IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 30, 2020, accepted December 9, 2020, date of publication December 22, 2020,
date of current version January 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3046246

An Approach for Optimizing Ensemble Intrusion Detection Systems

DERIS STIAWAN[1], AHMAD HERYANTO[1], ALI BARDADI[1], DIAN PALUPI RINI[1], IMAM MUCH IBNU SUBROTO[2], (Member, IEEE), KURNIABUDI[3,4], MOHD YAZID BIN IDRIS[5], (Member, IEEE), ABDUL HANAN ABDULLAH[5], (Member, IEEE), BEDINE KERIM[6], AND RAHMAT BUDIARTO[6]

[1]Faculty of Computer Science, Universitas Sriwijaya, Palembang 30662, Indonesia
[2]Department of Electrical Engineering, Universitas Islam Sultan Agung, Semarang 50112, Indonesia
[3]Faculty of Engineering, Universitas Sriwijaya, Palembang 30662, Indonesia
[4]Faculty of Computer Science, Universitas Dinamika Bangsa, Jambi 36138, Indonesia
[5]Faculty of Engineering, School of Computing, Universiti Teknologi Malaysia, Johor Bahru 81310, Malaysia
[6]College of Computer Science and IT, Albaha University, Al Bahah 65527, Saudi Arabia

Corresponding author: Deris Stiawan (deris@unsri.ac.id)

This work was supported in part by the PRN Research Scheme under Grant 160/E1/PRN/2020, in part by the INSINAS Research Scheme under Grant 104/INS-1/PPK/E4/2020, and in part by the Ministry of Education and Culture of the Republic of Indonesia through the research grant from the Institute of Research and Community Service, Universitas Sriwijaya, under Grant 0685/UN9/SK.BUK/KO/2020.

ABSTRACT Intrusion Detection System is yet an interesting research topic. With a very large amount of traffic in real-time networks, feature selection techniques that are effectively able to find important and relevant features are required. Hence, the most important and relevant set of features is the key to improve the performance of intrusion detection system. This study aims to find the best relevant selected features that can be used as important features in a new IDS dataset. To achieve the aim, an approach for generating optimized ensemble IDS is developed. Six features selection methods are used and compared, i.e.: Information Gain (IG), Gain Ratio (GR), Symmetrical Uncertainty (SU), Relief-F (R-F), One-R (OR) and Chi-Square (CS). The feature selection techniques produce sets of selected features. Each best selected number of features that are obtained from feature ranking step for respective feature selection technique will be used to classify attacks via four classification methods, i.e.: Bayesian Network (BN), Naïve Bayesian (NB), Decision Tree: J48 and SOM. Then, each feature selection technique with its respective best features is combined with each classifier method to generate ensemble IDSs. Lastly, the ensemble IDSs are evaluated using Hold-up, K-fold validation approaches, as well as F-Measure and statistical validation approaches. Experimental results using Weka tools on ITD-UTM dataset show the optimized ensemble IDSs using (SU and BN); using (CS and BN) or (CS and SOM) or (IG and NB); and using (OR and BN) with respective ten, four and seven best selected features achieves 81.0316%, 85.2593%, and 80.8625% of accuracy, respectively. In addition, ensemble IDSs using (SU and BN) and using (OR and J48) with ten and six best respective selected features, perform the best F-measure value, i.e.: 0.853 and 0.830, respectively. Indirect comparison with other ensemble IDS on different dataset is discussed.

INDEX TERMS Intrusion detection system (IDS), feature selection, feature classifier, hold-out, K-fold, F-measure.

I. INTRODUCTION

In real-time network traffic, a large amount of incoming packet data need to be identified in order to resolve security issue, i.e.: identifying whether the traffic is normal or attack. Therefore the Intrusion Detection System (IDS) yet an interesting research topic [1]. By rapid growing of the traffic size in networks and computers, accurately analyzing threats in real-time network traffic and then extracting the information from the basic features of packets become the main challenge in IDS. The process of extracting information from a collection of data known as feature extraction [2], can

The associate editor coordinating the review of this manuscript and approving it for publication was Firooz B. Saghezchi.

decrease the size of feature space, without losing the information in the feature space [3]. Feature extraction is a significant process in data mining and machine learning to reduce the dimensionality of the selected feature [4]. Many IDS research works have implemented feature extraction process, e.g.: works in [5] and [6] use Principle Component Analysis (PCA) to reduce the dimensionality of a large dataset, executed in the preprocessing phase.

