal values. Based on the number of target classes, the dataset can be classified into AWID-CLS dataset and AWID-ATK dataset. AWID-CLS dataset groups the instances into 4 main classes including normal, flooding, impersonation, and injection, while AWID-ATK dataset has 17 target classes that belong to the 4 main classes. On the other hand, based on the number of instances, all the datasets have two different versions: Full Set and Reduced Set. It is important to mention that these two versions are not related. The reduced set was collected independently from the full set at different times, with different tools, and in different environments.

For this research we have conducted experiments on the the reduced four class dataset (AWID-CLS-R-Tst) by using cross-validation method for classification purposes. In general, AWID-CLS-R-Tst set includes total 575,643 instances, and more detailed information about the numbers of specific attacks can be seen in Table 2.

4.1.3. CIC-IDS2017 dataset

The CIC-IDS2017 dataset was published by Canadian Institute for Cybersecurity (CIC) in 2017, it contains benign and the most up-to-date common attacks [78]. It also includes the results of the network traffic analysis using CICFlowMeter with labeled flows based on the time stamp, source and destination IPs, source and destination ports, protocols, and attacks (CSV files). This is one of the newest intrusion detection datasets, which covers necessary criteria with updated attacks such as DDoS, Brute Force, XSS, SQL Injection, Infiltration, Port Scan, and Botnet. In detail, this dataset contains 2,830,743 records devised on 8 files and each record includes 78 different features with its label.

In order to maintain the same order of magnitude of each dataset while taking into account the requirements of multi-classification, the Wednesday-workingHours set has been chosen for experiments through cross-validation method. This set includes total 691,406 instances belonging to 6 categories, and the static information of the set is given in Table 2.

4.2. Dataset preprocessing

Data preprocessing is the most time consuming and essential step in data mining. Realistic data typically comes from heterogeneous platforms and can be noisy, redundant, incomplete, and inconsistent [54]. Thus, it is important to transform raw data into a format suitable for analysis and knowledge discovery. Therefore, in this research, the preprocessing step involves data filtration, data transforming as well as data normalization.

Table 1
Statistics of the three sets of the NSL-KDD dataset.

Class	NSL-KDD		
	KDDTrain+	KDDTest+	KDDTest-21
Normal	67343	9711	2152
DoS	45927	7458	4342
PRB	11656	2421	2402
R2L	995	2754	2754
U2R	52	200	200
Attacks	58630	12833	9698
Total	125973	22544	11850

Table 2
Statistics of the AWID and CIC-IDS2017 datasets.

Class	AWID	Class	CICIDS-2017
	CLS-R-Tst		Wed.
Normal	530785	Normal	439683
Injection	16682	DoS slowloris	5796
Flooding	8097	DoS Slowhttptest	5499
Impersonation	20079	Dos Hulk	230124
		DoS GoldenEye	10293
		Heartbleed	11
Attacks	44858	Attacks	251723
Total	575643	Total	691406

4.2.1. Data filtration

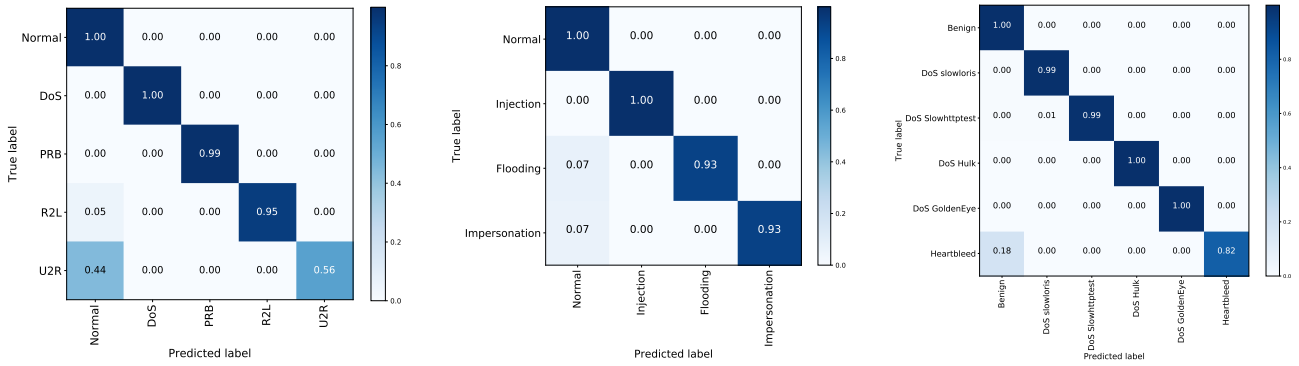
Due to the heterogeneity of the platforms, the raw data inevitably contain anomalous and redundant instances, which may have a negative influence on classification accuracy. In order to solve this problem, these records need to be removed from the dataset at the beginning of our experiments. For instance, the feature 'Fwd Header Length' appears twice in the CIC-IDS2017 dataset, and 'Flow Packets/s' includes abnormal values such as 'Infinity' and 'NaN'. Moreover, we have replaced missing values with zeroes and dropped out the features with constants values as they do not contribute to the class distinction. For example, the AWID-CLS-R-Tst set remains 84 features from the original 155 ones after data filtration.

4.2.2. Data transforming and normalization

The utilized datasets contain symbolic, continuous, and binary values. For instance, the feature 'protocol type' in the NSL-KDD datasets includes symbolic values such as: 'tcp', 'udp', and 'icmp'. As many classifiers accept only numerical values, the converting process is considered vital and has a significant impact on IDS accuracy. In this paper, we replace every single value with an integer in order to handle the symbolic features. Moreover, different scales among features can degrade the classification performance, for example, features that take on large numeric values, e.g., for the CIC-IDS2017 dataset, 'Flow Duration' can dominate the classifier's model relative to features with relatively small numeric

Table 3
Selected features for the NSL-KDD, AWID, and CIC-IDS2017 datasets.

NSL-KDD		AWID		CIC-IDS2017	
No.	Feature Name	No.	Feature Name	No.	Feature Name
3	service	1	frame.time_epoch	1	Destination Port
4	flag	15	radiotap.datarate	6	Total Length of Bwd Packets
5	src_bytes	16	radiotap.channel.freq	13	Bwd Packet Length Mean
6	dst_bytes	17	radiotap.channel.type.cck	15	Flow Bytes/s
14	root_shell	24	wlan.fc.frag	17	Flow IAT Mean
26	srv_serror_rate	29	wlan.duration	34	Bwd Header Length
29	same_srv_rate	32	wlan.ta	37	Min Packet Length
30	diff_srv_rate	35	wlan.frag	50	Down/Up Ratio
37	dst_host_srv_diff_host_rate			57	Subflow Bwd Bytes
39	dst_host_srv_serror_rate			58	Init_Win_bytes_forward
				59	Init_Win_bytes_backward
				67	Idle Std
				68	Idle Max



(a) NSL-KDD (b) AWID (c) CIC-IDS2017

Figure 2: Normalized confusion matrices based on the NSL-KDD, AWID, and CIC-IDS2017 datasets.

values such as 'Total Fwd Packets'. Accordingly, normalization is a 'scaling down' transformation which maps features onto a normalized range. A simple and fast approach called minimum-maximum method [49] is used in our experiments, which can be defined as:

$$\overline{x} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{15}$$

where x_{min} and x_{max} represent the minimum and maximum values of feature x.

4.3. Results and discussion

The performance of IDS is evaluated based on its capability of classifying network traffic into a correct type. In order to avoid the effect of data sampling when assessing the IDS, therefore, we conducted experiments by using repeated k-fold (kf) cross-validation method, and the value of k is considered as 10. In this paper, all the performance results reported are the average value of outputs from 10 iterations of 10f validation approach, and each experiment is repeated with different seed for avoiding biased results. More specifically, for each dataset, we provide the confusion matrix derived from the testing process of CFS-BA-Ensemble, and compare the performance of the proposed algorithm with no feature selection and some state-of-the-art methods in terms of several detection metrics, including Accuracy (Acc), precision, Detection Rate (DR), F-Measure, Attack Detection Rate (ADR), and False Alarm Rate (FAR). The mathematical calculations of the utilized evaluation metrics are explained in [25].

First, essential features are identified by utilizing the proposed CFS-BA approach to evaluate the integrity of the reduced feature subset in the feature selection stage. Then, candidate features are selected from the original ones for the next stage. Table 3 shows the numbers and names of selected features for NSL-KDD, AWID, and CIC-IDS2017 datasets. By implementing CFS-BA alone, the approach is seen to reduce the dimensionality drastically and eliminate the irrelevant features of the dataset. Finally, in order to significantly improve the predictive performance of IDS, an ensemble classifier which consists of three different decision tree classifiers is used in a vote algorithm.

Fig. 2(a), Fig. 2(b), and Fig. 2(c) separately indicate the multi-class classification performance of the proposed IDS with 10f cross-validation among the NSL-KDD, AWID, and

Table 4
Performance classification for feature selection based on NSL-KDD with 10f validation.

(a).The performance results based on the original features (41 features)							
Classifier	Acc	Precision	DR	F-Measure	ADR	FAR	MBT(s)
C4.5	0.941	0.945	0.941	0.943	0.913	0.035	16.91
RF	0.949	0.944	0.949	0.947	0.903	0.021	14.98
ForestPA	0.945	0.942	0.945	0.944	0.913	0.028	43.16
Ensemble	0.953	0.951	0.953	0.952	0.919	0.016	51.44
(b).The performance results based on the selected features using CFS-BA (10 features)							
Classifier	Acc	Precision	DR	F-Measure	ADR	FAR	MBT(s)
C4.5	0.988	0.987	0.988	0.988	0.986	0.012	2.93
RF	0.991	0.988	0.991	0.989	0.987	0.009	8.63
ForestPA	0.987	0.989	0.987	0.988	0.985	0.008	29.66
Ensemble	**0.998**	**0.998**	**0.998**	**0.998**	**0.997**	**0.001**	36.28

CIC-IDS2017 datasets. It is observed that the performance of most classifications is adequate, while several attacks can not be classified very well, such as 'U2R' and 'Heartbleed'. As seen in Table 1 and Table 2, the numbers of these instances are much less than others, which significantly affects the classification results of these attacks. In detail, there are only 11 instances with the label 'Heartbleed' out of 251,723 attack instances in the CIC-IDS2017 and 52 'U2R' instances in the KDDTrain+ set, which poses a challenge for the IDS to correctly classify them. In general, the proposed method is not focused on a specific class, it is proposed for selecting relevant features for all classes, which could not guarantee the performance of every type of attacks, especially some attacks with very few instances in the datasets. However, as the classification results for normal instances are pretty well among these datasets, the developed system can be used for intrusion detection.

4.3.1. Comparison with no feature selection

In order to evaluate the performance of the proposed IDS, we make a comparison between the proposed feature selection approach and without feature selection to distinguish attacks from benign instances. Thanks to the selection of relevant features by the proposed CFS-BA algorithm, the average values of these metrics, such as Acc, precision, DR, F-Measure, and ADR, have increased significantly.

Table 4 summarizes the performance based on the NSL-KDD dataset, which includes the results of the base and ensemble classifiers. It is indicated that the ensemble classifier is not good enough in some metrics without implementing feature selection. By contrast, the proposed CFS-BA-Ensemble method performs best on all the three sets. In detail, our model exhibits the highest accuracy of 0.998, F-Measure of 0.998, ADR of 0.997 and the lowest FAR of 0.001 based on the NSL-KDD dataset. As seen in Table 5, the proposed CFS-BA-Ensemble approach still achieves the best performance results in most respects on the AWID dataset, such as the highest accuracy of 0.995, the highest ADR of 0.959, and the lowest FAR of 0.002. Each base classifier using the selected feature exhibits higher accuracy and ADR than the ensemble classifier with the original features, which strongly proves the effectiveness of the proposed feature selection method. Similarly, the result of the comparison on the CIC-IDS2017 dataset is shown in Table 6, we observe that the performance of the proposed feature selection approach outperforms that of all features in every respect, and the CFS-BA-Ensemble approach achieves the highest accuracy rate of 0.999, DR of 0.999, and ADR of 0.999 with only 13 features, which also outperforms all other individual classifiers. In contrast, the best accuracy values of the C4.5, RF, and ForestPA classifiers are 0.983, 0.993, 0.988 using CFS-BA based feature selection method, respectively.

Furthermore, due to the dimensionality reduction of the subsets, the proposed CFS-BA-Ensemble model reduces the time overhead when it is applied to the feature selection and ensemble model. Table 4-Table 6 also show a comparison of the average model building time (MBT) consumed by single training based on the different numbers of features. For the NSL-KDD dataset, although it does not take much time to build the ensemble model on this set, the reduction still takes almost 30% of the original MBT when applying CFS-BA for feature selection. Due to the huge amount of data with high dimensionality, the ensemble method with the original features takes approximately 500s and 1000s for the AWID and CIC-IDS2017 datasets separately. Thanks to the feature selection method, the ensemble model with CFS-BA has mitigated the MBT considerably compared with that using all original features, all the MBTs of CFS-BA-Ensemble model on these three datasets have been restricted within 100s. Especially for the CIC-IDS2017 dataset, there is a significant reduction on the MBT of the ensemble classifier when using the CFS-BA based feature selection method, from 977.94s to 98.42s.

4.3.2. Comparison with other feature selection methods

As explained in Section 4.1, the benchmark datasets reflect a contemporary and complex threat environment. The increased number of attack classes and its highly imbalanced records pose a significant challenge to every machine learning approach. In order to further evaluate our proposed IDS model, we compare it with some well-known feature selection methods, namely IG (Information Gain) [11], IGR (Information Gain Ratio) [58], GA (Genetic Algorithm) [65],

Table 5
Performance classification for feature selection based on AWID with 10f validation.

(a).The performance results based on the original features (84 features)							
Classifier	Acc	Precision	DR	F-Measure	ADR	FAR	MBT(s)
C4.5	0.954	0.953	0.999	0.976	0.789	0.034	94.93
RF	0.979	0.982	0.996	0.989	0.783	0.004	142.84
ForestPA	0.966	0.982	0.981	0.981	0.784	0.019	435.11
Ensemble	0.982	0.982	0.999	0.990	0.784	0.002	488.46
(b).The performance results based on the selected features using CFS-BA (8 features)							
Classifier	Acc	Precision	DR	F-Measure	ADR	FAR	MBT(s)
C4.5	0.985	0.985	0.985	0.985	0.913	0.010	9.96
RF	0.992	0.992	0.992	0.992	0.945	0.004	26.51
ForestPA	0.990	0.989	0.990	0.989	0.902	0.003	79.93
Ensemble	**0.995**	**0.995**	0.995	**0.995**	**0.956**	**0.001**	92.62

Table 6
Performance classification for feature selection based on CIC-IDS2017 with 10f validation.

(a).The performance results based on the original features (78 features)							
Classifier	Acc	Precision	DR	F-Measure	ADR	FAR	MBT(s)
C4.5	0.960	0.961	0.984	0.973	0.918	0.016	212.59
RF	0.968	0.985	0.981	0.983	0.946	0.019	244.85
ForestPA	0.967	0.978	0.984	0.981	0.938	0.016	859.62
Ensemble	0.977	0.991	0.988	0.990	0.956	0.012	977.94
(b).The performance results based on the selected features using CFS-BA (13 features)							
Classifier	Acc	Precision	DR	F-Measure	ADR	FAR	MBT(s)
C4.5	0.983	0.996	0.989	0.992	0.974	0.011	32.02
RF	0.993	0.995	0.998	0.996	0.984	0.003	58.04
ForestPA	0.988	0.993	0.988	0.991	0.978	0.006	80.82
Ensemble	**0.999**	**0.999**	**0.999**	**0.999**	**0.999**	**0.001**	98.42

PSO (Particle Swarm Optimization) [98], and MBAFS (Modified Bat Algorithm for Feature Selection) [93] by conducting experiments based on these three datasets. Likewise, in this comparative study we use the common metrics in the context of Acc, F-Measure, ADR, and FAR. Especially, to figure out the efficiency of the proposed IDS, the comparison has also been done in terms of number of selected features and its selection time. Fig. 3 summarizes the average performance of our model as compared to the other feature selection methods based on the same proposed voting based ensemble classifier.

First, as shown in Fig. 3(a), the accuracy of our proposed model outperforms that of other algorithms based feature selection in every dataset, and the proposed CFS-BA-Ensemble approach achieves the highest average accuracy rate of 99.81%, 99.52%, and 99.89% over the NSL-KDD, AWID, and CIC-IDS2017 datasets, respectively. Similarly, Fig. 3(b) indicates that our proposed model exhibits better F-Measure than other feature selection methods on all datasets through extracting more relevant feature subsets, which increase the value of F-Measure from 0.969 to 0.998, 0.961 to 0.995, and 0.957 to 0.999 over these three datasets. Next, the attack detection rate, which stands for the accuracy rate for the attack classes, is an important indicator to evaluate the performance of an IDS. According to Fig. 3(c), it can be observed that the attack detection rate of our proposed model ranges from 95.64% to 99.92%, which significantly exceeds other feature selection methods based on any one of the five sets. Moreover, as Fig. 3(d) illustrates, our proposed CFS-BA based model achieves the lowest FAR values of 0.08%, 0.15%, and 0.12% based on the NSL-KDD, AWID, and CIC-IDS2017 datasets separately. In comparison with other feature selection methods, our proposed model has mitigated FAR considerably on each dataset and guaranteed the effectiveness of an IDS.

Notably, Fig. 3(e) and Fig. 3(f) exhibit the number of selected features using different algorithms and its selection time, which can indicate the efficiency of an IDS. When compared to IG and IGR, although the proposed method takes a little more time than them, CFS-BA selects less features, and as seen in Fig. 3(a), the accuracy of the proposed IDS is much higher than that of IG and IGR. For GA and PSO based feature selection methods, each of them obtains less features than CFS-BA on the AWID dataset, however, they need more feature selection time on all the five sets and could not achieve better detection accuracy. MBAFS, a modified bat algorithm for feature selection, is considered to be most similar to our feature selection method. According to Fig. 3(a)−3(d), MBAFS performs better than any of other methods in terms of these performance metrics except the proposed CFS-BA, and its performance is only slightly worse than our method. Since MBAFS introduces random bats and mutation mechanism, the search space is expanded in every iteration and the subset may be generated in any