On the other hand, the new IDS dataset has several attributes need to be selected which act as significant features for detecting potential attacks [7]. Various techniques of ranking search and feature selection have been suggested for the machine learning practice such as Information Gain (IG) [8], Gain Ratio (GR) [9], Symmetrical Uncertainty (SU) [10], R-F [11], and Chi-Square [12].

Ranking search is a method that gives a score to a set of entities, each of them is then computed and sorted based on the assigned scores. The scores represent the degree of relevance depending on the applications [13]. Each feature selection technique produces different features scores and ranks, according to its computation strategy and search method. It is a significant task to investigate relevant features that can be used to effectively identify whether the traffic is normal or attack. Therefore, it is necessary to do an analysis and validation of ranking-based feature selection techniques and their ability to produce important features that have an impact on improving the performance of intrusion detection systems.

In this study, the IG, GR, SU, R-F, One-R and CS feature selections are used to determine significant features on Intrusion Threat Detection, Universiti Teknologi Malaysia (ITD-UTM) dataset [14]. Candidate features will be used by the ensemble of the feature selection techniques with four classification algorithms/methods, i.e.: Bayesian Network, J48, Naïve Bayesian (NB), and Self-Organizing Map Neural Network (SOM). The results for each classification then are compared and analyzed based on the classification accuracy, means, and standard deviation. Finally the selected features will undergo data validation to verify the significance of the chosen features.

The rest of the paper is structured as follows. Section 2 presents relevant researches. Section 3 discusses the research methodology. Section 4 explains the experimental set up, results and discussion. Finally, Section 5 draws conclusion and potential future works.

II. RELEVANT WORK

IDS research is very interesting and challenging, this is proven by IDS survey research in [15], [16], and [17]. Many techniques and models have been studied and developed to improve the performance of ensemble IDS to detect various forms of attack as summarized in Table 1.

Existing IDSs propose different methods to improve detection performance. Researchers mainly use DR, false alarm, accuracy, F-measure as performance measures. Some researchers consider computation time as well. Despite, the methods proposed in previous studies achieve detection performance improvement, yet having several weaknesses include: detection accuracy for each type of attacks, selection of the most relevant features and computation time. Therefore the main objective of this study is to determine the most relevant features, so as to reduce computational complexity without sacrificing detection performance.

TABLE 1. The summary of relevant work in IDS.

Ref.#	IDS Method	Pros	Cons
[18]	KPCA, SVM+ GA	Has higher prediction accuracy, faster convergence speed, and better generalization.	GA is computationally expensive.
[19]	Cluster Center and *k-NN.*	Higher accuracy, detection rate (DR), lower false alarm rate (FAR).	Require extra computation resources to extract the distance based features.
[20]	Anomaly detection+ misuse detection.	Effectively detects an anomaly with a low FAR. k-NN algorithm is used to improve DR.	Weak in detecting R2L attacks.
[21]	C4.5 algorithm + signature-based IDS.	The proposed approach achieves good efficiency in finding attack with lesser time consumed.	Using IG for feature selection, human intervention is required
[12]	Chi-square FS+ multi class SVM.	High DR and low FPR vs. traditional Approaches.	U2R detection capabilities still need to be improved.
[22]	IDS + ML algorithm,Random Forest, MLP, & LibSVM	Superior response vs Random Forest, in term of response time, DR, and false-positive rate.	From 41 features, 17 features were selected, it can still be reduced.
[23]	K-Nearest Neighbor classifier algorithm	Better accuracy, precision, recall and F-measure values.	Analyze 78 features, not considering computation time.
[24]	Chi-Square+ SVM+ Modified NB + LPBoost	Improve accuracy and has better generalization when integrating multiple classifiers.	Has a high accuracy, MNB suffering in detecting DoS and U2R.

*KPCA:kernel principal component analysis; SVM:support vector machine; GA:genetic algorithm; kNN: k-nearest neighbors; ML: machine learning; MLP: multi-layer perceptron; LibSVM: Library SVM.