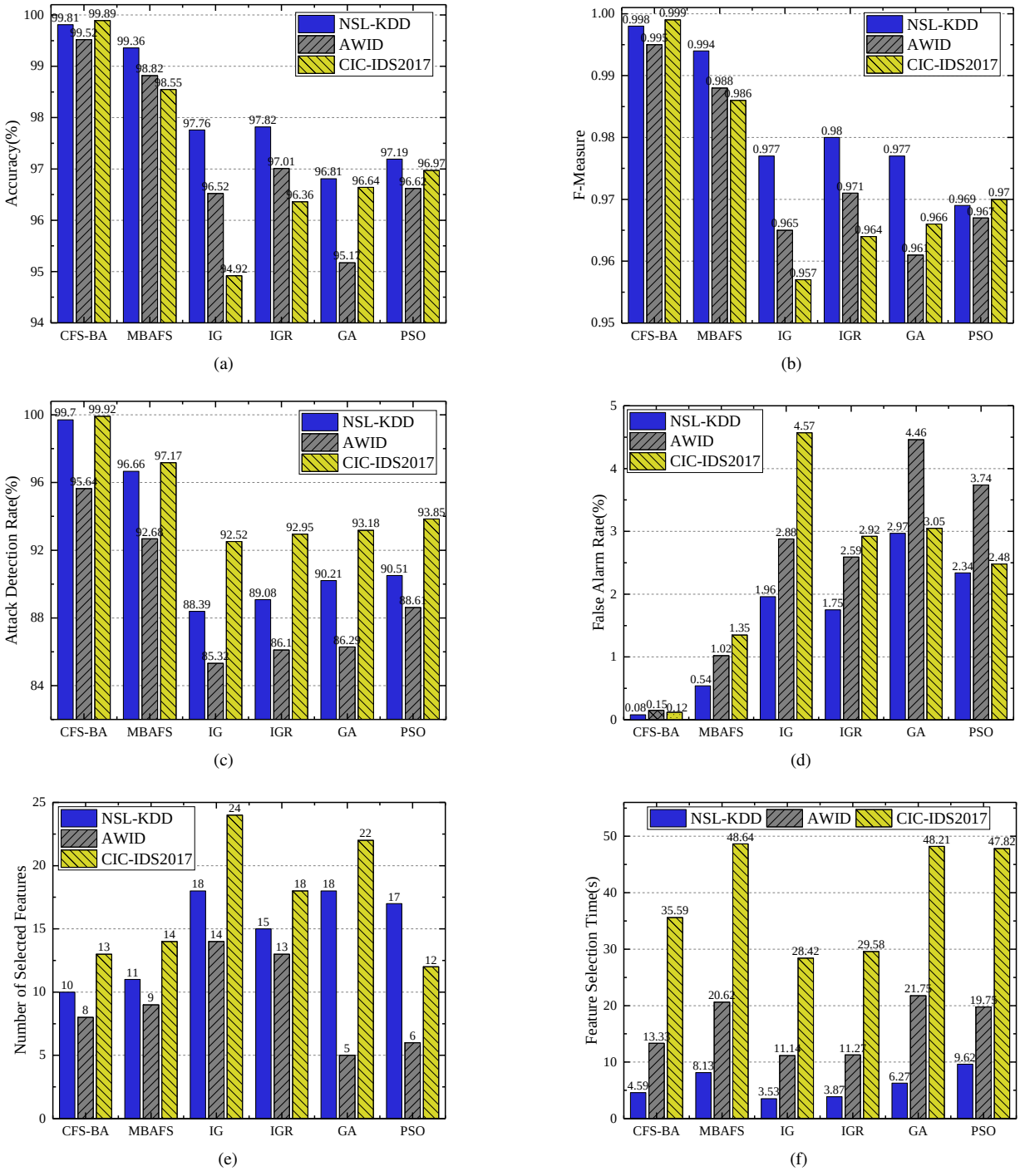


Figure 3: Comparison performance with other feature selection methods based on the three datasets.

uncertain direction. Therefore, as Fig. 3(e) illustrates, the subset selected by MBAFS contains one more feature than CFS-BA on any of these five sets, which may affect the performance of the IDS if the additional feature are not highly correlated. In addition, as the number of iterations before convergence increases, the feature selection time taken by MBAFS is more than ours, which can be seen in Fig. 3(f). In general, CFS-BA is superior to other feature selection methods in terms of performance and efficiency.

4.3.3. Comparison with other classifiers

Similarly, to evaluate the performance of our proposed ensemble classifier, experiments have been conducted using different classification algorithms among five preprocessed sets with CFS-BA feature selection. First, the proposed voting based ensemble classifier with AOP combination rule is chosen, and we construct a stacking classifier with C4.5, RF, and Forest PA as base classifiers, and Logistic Regression (LR) [47] as meta classifier to make a comparison with

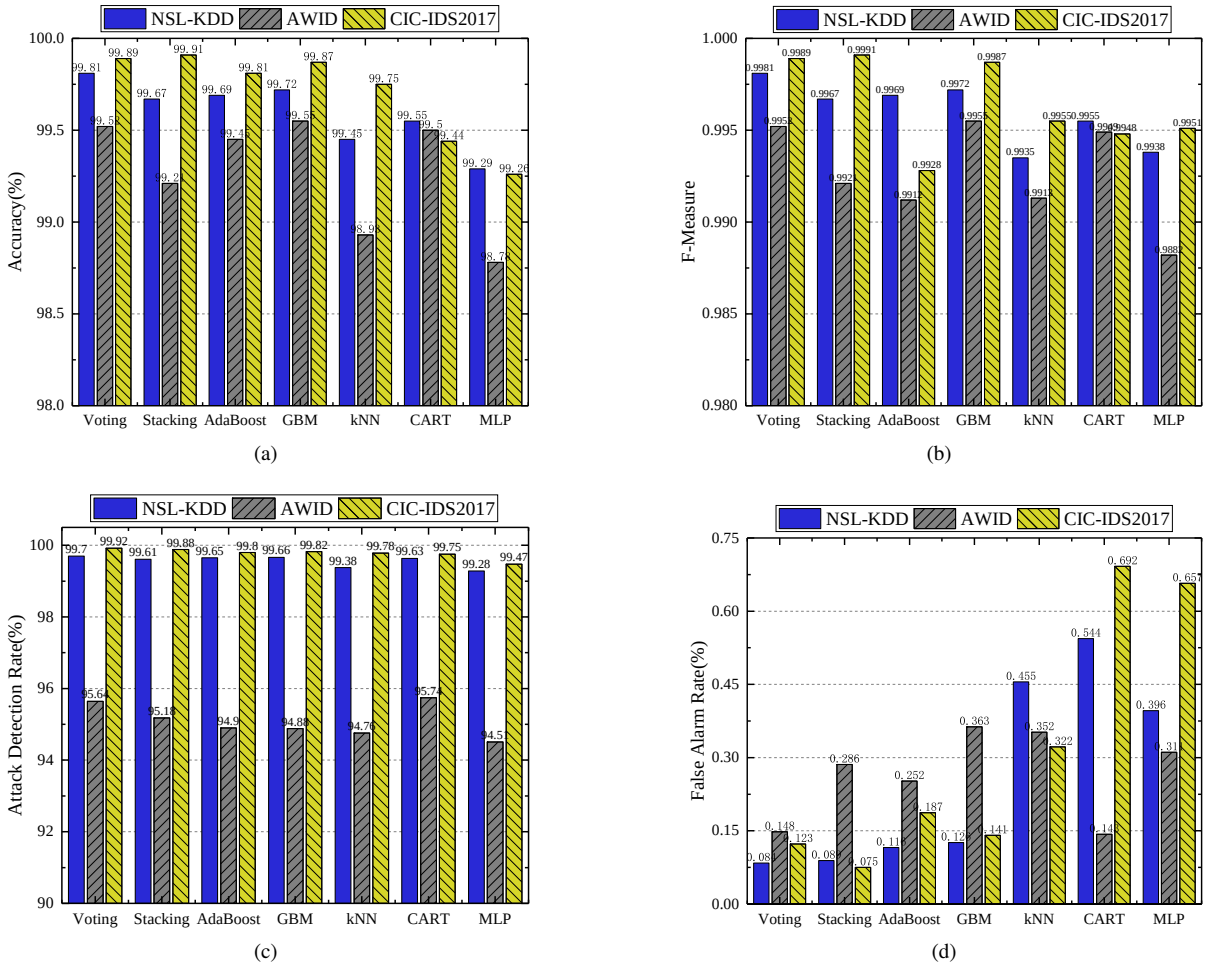


Figure 4: Comparison performance of per classifier across three datasets with 10f cross-validation.

our voting classifier. Second, we select some widely studied ensemble algorithms, such as AdaBoost (AB) [39] and Gradient Boosted Machine (GBM) [30] to make a comparison. Third, some single classifiers like k-Nearest Neighbor (kNN) [55], Classification and Regression Trees (CART) [17], and Multi-Layer Perceptron (MLP) [52] have been chosen as well.

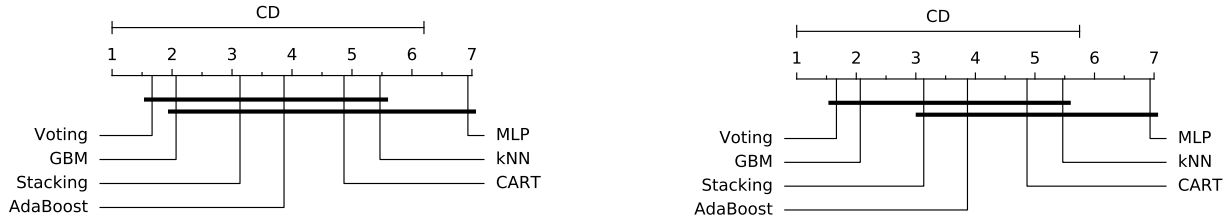
However, an algorithm may not be able to achieve good results over all datasets, which makes quite difficult to compare different algorithms over multiple datasets. In order to perform the comparison of classifiers in a correct way [88], it is necessary to statistically analyze the significance of the classifiers' performance. Thus, the significance tests has been considered essential to find whether the classifiers are significantly different from each other or not [22]. In order to give a thoroughly comparative study, two statistical significance tests, Friedman test [31] and Nemenyi post-hoc test [63], are adopted. In our case, the null-hypothesis is that there is no performance difference among different classifiers, and it can be rejected if at least one classifier is found significantly different from at least one other classifier. Since there are 7 classifiers to be compared in this case, Friedman test is chosen to prove whether at least one classifier performs

Table 7
Average ranks for 10f cross-validation across three datasets.

	Accuracy	F-Measure	ADR	FAR
Voting	1.667	1.967	1.467	1.867
Stacking	3.133	2.933	3.600	2.733
AdaBoost	3.867	5.033	3.733	3.333
GBM	2.067	2.367	3.400	4.000
kNN	5.467	5.233	5.533	5.533
CART	4.867	4.633	3.467	4.533
MLP	6.933	5.867	6.800	6.000

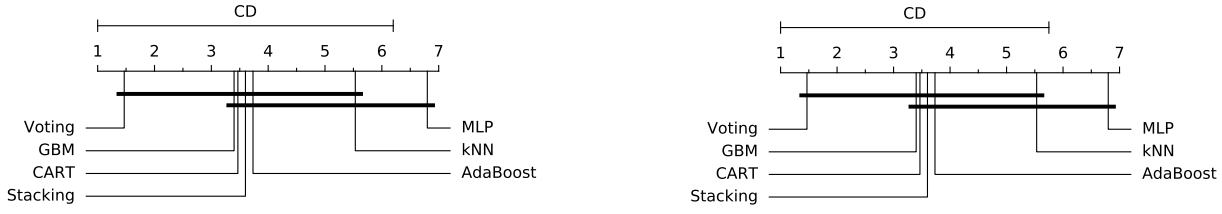
significantly better than another one over all datasets [84]. If the Friedman test reports a significant difference, to detect between which classifiers those differences appear, the Nemenyi post-hoc test will be then proceeded for pairwise multiple comparisons.

For the Friedman test, it ranks the algorithms for each dataset separately. For example, for a given dataset, the algorithm performing best gets the rank of 1, the second best gets rank 2, and so on. After that, Friedman test will do it again over another dataset until we obtain all rankings on all



(a) $\alpha = 0.05$ (b) $\alpha = 0.1$

Figure 5: Critical difference of all classifiers in term of accuracy metric.



(a) $\alpha = 0.05$ (b) $\alpha = 0.1$

Figure 6: Critical difference of all classifiers in term of attack detection rate metric.

datasets. Let r_{ij} be the rank of the j-th algorithm on the i-th dataset, where $i = 1, 2, ..., n$, and $j = 1, 2, ..., k$. Then, the average rank of j-th algorithm can be calculated as Eq. 16, and the Friedman statistic (F-Statistic) can be computed as Eq. 18, where χ_F^2 is calculated as Eq. 17.

$$R_j = \frac{1}{n}\sum_{i=1}^{n} r_{ij} \tag{16}$$

$$\chi_F^2 = \frac{12n}{k(k+1)}\left[\sum_{j=1}^{k} R_j^2 - \frac{k(k+1)^2}{4}\right] \tag{17}$$

$$F\text{-}Stastic = \frac{(n-1)\chi_F^2}{n(k-1)-\chi_F^2} \tag{18}$$

$$CD = q_\alpha\sqrt{\frac{k(k+1)}{6n}} \tag{19}$$

If the null-hypothesis is accepted, F-Statistic will be distributed according to the F-distribution for a given α with $k-1$ and $(k-1)(n-1)$ degrees of freedom. In this study, the values of k and n are set as 7 and 3, and two values of α (0.05 and 0.1) are considered. Otherwise, if we reject the null-hypothesis, then the Nemenyi post-hoc test will be performed to compare all classifiers with each other. The performance of two classifiers is significantly different when the difference between their average ranks is greater or equal to the critical difference (CD), where CD can be calculated as Eq. 19. In Eq. 19, k and n still represent the numbers of classifiers and datasets, and q_α is the critical value. Due to 7 classifiers are used for comparison, according to Table 5 (a) in [22], $q_{0.05} = 2.949$ and $q_{0.1} = 2.693$ in our case.

First, we analyze the average value of all mentioned metrics achieved with 10f cross-validation on the three datasets, which is shown in Fig. 4. It is observed from Fig. 4(a) that

Table 8
Friedman test statistics for 10f cross-validation.

	Accuracy	F-Measure	ADR	FAR
F-Statistic	6.5665	2.0810	3.3242	1.7904
p value	0.0029	0.1319	0.0363	0.1839
$\alpha = 0.05$	R	A	R	A
$\alpha = 0.1$	R	A	R	A

Voting, GBM, and Stacking outperform other classifiers in terms of accuracy (99.81%, 99.55%, and 99.91%) over NSL-KDD, AWID, and CIC-IDS2017 datasets separately but MLP achieves lowest accuracy values across all datasets. Similarly, Voting, GBM, and Stacking perform best in terms of F-Measure according to Fig. 4(b). However, kNN, MLP, and AdaBoost perform worst with F-Measure of 0.9935, 0.9882, and 0.9928. In terms of ADR metric, our proposed Voting based ensemble classifier performs best by achieving 99.7% and 99.92% on the NSL-KDD and CIC-IDS2017 datasets, and CART achieves the highest ADR value (95.74%) on the AWID dataset, whereas MLP is the worst performer over the three datasets. Fig. 4(d) indicates the average FAR values of all classifiers over all three datasets. Voting, CART, and Stacking separately exhibit the lowest FAR values of 0.084, 0.143, and 0.075 for NSL-KDD, AWID, and CIC-IDS2017 datasets. However, CART achieves the worst performance in terms of FAR for the NSL-KDD and CIC-IDS2017 datasets, and the worst performing classifier is GBM for the AWID dataset.

Then, the performance results are statistically assessed using Friedman and Nemenyi post-hoc test. According to experimental results, the average ranks of all the classifiers for 10f cross-validation are shown in Table 7. Thus, the F-Statistic and p value for each performance metric is com-

Table 9
Comparison of different combination rules under the NSL-KDD dataset based on accuracy.

	Average of probabilities	Majority voting	Product of probabilities	Minimum probability	Maximum probability
Normal	**99.90**	99.75	98.67	98.67	98.71
DoS	**99.92**	99.63	99.48	99.48	99.32
PRB	**99.48**	98.53	97.32	97.32	98.69
R2L	**94.57**	93.38	90.55	90.15	89.61
U2R	55.77	**57.69**	53.85	53.85	51.92

Table 10
Comparison of different combination rules under the AWID dataset based on accuracy.

	Average of probabilities	Majority voting	Product of probabilities	Minimum probability	Maximum probability
Normal	**99.85**	99.75	98.67	98.67	98.71
Injection	**99.98**	99.90	98.91	98.91	99.15
Flooding	**92.71**	90.16	88.92	89.29	86.45
Impersonation	93.21	91.89	89.45	89.45	**93.63**

puted, and Table 8 shows Friedman test statistics for 10f cross-validation results. From the results it is observed that p values under accuracy and ADR are less than 0.05, therefore the null-hypothesis is rejected and we can conclude that the performance of the classifiers is significantly different in terms of accuracy and ADR metrics. In order to detect which classifier pairs perform significantly different, Nemenyi post-hoc test is performed, and the results of the pairwise comparison over accuracy and ADR values are presented in Fig. 5 and Fig. 6. It is indicated that for accuracy metric the classifier's performance is highly significant (shown in Fig. 5(a)) in the case of Voting-MLP and less significant (shown in Fig. 5(b)) in the case of GBM-MLP, whereas remaining pairs are not significant. As shown in Fig. 6, the classifier's ADR measure is only found highly significant in case of Voting-MLP pair, while all other pairs are not significant. The experimental results show that Voting and GBM are suitable classifiers if the IDS demands high accuracy, and we highly suggest our Voting based ensemble classifier due to it also shows outstanding performance in terms of ADR metric.

4.3.4. Comparison with other combination rules

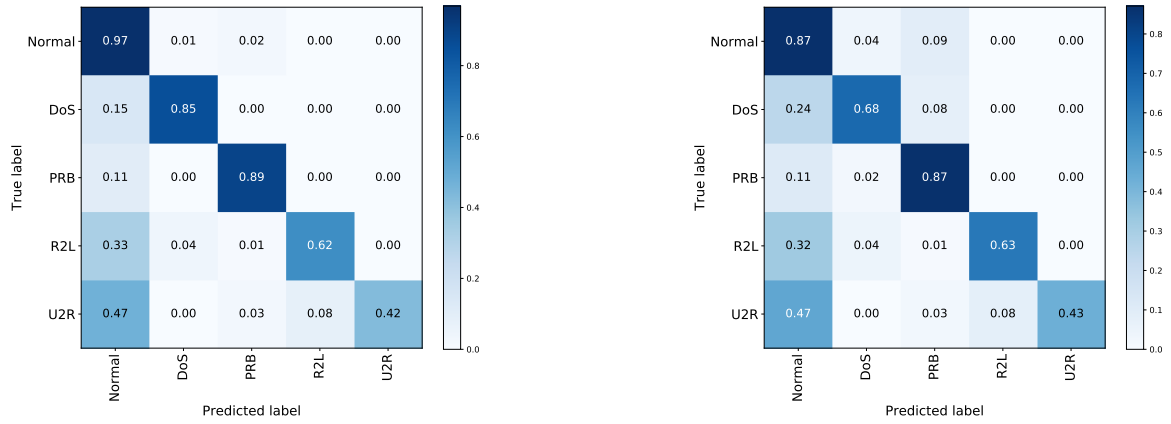
In this section, we explain the experimental results using CFS-BA-Ensemble approach with different combination rules we reached during the experiments. Similarly, the average accuracy values of outputs from 10 iterations of 10f validation approach are used for evaluation of the models. As mentioned in Section 3.2.4, minimum probability, maximum probability, majority voting, product of probabilities, and average of probabilities are common combination rules when using voting technique to construct an ensemble classifier. Therefore, in order to evaluate the multi-classification performance of these aggregation methods, from Table 9 to Table 11, we compare and analyze the average accuracy values of each combination rule for each attack type of different datasets.