In this research, the ensemble method is used to improve the performance of IDS [25]. In this case, we implement an ensemble of feature selection techniques as inspired by work in [26] that ensembles filter and wrapper feature selection techniques and succeeded in improving IDS performance. Ensemble approach is also proposed by Zhou *et al.* [27] to select optimal feature that improved the IDS performance.

Research works in [28]–[30] have reported that the preprocessing is a critical step in IDS. This preprocessing impacts the performance of detection algorithms. There are three main steps in preprocessing, namely: data creation, features construction and features reduction [28]. Research work in [10], applies a preprocessing step to replace missing values and discretization. The preprocessing steps in [29] involve three stages, namely: data transferring, data normalization, and

TABLE 2. The summary of relevant work in feature selection in IDS.

Ref.#	FS Method	Pros	Cons
[32]	Information Gain	- Reduce the 41 NSL-KDD features to 20 features; - Improve the ensemble detection.	Using threshold value to select features, the threshold value defined by human.
[33]	Chi-square	- Reduce 41 NSL-KDD to 22 optimal feature; - Improve Naïve Bayes Classifier performance.	Lack in detection minority attacks in NSL-KDD dataset.
[34]	Information Gain	- Calculate feature metric; - Efficient and accurate detection different attack.	The minimum metric value defined by human (IG >= 0.001).
[35]	Information Gain	- Reduce the 41 NSL-KDD features to 8 features; - Improves the accuracy rate and reduces the detection time.	The minimum metric value defined by human (IG > 0.40).
[36]	Correlation-Based	- Reduce 41 NSL-KDD to 17 optimal feature; - Improve the ANN in detection.	Need more resources when implemented in large scale network.
[37]	CfsSubset Evaluation + Best First Search Algorithm	- 13 attributes are selected from the 42; - Reduce calculations and processing time; - Improve the detection accuracy.	Human intervention is needed to control the level of backtracking.
[68]	SU+ACO	- Addresses the issue of threshold value using subset generation capability in the filter methods.	Accuracy is not optimum yet.

feature selection. Meanwhile, researchers in [30] implement preprocessing step to remove irrelevant features in the dataset or known as feature extraction. The preprocessing steps are needed to prepare the data before it is analyzed using a machine learning algorithm. Therefore, the preprocessing step is an important step in Machine Learning algorithm and may impact the detection rate [31].

Various studies have been conducted on the selection of the most dominant features, with various feature selection techniques as summarized in Table 2.

Previous studies have shown that the feature selection technique is able to produce the most relevant features and can improve attack detection performance. Various techniques have been proposed with various performance. IG is one of the most widely used techniques. The IG feature selection technique generates a ranking of features based on the weight value. However, there is no standard reference for minimum weight to determine the number of relevant features. For example Researchers in [34] and [35] use a score feature above 0.001 and a score feature above 0.4, respectively. Meanwhile, research work in [7] considers the minimum weight score of 0.8.

TABLE 3. The comparison of datasets and real network.

ID Dataset	File ID	Pentest Duration	Total Packets	Attack Variety
DARPA	1999_thursday_inside	21:59:50	1,563,069	75
ISCX	Testbed_12june	23:59:59	5,973,980	63
CICIDS-'17	July 5, 2017 DoS/DDoS	03:15:58	2,830,743	744
ITD UTM	Attack_test2_Redhat	02:05:55	2,331,773	715
Real Live	4 hours in 1e Pentest	04:12:29	3,771,024	750

The research works in [34] and [35] also have inspired the authors of this article to produce an ensemble IDS that is able to identify attacks on real networks, along with reliable features selection method that produces the most important and relevant features, which in turn, increase the performance of the detection accuracy as well as shorten the detection time.

III. RESEARCH METHOD

This section describes the creation of the dataset and the proposed approach that consists of the preprocessing, feature selection process, and validation of the generated ensemble IDS

A. DATASET SPESIFICATION

This study uses Intrusion Threat Detection (ITD-UTM) dataset, developed at Universiti Teknologi Malaysia in 2012, as introduced and described in [14]. We conducted experiments on penetration test on three (3) publicly available datasets, i.e.: DARPA, ISCX, CICIDS-2017, ITD-UTM, and on production live network in COMNETs Lab, Universitas Sriwijaya following the procedures described in [38]. Table 3 shows the results.

Results in Table 3 show that ITD-UTM and CICIDS-2017 datasets [39] have attack varieties close to the real traffic, means that the dataset represent well the real traffic, in term of complexity of the attacks. Furthermore, reasons of choosing ITD-UTM dataset are as follows.

- The need of a dataset that represents closely real network with high complexity in term of attacks types.
- As an alternative reliable benchmark dataset for researchers for their future works in IDS.
- No research works on IDS using the ITD-UTM dataset for the benchmark.

Fig. 1 shows the high-level view of the features extraction process from the ITD-UTM dataset (FreeBSD attack scenario). The first module, i.e.: Data Reader collects data by either sniffing it directly from the network connection by using the "tcpdump -w" command, or extracting already

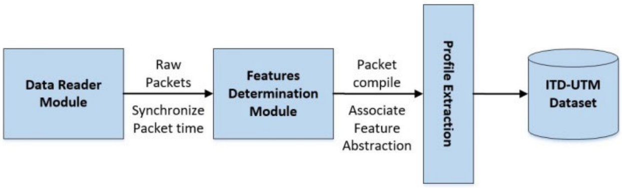

FIGURE 1. Traffic profile database creation process.

saved tcpdump file using the "tcpdump -r" function. The second module contains a feature determination tool for recording time connection reconstruction, reducing the number of features from overall attributes captured during the recording and it is in charge of constructing and storing the uncounted connections. Having done feature determination, features are extracted from traffic packets' attributes to create traffic profiles (attacks and normal traffics). The last module is a traffic profile database as the dataset that stores the content of attacks and normal access simulations. The labels provided from the connection log are based on features identified and time reconstruction during the attack scenarios in penetration testing.

B. THE PROPOSED APPROACH

Fig. 2 shows the overall architecture of the proposed approach carried out in this study. The proposed approach ensembles features ranking and features selection with aforementioned classification algorithms/methods and is divided into three main phases as follows.

1) PREPROCESSING THE DATA

The initial step before the feature selection step is to preprocess the collected data. In this step, the dataset is processed and cleaned prior to classifications for better generalization of error and accuracy. Normally, real traffic data contains noise and it tends to be incomplete and inconsistent. Thus, this preprocessing step involves finding the missing value of each instance, data normalization and data division. The output of the preprocessing step is a new IDS dataset that contains of 22 attributes (input features), 1 attribute (i.e.: class variable including: Normal, U2R, R2L, Probes, and DoS) and 11,878 instances. The data set characteristics are multivariate; the feature characteristics are categorical and real number.

The first preprocessing that needs to be carried out with the IDS dataset is finding the missing value of each instance. We found out that there are 576 instances, which have missing values and must be removed from the dataset. Having done removing the data that has missing value, the total numbers of instances that can be processed is made up to 11,302 instances. The second preprocessing is data normalization, Data normalization is typically used in machine learning processes, hence the values of all variables in the dataset are normalized into value range of [0, 1]. However, in the new IDS dataset, normalization is not performed, because the value range varies too large among the attributes/variables. Normalization results in long decimal values may cause overflow problem.

The last preprocessing is data division. The aim of this preprocessing is to make the dataset represent the real problem as close as possible, i.e. representing attacks traffic as well as representing normal traffic. In this case, the dataset is partitioned into two sets: a training dataset and a testing dataset. The training dataset is a set of data used in machine learning to discover a potentially predictive model, while the testing dataset is a set of data used in machine learning to assess the strength and utility of a predictive model. The data in the testing dataset contains the knowledge values for prediction discovered during the training process, so it is unable to determine whether the model presume is correct.

2) EXAMINING FEATURE RANKING FOR FEATURE SELECTION AND CLASSIFIER METHOD

Feature ranking, also called as feature weighting, assesses individual features and assigns them weights according to their degrees of relevance [40], [41], while feature selection (FS) has been evaluated by [42] and [43]. In Feature Ranking algorithms category, a subset of features is often selected from the top of a ranking list. This approach is efficient for high–dimensional data due to its linear time complexity in terms of dimensionality [41].