Table 9 shows the accuracy values of each rule for the NSL-KDD dataset. For 'Normal', 'DoS', 'PRB', and 'R2L', the average of probabilities combination rule achieves the highest performance accuracy values of 99.90%, 99.92%, 99.48%, and 94.57% compared to the other combination rules. Although the performance of majority voting rule is better than average of probabilities in 'U2R' attack, the improvement in accuracy may lead to only one more attack instance identified due to the number of 'U2R' instances in the NSL-KDD dataset. Therefore, compared to majority voting, we prefer to use average of probabilities combination rule for more accurate classification on most instances of the NSL-KDD dataset.

According to the results shown in Table 10, the highest accuracy values of 99.85%, 99.98%, and 92.71% are obtained for 'Normal', 'Injection', and 'Flooding' with average of probabilities combination rule based on the AWID dataset. For the 'Impersonation' attack, the performance of maximum probability rule is slightly better than AOP but, when we consider most of the cases and the difference between accuracy values for two cases, we still suggest the average of probabilities rule for the AWID dataset. Similarly, as shown in Table 11, it is obviously observed that the best performance is still achieved with the average of probabilities combination rule for most classes of the CIC-IDS2017 dataset, such as 'Benign', 'DoS slowloris', 'DoS Hulk', and 'DoS GoldenEye'. The majority voting combination rule achieves the highest accuracy of 99.02% for 'DoS Slowhttptest', however, it produces the worst accuracy of 97.77% for 'DoS Hulk' and has no advantages for other types of attacks when compared to the average of probabilities combination rule. According to results in this table, it can also be indicated that the performance of the maximum probability combination rule for 'Heartbleed' (90.91%) is better than the other rules. However, as seen in Table 2, there are only 11 instances of 'Heartbleed' contained in the CIC-IDS2017 dataset. The advantage on the classification for 'Heartbleed' attacks cannot make up for its drawbacks on the other attacks when in comparison with the average of

Table 11
Comparison of different combination rules under CIC-IDS2017 dataset based on accuracy.

	Average of probabilities	Majority voting	Product of probabilities	Minimum probability	Maximum probability
Benign	**99.88**	99.73	97.80	97.82	97.34
DoS slowloris	**99.26**	98.34	97.48	97.48	96.64
DoS Slowhttptest	98.95	**99.02**	97.29	97.29	96.02
Dos Hulk	**99.97**	97.77	98.56	97.80	97.77
DoS GoldenEye	**99.59**	99.10	97.64	97.64	97.15
Heartbleed	81.82	81.82	72.73	72.73	**90.91**



(a) KDDTest+ (b) KDDTest-21

Figure 7: Normalized confusion matrices based on the KDDTest+ and KDDTest-21 sets.

probabilities combination rule.

Based on the experimental results on the three datasets, therefore, in this study, we decided to apply the AOP as combination rule in our proposed CFS-BA-Ensemble model.

4.3.5. Comparison with the state of the art methods

NSL-KDD dataset provides some different test sets, such as KDDTest+ and KDDTest-21 for benchmarking the machine learning algorithms. To evaluate the proposed model on unseen attacks, we have conducted experiments with the proposed CFS-BA-Ensemble model by using the datasets KDDTrain+ for training and KDDTest+ and KDDTest-21 for testing. Fig. 7 indicates the multi-class classification performance of the proposed IDS based on the KDDTest+ and KDDTest-21 test sets. As seen in Fig. 7(a), the proposed model can achieve the accuracy of 0.97 for normal traffic, whereas 0.85 and 0.89 for 'DoS' and 'PRB' attacks based on the KDDTest+ set. Similarly, it can be observed from Fig. 7(b) that our method can obtain the accuracy of 0.87, 0.68, and 0.87 for 'Normal', 'DoS', and 'PRB' instances. Moreover, our proposed method obtains accuracy values of 0.62 and 0.42 for the 'R2L' and 'U2R' attacks on KDDTest+, which are similar to that on KDDTest-21. On one hand, the proposed model has been trained on the KDDTrain+ set, where the 'R2L' and 'U2R' instances occupy the smallest proportion of all instances. On the other hand, the 'R2L' and 'U2R' instances are the same in the KDDTest+ and KDDTest-21 set according to Table 1, so the classification performance for them is almost the same.

To extend the benchmark, we also have compared our CFS-BA-Ensemble with the performance achieved by previous studies that use the datasets KDDTest+ and KDDTest-21 for testing. The comparison results with some of the existing approaches on these two sets are shown in Table 12. The highest detection accuracy is achieved by the proposed approach based on the experimental results on KDDTest+, which outperforms the other recent IDS techniques, including FSSL [9], FSSL-EL [34], and TSE-IDS [82]. Besides having superior detection accuracy, the proposed method also outperforms significantly other approaches in terms of detection rate metric. Even though EM-FS [72] performs best in terms of FAR metric, it only achieves the accuracy of 84.25% based on 35 features. However, our proposed method can obtain higher accuracy of 87.37% with FAR of 3.19% based on only 10 features, which still outperforms EM-FS to some extent. Moreover, according to the experimental results tested on the KDDTest-21 set, the proposed approach can achieve the accuracy of 73.57%, DR of 73.6%, and FAR of 12.92% with a 10-feature subset, which clearly outperforms other state of the art classifiers in terms of all the evaluation metrics shown in Table 12.

In order to further interpret the advantages of the proposed approach, in this section, some state of the art studies applied on each dataset are compared with our proposed CFS-BA-Ensemble method. More precisely, the comparison includes the feature selection techniques, the classifica-

Table 12
Comparison results with other existing methods on KDDTest+ and KDDTest-21.

Method	Dataset	Feature selection	Classification method	# Features	Acc(%)	DR(%)	FAR(%)
NBTree [85]	KDDTest+	N/A	NB Tree	41	82.02	N/A	N/A
Fuzzy [50]	KDDTest+	N/A	Fuzzy classifiers	41	82.74	86.7	3.9
SVM [71]	KDDTest+	N/A	SVM	41	82.37	82	15
FS+GAR-forest [43]	KDDTest+	Symmetrical	GAR-forest	32	85.056	85.1	12.2
TDTC [64]	KDDTest+	LDA+PCA	NB+CF-kNN	N/A	84.86	N/A	4.86
FSSL [9]	KDDTest+	Clustering	FSSL	41	84.12	N/A	N/A
EM-FS [72]	KDDTest+	IGR	Bagging(C4.5)	35	84.25	N/A	**2.79**
FSSL-EL [34]	KDDTest+	PCA	Ensemble(CART)	20	84.54	N/A	5.31
TSE-IDS [82]	KDDTest+	Hybrid	Two-stage Ensemble	37	85.797	86.8	11.7
NBTree [85]	KDDTest-21	N/A	NB Tree	41	66.16	N/A	N/A
FSSL [9]	KDDTest-21	Clustering	FSSL	41	68.82	N/A	N/A
FSSL-EL [34]	KDDTest-21	PCA	Ensemble(CART)	20	71.29	N/A	20.35
TSE-IDS [82]	KDDTest-21	Hybrid	Two-stage Ensemble	37	72.52	72.5	18.00
Proposed	KDDTest+	CFS-BA	Voting(C4.5,RF,ForestPA)	**10**	**87.37**	**87.4**	3.19
Proposed	KDDTest-21	CFS-BA	Voting(C4.5,RF,ForestPA)	**10**	**73.57**	**73.6**	**12.92**

N/A: name not available.

tion method, the number of selected features, accuracy, FAR, and DR for intrusion detection. Furthermore, to compare more fairly with these existing methods, we ensure that the datasets used by these methods, even if the specific version of the datasets, are the same as ours. Similarly, these existing methods all adopt cross-validation approach. The results of our proposed method compared to the existing approaches in each dataset are presented in Tables 13 and 14.

Although the multi-class classification performance of our proposed method has been proven through experiments, to provide more reference for the readers, we still compare the results of our CFS-BA-Ensemble method with other earlier researches in binary classification based on NSL-KDD, AWID, and CIC-IDS2017 datasets, which is shown in Table 13. First of all, it can be seen in Table 13 that our proposed model outperforms other similar ensemble classifiers, such as FS-EL [83], XGBoost-IDS [13], and TSE-IDS [82] when using 10f cross-validation as a validation technique. There are also some deep learning methods for IDS in the current literature such as DEMISe [69], DeepWindow [79], and HELAD [99]. Even though HELAD performs very well in terms of accuracy rate, DR, and FAR, the proposed method can still achieve a better performance compared with these methods. When compared to these binary classification methods, the proposed CFS-BA-Ensemble method has a slight advantage on accuracy and DR against all of them applied on the three datasets. And although some of previous studies did not provide data for FAR, it achieves extremely low FAR by 0.08%, 0.15%, and 0.12% across all datasets, which is a useful property for real-world IDSs. Additionally, the proposed method may not be the best by considering the number of selected features, however, it is worth selecting only a few more features to effectively improve the performance of the classifier.

As shown in the following Table 14, we then compare the performance of our proposed method, CFS-BA-Ensemble, with these existing methods for multi-class attack classification. For the NSL-KDD dataset, OR+FS [76] exhibits a high accuracy value of 99.43% based on the selected 6 features, however, the proposed approach achieves higher accuracy by 0.38% with 4 more features needed. Similarly, in contrast to earlier ensemble methods on AWID and CIC-IDS2017 datasets, like MVWIDS [7], ELWNIDS [87], and DARE [10], the proposed approach achieves better performances in accuracy and DR while limiting FAR at a lower level.

In general, the proposed method achieves promising results in the context of accuracy rate, DR, and FAR across the NSL-KDD, AWID, and CIC-IDS2017 datasets in comparison with the earlier studies. However, it should be noted that Table 13 and 14 just provide a snapshot of comparison between our proposed model and the state of the art methods in intrusion detection problem. Thus, there might be some limitations in this comparison. For example, data processing method, parameter setting of the algorithm, and many other experimental factors are all unknown for the existing techniques. Although we have tried to make as fair a comparison as possible, we cannot claim that our proposed intrusion detection model always performs better when compared to any of the other methods in the context of intrusion detection. However, according to the comparison results indicated in Table 13 and 14, our proposed CFS-BA-Ensemble method still provides a powerful competitive advantage in the intrusion detection domain.

5. Conclusions

Although many machine learning approaches have been proposed to increase the efficacy of IDSs, it is still a problem for existing intrusion detection algorithms to achieve good performance. In this paper, to deal with the high-dimensional and unbalanced network traffic, we propose a novel intrusion detection framework, which is based on the feature selection

Table 13
Comparison results with other state of the art binary classification approaches.

Method	Dataset	Feature selection	Classification method	# Features	Acc(%)	DR(%)	FAR(%)
FRCM [66]	KDDTrain+	Greedy Stepwise	Fuzzy Ownership NN	11	99.6356	99.6145	0.309
FS-EL [83]	KDDTrain+	CFS+PSO	Boosting(CART)	11	99.7285	99.77	N/A
OneR-BN [67]	KDDTrain+	OneR	BN+TAN	N/A	99.7412	99.7646	0.2792
TSE-IDS [82]	KDDTrain+	Hybrid	Two-stage Ensemble	37	96.388	N/A	N/A
DEMISe [69]	AWID-CLS-R	Autoencoder,MI	RBFC	**7**	98	99.04	3
SSLA [74]	AWID-CLS-R	N/A	Ladder Network	95	99.28	99.45	0.23
DARE [10]	CIC-IDS2017(Wed.)	N/A	One-class SVM	**10**	66	57	N/A
XGBoost-IDS [13]	CIC-IDS2017(Wed.)	N/A	XGBoost	80	91.36	98.38	12
ZED-IDS [19]	CIC-IDS2017(Wed.)	N/A	Autoencoder	83	95.73	95.82	4.32
DeepWindow [79]	CIC-IDS2017(Wed.)	MI+MIC	LSTM	N/A	99.5	99.4	N/A
HELAD [99]	CIC-IDS2017(Wed.)	DIS+DBN	Autoencoder+LSTM	50	99.58	99.58	2.15
Proposed	KDDTrain+	CFS-BA	Voting(C4.5,RF,ForestPA)	**10**	**99.81**	**99.8**	**0.08**
Proposed	AWID-CLS-R	CFS-BA	Voting(C4.5,RF,ForestPA)	8	**99.52**	**99.5**	**0.15**
Proposed	CIC-IDS2017(Wed.)	CFS-BA	Voting(C4.5,RF,ForestPA)	13	**99.89**	**99.9**	**0.12**

N/A: name not available.

Table 14
Comparison results with other state of the art multi-class classification approaches.

Method	Dataset	Feature selection	Classification method	# Features	Acc(%)	DR(%)	FAR(%)
AR-C4.5 [20]	KDDTrain+	Attribute Ratio	C4.5	22	99.794	N/A	N/A
SS-BN [97]	KDDTrain+	Sequential Search	Bayesian Network	11	98.98	N/A	0.60
OR+FS [76]	KDDTrain+	IQR,CFS+BFS	kNN	**6**	99.43	N/A	N/A
IG-RT [86]	AWID-CLS-R	IG	Random Tree	41	95.12	92	0.538
MVWIDS [7]	AWID-CLS-R	N/A	Voting(ET,RF,Bagging)	20	96.32	96	N/A
ELWNIDS [87]	AWID-CLS-R	CFS	RF	18	99.096	N/A	0.248
DARE [10]	CIC-IDS2017(Wed.)	N/A	RF	**10**	98	98	N/A
DeepDetect [8]	CIC-IDS2017(Wed.)	N/A	ANN	80	98.694	98.694	1.882
XGBoost-IDS [13]	CIC-IDS2017(Wed.)	N/A	XGBoost	80	99.54	99.54	0.15
Proposed	KDDTrain+	CFS-BA	Voting(C4.5,RF,ForestPA)	10	**99.81**	**99.8**	**0.08**
Proposed	AWID-CLS-R	CFS-BA	Voting(C4.5,RF,ForestPA)	**8**	**99.52**	**99.5**	**0.15**
Proposed	CIC-IDS2017(Wed.)	CFS-BA	Voting(C4.5,RF,ForestPA)	13	**99.89**	**99.9**	**0.12**

N/A: name not available.

and ensemble learning techniques. First, we propose a CFS-BA algorithm with the aim of selecting the optimal subset based on the correlation between features. Then, the ensemble classifier based on C4.5, RF, and ForestPA with the AOP rule is introduced to construct the classification model. Finally, the proposed IDS is evaluated by 10f cross-validation over three intrusion detection datasets.

The experimental results are promising with an accuracy of classification equal to 99.81%, 99.8% DR and 0.08% FAR with a subset of 10 features for the NSL-KDD dataset, and the obtained results for the AWID provide accuracy of 99.52% and 0.15% FAR with a subset composed of only 8 features. Remarkably, our model achieves the highest accuracy of 99.89% and DR of 99.9% on the subset of 13 features for the CIC-IDS2017 dataset. Then, the comparison with no feature selection method demonstrates encouraging performance on several metrics, and it should be noted that our proposal sharply reduces the MBT from 977.94s to 98.42s on the CIC-IDS2017 dataset. Our method also outperforms related feature selection approaches in terms of Acc, F-Measure, ADR, and efficiency while limiting FAR at relatively low levels. In addition, our solution shows outstanding performance in terms of ADR metric when compared to other classification algorithms, and the comparison results with the state of the art methods indicate that the proposed CFS-BA-Ensemble method can provide a powerful competitive advantage in the intrusion detection domain. Although the proposed CFS-BA Ensemble method has indicated superior performance, in the future work, its capability could be further improved to deal with rare attacks from the massive network traffic.

Acknowledgment

This work is supported by National Key Research and Development Program of China under Grant No. 2018YFB 1800602 and No. 2017YFB0801703, CERNET Innovation Project (NGIICS20190101, NGII20170406), and Ministry of Education-China Mobile Research Fund Project (MCM20 180506).

References

[1] Abdullah, M., Balamash, A., Alshannaq, A., Almabdy, S., 2018. Enhanced intrusion detection system using feature selection method and ensemble learning algorithms. International Journal of Computer Science and Information Security (IJCSIS) 16.

[2] Acharya, N., Singh, S., 2018. An iwd-based feature selection method for intrusion detection system. Soft Computing 22, 4407–4416. doi:10.1007/s00500-017-2635-2.

[3] Adnan, M.N., Islam, M.Z., 2017. Forest pa: Constructing a decision forest by penalizing attributes used in previous trees. Expert Systems with Applications 89, 389–403. doi:10.1016/j.eswa.2017.08.002.

[4] Al-Jarrah, O.Y., Alhussein, O., Yoo, P.D., Muhaidat, S., Taha, K., Kim, K., 2015. Data randomization and cluster-based partitioning for botnet intrusion detection. IEEE transactions on cybernetics 46, 1796–1806. doi:10.1109/TCYB.2015.2490802.

[5] Aldwairi, T., Perera, D., Novotny, M.A., 2018. An evaluation of the performance of restricted boltzmann machines as a model for anomaly network intrusion detection. Computer Networks 144, 111–119. doi:10.1016/j.comnet.2018.07.025.

[6] Aljawarneh, S., Aldwairi, M., Yassein, M.B., 2018. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. Journal of Computational Science 25, 152–160. doi:10.1016/j.jocs.2017.03.006.

[7] Alotaibi, B., Elleithy, K., 2016. A majority voting technique for wireless intrusion detection systems, in: 2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT), IEEE. pp. 1–6. doi:10.1109/LISAT.2016.7494133.

[8] Asad, M., Asim, M., Javed, T., Beg, M.O., Mujtaba, H., Abbas, S., 2019. Deepdetect: Detection of distributed denial of service attacks using deep learning. The Computer Journal doi:10.1093/comjnl/bxz064.

[9] Ashfaq, R.A.R., Wang, X.Z., Huang, J.Z., Abbas, H., He, Y.L., 2017. Fuzziness based semi-supervised learning approach for intrusion detection system. Information Sciences 378, 484–497. doi:10.1016/j.ins.2016.04.019.

[10] Attak, H., Combalia, M., Gardikis, G., Gastón, B., Jacquin, L., Katsianis, D., Litke, A., Papadakis, N., Papadopoulos, D., Pastor, A., et al., 2018. Application of distributed computing and machine learning technologies to cybersecurity. Space 2, I2CAT.

[11] Azhagusundari, B., Thanamani, A.S., 2013. Feature selection based on information gain. International Journal of Innovative Technology and Exploring Engineering (IJITEE) 2, 18–21.

[12] Bala, R., Nagpal, R., 2019. A review on kdd cup99 and nsl nsl-kdd dataset. International Journal of Advanced Research in Computer Science 10.

[13] Bansal, A., Kaur, S., 2018. Extreme gradient boosting based tuning for classification in intrusion detection systems, in: International Conference on Advances in Computing and Data Sciences, Springer. pp. 372–380. doi:10.1007/978-981-13-1810-8_37.

[14] Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A., 2016. Feature selection for high-dimensional data. Progress in Artificial Intelligence 5, 65–75. doi:10.1007/s13748-015-0080-y.

[15] Breiman, L., 1996. Bagging predictors. Machine learning 24, 123–140. doi:10.1007/BF00058655.

[16] Breiman, L., 2001. Random forests. Machine learning 45, 5–32. doi:10.1023/A:1010933404324.

[17] Breiman, L., 2017. Classification and regression trees. Routledge. doi:10.1201/9781315139470.

[18] Catal, C., Nangir, M., 2017. A sentiment classification model based on multiple classifiers. Applied Soft Computing 50, 135–141. doi:10.1016/j.asoc.2016.11.022.

[19] Catillo, M., Rak, M., Villano, U., 2019. Discovery of dos attacks by the zed-ids anomaly detector. Journal of High Speed Networks , 1–17doi:10.3233/JHS-190620.

[20] Chae, H.s., Choi, S.H., 2014. Feature selection for efficient intrusion detection using attribute ratio. International Journal of Computers and Communications 8.

[21] Chen, X.Y., Ma, L.Z., Chu, N., Zhou, M., Hu, Y., 2013. Classification and progression based on cfs-ga and c5. 0 boost decision tree of tcm zheng in chronic hepatitis b. Evidence-Based Complementary and Alternative Medicine 2013. doi:10.1155/2013/695937.

[22] Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. Journal of Machine learning research 7, 1–30.

[23] Du, M., Wang, K., Chen, Y., Wang, X., Sun, Y., 2018a. Big data privacy preserving in multi-access edge computing for heterogeneous internet of things. IEEE Communications Magazine 56, 62–67. doi:10.1109/MCOM.2018.1701148.

[24] Du, M., Wang, K., Xia, Z., Zhang, Y., 2018b. Differential privacy preserving of training model in wireless big data with edge computing. IEEE Transactions on Big Data doi:10.1109/TBDATA.2018.2829886.

[25] Elhag, S., Fernández, A., Altalhi, A., Alshomrani, S., Herrera, F., 2019. A multi-objective evolutionary fuzzy system to obtain a broad and accurate set of solutions in intrusion detection systems. Soft Computing 23, 1321–1336. doi:10.1007/s00500-017-2856-4.

[26] Elhag, S., Fernández, A., Bawakid, A., Alshomrani, S., Herrera, F., 2015. On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on intrusion detection systems. Expert Systems with Applications 42, 193–202. doi:10.1016/j.eswa.2014.08.002.

[27] Feng, Q., Liu, J., Gong, J., 2015. Uav remote sensing for urban vegetation mapping using random forest and texture analysis. Remote sensing 7, 1074–1094. doi:10.3390/rs70101074.

[28] Feng, X., Xiao, Z., Zhong, B., Qiu, J., Dong, Y., 2018. Dynamic ensemble classification for credit scoring using soft probability. Applied Soft Computing 65, 139–151. doi:10.1016/j.asoc.2018.01.021.

[29] Freund, Y., Schapire, R.E., et al., 1996. Experiments with a new boosting algorithm, in: icml, Citeseer. pp. 148–156.

[30] Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. Annals of statistics , 1189–1232.

[31] Friedman, M., 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the american statistical association 32, 675–701. doi:10.2307/2279372.

[32] Friston, K., Stephan, K., Li, B., Daunizeau, J., 2010. Generalised filtering. Mathematical Problems in Engineering 2010. doi:10.1155/2010/621670.

[33] Gaikwad, D., Thool, R.C., 2015. Intrusion detection system using bagging ensemble method of machine learning, in: 2015 International Conference on Computing Communication Control and Automation, IEEE. pp. 291–295. doi:10.1109/ICCUBEA.2015.61.

[34] Gao, Y., Liu, Y., Jin, Y., Chen, J., Wu, H., 2018. A novel semi-supervised learning approach for network intrusion detection on cloud-based robotic system. IEEE Access 6, 50927–50938. doi:10.1109/ACCESS.2018.2868171.

[35] Hajisalem, V., Babaie, S., 2018. A hybrid intrusion detection system based on abc-afs algorithm for misuse and anomaly detection. Computer Networks 136, 37–50. doi:10.1016/j.comnet.2018.02.028.

[36] Hota, H., Shrivas, A.K., 2014. Decision tree techniques applied on nsl-kdd data and its comparison with various feature selection techniques, in: Advanced Computing, Networking and Informatics-Volume 1. Springer, pp. 205–211. doi:10.1007/978-3-319-07353-8_24.

[37] Hssina, B., Merbouha, A., Ezzikouri, H., Erritali, M., 2014. A comparative study of decision tree id3 and c4. 5. International Journal of Advanced Computer Science and Applications 4, 0–0.

[38] Hu, J., 2018. An approach to eeg-based gender recognition using entropy measurement methods. Knowledge-Based Systems 140, 134–141. doi:10.1016/j.knosys.2017.10.032.

[39] Hu, W., Hu, W., Maybank, S., 2008. Adaboost-based algorithm for network intrusion detection. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 38, 577–583. doi:10.1109/TSMCB.2007.914695.

[40] Hung, C., Chen, J.H., 2009. A selective ensemble based on expected probabilities for bankruptcy prediction. Expert systems with applications 36, 5297–5303. doi:10.1016/j.eswa.2008.06.068.

[41] Jabbar, M., Aluvalu, R., Reddy, S.S.S., 2017. Cluster based ensemble classification for intrusion detection system, in: Proceedings of the 9th International Conference on Machine Learning and Computing,

pp. 253–257. doi:10.1145/3055635.3056595.

[42] Joldzic, O., Djuric, Z., Vuletic, P., 2016. A transparent and scalable anomaly-based dos detection method. Computer Networks 104, 27–42. doi:10.1016/j.comnet.2016.05.004.

[43] Kanakarajan, N.K., Muniasamy, K., 2016. Improving the accuracy of intrusion detection using gar-forest with feature selection, in: Proceedings of the 4th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA) 2015, Springer. pp. 539–547. doi:10.1007/978-81-322-2695-6_45.

[44] Khammassi, C., Krichen, S., 2017. A ga-lr wrapper approach for feature selection in network intrusion detection. computers & security 70, 255–277. doi:10.1016/j.cose.2017.06.005.

[45] Khan, M.A., Karim, M., Kim, Y., et al., 2019. A scalable and hybrid intrusion detection system based on the convolutional-lstm network. Symmetry 11, 583. doi:10.3390/sym11040583.

[46] Kim, J., Kim, J., Thu, H.L.T., Kim, H., 2016. Long short term memory recurrent neural network classifier for intrusion detection, in: 2016 International Conference on Platform Technology and Service (PlatCon), IEEE. pp. 1–5. doi:10.1109/PlatCon.2016.7456805.

[47] Kleinbaum, D.G., Klein, M., 2010. Logistic Regression. Statistics for Biology and Health, Springer Science+Business Media, LLC. doi:10.1007/978-1-4419-1742-3.

[48] Kolias, C., Kambourakis, G., Stavrou, A., Gritzalis, S., 2015. Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. IEEE Communications Surveys & Tutorials 18, 184–208. doi:10.1109/COMST.2015.2402161.

[49] Kotsiantis, S., Kanellopoulos, D., Pintelas, P., 2006. Data preprocessing for supervised leaning. International Journal of Computer Science 1, 111–117.

[50] Krömer, P., Platoš, J., Snášel, V., Abraham, A., 2011. Fuzzy classification by evolutionary algorithms, in: 2011 IEEE International Conference on Systems, Man, and Cybernetics, IEEE. pp. 313–318. doi:10.1109/ICSMC.2011.6083684.

[51] Lee, W., Stolfo, S.J., Mok, K.W., 1999. A data mining framework for building intrusion detection models, in: Proceedings of the 1999 IEEE Symposium on Security and Privacy (Cat. No. 99CB36344), IEEE. pp. 120–132. doi:10.1109/SECPRI.1999.766909.

[52] Leung, H., Haykin, S., 1991. The complex backpropagation algorithm. IEEE Transactions on signal processing 39, 2101–2104. doi:10.1109/78.134446.

[53] Li, H., Sun, J., 2013. Predicting business failure using an rsf-based case-based reasoning ensemble forecasting method. Journal of Forecasting 32, 180–192. doi:10.1002/for.1265.

[54] Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J., Liu, H., 2018. Feature selection: A data perspective. ACM Computing Surveys (CSUR) 50, 94. doi:10.1145/3136625.

[55] Liao, Y., Vemuri, V.R., 2002. Use of k-nearest neighbor classifier for intrusion detection. Computers & security 21, 439–448. doi:10.1016/S0167-4048(02)00514-X.

[56] Liu, H., Yu, L., 2005. Toward integrating feature selection algorithms for classification and clustering. IEEE Transactions on Knowledge & Data Engineering , 491–502doi:10.1109/TKDE.2005.66.

[57] Malik, A.J., Shahzad, W., Khan, F.A., 2015. Network intrusion detection using hybrid binary pso and random forests algorithm. Security and Communication Networks 8, 2646–2660. doi:10.1002/sec.508.

[58] Mansouri, S., et al., 2019. Intrusion detection system using an ant colony gene selection method based on information gain ratio using fuzzy rough sets. AUT Journal of Modeling and Simulation doi:10.22060/MISCJ.2019.14535.5110.

[59] Maza, S., Touahria, M., 2018. Feature selection algorithms in intrusion detection system: A survey. KSII Transactions on Internet & Information Systems 12. doi:10.3837/tiis.2018.10.024.

[60] Mi, J., Wang, K., Li, P., Guo, S., Sun, Y., 2018. Software-defined green 5g system for big data. IEEE Communications Magazine 56, 116–123. doi:10.1109/MCOM.2017.1700048.

[61] Mishra, P., Varadharajan, V., Tupakula, U., Pilli, E.S., 2018. A detailed investigation and analysis of using machine learning techniques for intrusion detection. IEEE Communications Surveys & Tutorials doi:10.1109/COMST.2018.2847722.

[62] Moustafa, N., Turnbull, B., Choo, K.K.R., 2018. An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. IEEE Internet of Things Journal doi:10.1109/JIOT.2018.2871719.

[63] Nemenyi, P., 1962. Distribution-free multiple comparisons. Biometrics 18, 263.

[64] Pajouh, H.H., Javidan, R., Khayami, R., Ali, D., Choo, K.K.R., 2016. A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in iot backbone networks. IEEE Transactions on Emerging Topics in Computing doi:10.1109/TETC.2016.2633228.

[65] Pal, S.K., Wang, P.P., 2017. Genetic algorithms for pattern recognition. CRC press. doi:10.1201/9780203713402.

[66] Panigrah, A., Patra, M.R., 2016. Fuzzy rough classification models for network intrusion detection. Transactions on Machine Learning and Artificial Intelligence 4, 07. doi:10.14738/tmlai.42.1882.

[67] Panigrahi, A., Patra, M., 2019. Anomaly based network intrusion detection using bayes net classifiers. International Journal of Scientific and Technology Research 8, 481–485.

[68] Papamartzivanos, D., Mármol, F.G., Kambourakis, G., 2018. Dendron: Genetic trees driven rule induction for network intrusion detection systems. Future Generation Computer Systems 79, 558–574. doi:10.1016/j.future.2017.09.056.

[69] Parker, L.R., Yoo, P.D., Asyhari, T.A., Chermak, L., Jhi, Y., Taha, K., 2019. Demise: interpretable deep extraction and mutual information selection techniques for iot intrusion detection, in: Proceedings of the 14th International Conference on Availability, Reliability and Security, pp. 1–10. doi:10.1145/3339252.3340497.

[70] Paulauskas, N., Auskalnis, J., 2017. Analysis of data pre-processing influence on intrusion detection using nsl-kdd dataset, in: 2017 Open Conference of Electrical, Electronic and Information Sciences (eStream), IEEE. pp. 1–5. doi:10.1109/eStream.2017.7950325.

[71] Pervez, M.S., Farid, D.M., 2014. Feature selection and intrusion classification in nsl-kdd cup 99 dataset employing svms, in: The 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014), IEEE. pp. 1–6. doi:10.1109/SKIMA.2014.7083539.

[72] Pham, N.T., Foo, E., Suriadi, S., Jeffrey, H., Lahza, H.F.M., 2018. Improving performance of intrusion detection system using ensemble methods and feature selection, in: Proceedings of the Australasian Computer Science Week Multiconference, ACM. p. 2. doi:10.1145/3167918.3167951.

[73] Quinlan, J.R., 2014. C4. 5: programs for machine learning. Elsevier.

[74] Ran, J., Ji, Y., Tang, B., 2019. A semi-supervised learning approach to ieee 802.11 network anomaly detection, in: 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), IEEE. pp. 1–5. doi:10.1109/VTCSpring.2019.8746576.

[75] Rosset, S., Inger, A., 2000. Kdd-cup 99: knowledge discovery in a charitable organization's donor database. SIGKDD Explorations 1, 85–90. doi:10.1145/846183.846204.

[76] Sainis, N., Srivastava, D., Singh, R., 2018. Feature classification and outlier detection to increased accuracy in intrusion detection system. International Journal of Applied Engineering Research 13, 7249–7255.

[77] Salo, F., Nassif, A.B., Essex, A., 2019. Dimensionality reduction with ig-pca and ensemble classifier for network intrusion detection. Computer Networks 148, 164–175. doi:10.1016/j.comnet.2018.11.010.

[78] Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A., 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization., in: ICISSP, pp. 108–116. doi:10.5220/0006639801080116.

[79] Shi, Z., Li, J., Wu, C., Li, J., 2019. Deepwindow: An efficient method for online network traffic anomaly detection, in: 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), IEEE. pp. 2403–2408. doi:10.1109/HPCC/

SmartCity/DSS.2019.00335.
[80] Singh, S., Singh, A.K., 2018a. Detection of spam using particle swarm optimisation in feature selection. Pertanika Journal of Science & Technology 26.
[81] Singh, S., Singh, A.K., 2018b. Web-spam features selection using cfs-pso. Procedia Computer Science 125, 568–575. doi:10.1016/j.procs.2017.12.073.
[82] Tama, B.A., Comuzzi, M., Rhee, K.H., 2019. Tse-ids: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system. IEEE Access 7, 94497–94507. doi:10.1109/ACCESS.2019.2928048.
[83] Tama, B.A., Rhee, K.H., 2017. An extensive empirical evaluation of classifier ensembles for intrusion detection task. Computer Systems Science and Engineering 32, 149–158.
[84] Tama, B.A., Rhee, K.H., 2019. An in-depth experimental study of anomaly detection using gradient boosted machine. Neural Computing and Applications 31, 955–965. doi:10.1007/s00521-017-3128-z.
[85] Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A., 2009. A detailed analysis of the kdd cup 99 data set, in: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, IEEE. pp. 1–6. doi:10.1109/CISDA.2009.5356528.
[86] Thanthrige, U.S.K.P.M., Samarabandu, J., Wang, X., 2016. Machine learning techniques for intrusion detection on public dataset, in: 2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), IEEE. pp. 1–4. doi:10.1109/CCECE.2016.7726677.
[87] Vaca, F.D., Niyaz, Q., 2018. An ensemble learning based wi-fi network intrusion detection system (wnids), in: 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA), IEEE. pp. 1–5. doi:10.1109/NCA.2018.8548315.
[88] Verma, A., Ranga, V., 2019. Machine learning based intrusion detection systems for iot applications. Wireless Personal Communications , 1–24.doi:10.1007/s11277-019-06986-8.
[89] Wang, K., Du, M., Maharjan, S., Sun, Y., 2017. Strategic honeypot game model for distributed denial of service attacks in the smart grid. IEEE Transactions on Smart Grid 8, 2474–2482. doi:10.1109/TSG.2017.2670144.
[90] Wang, K., Du, M., Sun, Y., Vinel, A., Zhang, Y., 2016a. Attack detection and distributed forensics in machine-to-machine networks. IEEE Network 30, 49–55. doi:10.1109/MNET.2016.1600113NM.
[91] Wang, K., Du, M., Yang, D., Zhu, C., Shen, J., Zhang, Y., 2016b. Game-theory-based active defense for intrusion detection in cyber-physical embedded systems. ACM Transactions on Embedded Computing Systems (TECS) 16, 18. doi:10.1145/2886100.
[92] Witten, I.H., Frank, E., Hall, M.A., Pal, C.J., 2016. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.
[93] Yang, B., Lu, Y., Zhu, K., Yang, G., Liu, J., Yin, H., 2017. Feature selection based on modified bat algorithm. IEICE TRANSACTIONS on Information and Systems 100, 1860–1869. doi:10.1587/transinf.2016EDP7471.
[94] Yang, X.S., 2010. A new metaheuristic bat-inspired algorithm, in: Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, pp. 65–74. doi:10.1007/978-3-642-12538-6_6.
[95] Yang, X.S., 2014. Nature-inspired optimization algorithms. Elsevier.
[96] Yang, X.S., He, X., 2013. Bat algorithm: literature review and applications. International Journal of Bio-Inspired Computation 5, 141–149. doi:10.1504/IJBIC.2013.055093.
[97] Zhang, F., Wang, D., 2013. An effective feature selection approach for network intrusion detection, in: 2013 IEEE Eighth International Conference on Networking, Architecture and Storage, IEEE. pp. 307–311. doi:10.1109/NAS.2013.49.
[98] Zhang, Y., Wang, S., Ji, G., 2015. A comprehensive survey on particle swarm optimization algorithm and its applications. Mathematical Problems in Engineering 2015. doi:10.1155/2015/931256.
[99] Zhong, Y., Chen, W., Wang, Z., Chen, Y., Wang, K., Li, Y., Yin, X., Shi, X., Yang, J., Li, K., 2020. Helad: A novel network anomaly detection model based on heterogeneous ensemble learning. Computer Networks 169, 107049. doi:10.1016/j.comnet.2019.107049.

Yuyang Zhou is currently pursuing the Ph.D. degree with the Cyber Science and Engineering School, Southeast University. His research interests include cyber security, traffic classification, and moving target defense.

Guang Cheng received the BS degree in Traffic Engineering from Southeast University in 1994, the MS degree in Computer Application from Heifei University of Technology in 2000, and the Ph.D degree in Computer Network from Southeast University in 2003. He is a full professor in the School of Cyber Science and Engineering, Southeast University, Nanjing, China. He is a senior member of the IEEE. His research interests include network security, network measurement and traffic behavior analysis.

Shanqing Jiang is currently pursuing the Ph.D. degree with the Cyber Science and Engineering School, Southeast University. His research interests include cyber security, traffic classification, and active defense.

Mian Dai is currently pursuing the Ph.D. degree with the Cyber Science and Engineering School, Southeast University. His research interests include traffic classification, network measurement and software defined network.