Feature ranking and feature selection techniques have been proposed in machine learning literatures [40]–[42]. The purpose of these methods is to discard irrelevant or redundant features from a given feature vector. In this research work, feature ranking and selection methods are used with two basic steps of general architecture subset generation and subset evaluation for the ranking of each feature using IG, GR, SU and R-F, that are entropy-based feature selection, and using OR and CS that are statistical-based feature selection.

The limitation of these feature selection techniques is they do not consider dependencies between the candidate feature and unselected features.

The following are brief descriptions on the six feature selection techniques used in this article.

a: INFORMATION GAIN (IG)

This feature selection technique determines the best features through computing feature entropy. Entropy is an uncertainty degree used for inferring features distribution in concise form [44], and then the features are selected based on a simple rank. It is categorized as filter-based feature selection mechanism [40], [7]. It calculates the entropy using (1).

$$Entropy\,(S) = \sum\nolimits_{i}^{c} -P_i log_2 P_i \tag{1}$$

where c is the number of values in the class classification and P_i is the number of samples for class i. After getting the entropy value, the information gain value is calculated by (2).

$$Gain\,(S, A) = Entropy\,(s) - \sum\nolimits_{Values(A)} \frac{|sv|}{|s|} Entropy(S_v) \tag{2}$$

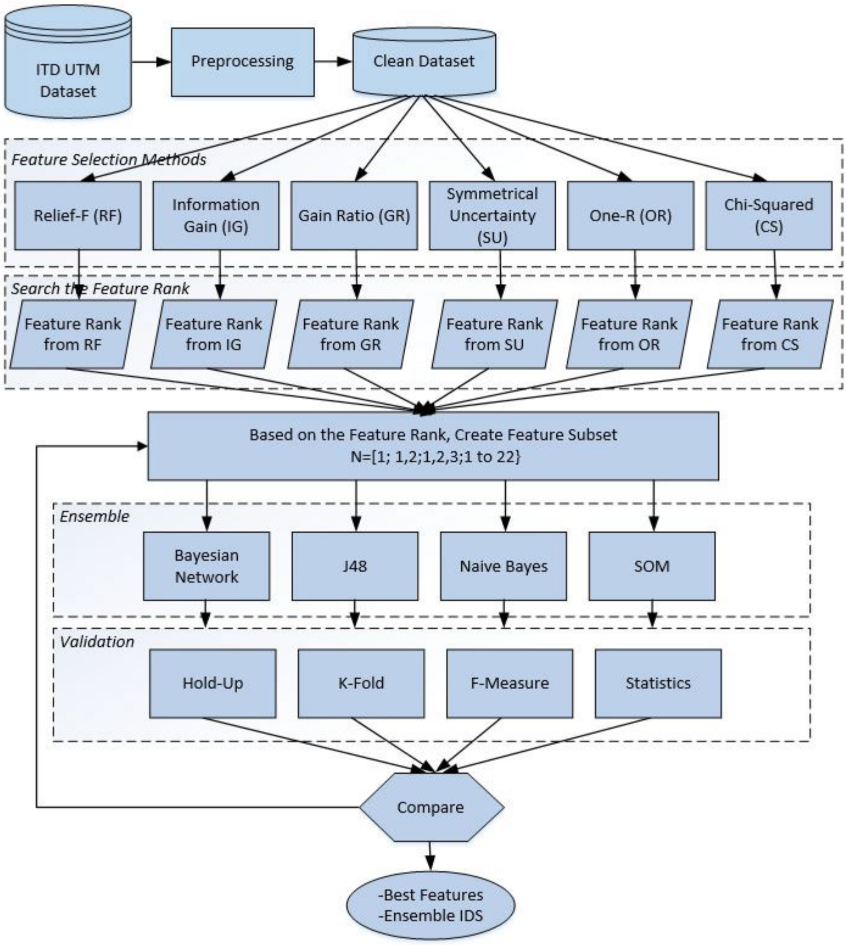

FIGURE 2. Proposed approach.

where A is an attribute, v is a possible value for attribute A, $Values(A)$ are a set of possible values for A. $|S_v|$ is the number of samples for value v. $|S|$ is the number of samples for all data samples and Entropy (S_v) is entropy for samples that have a value of v.

b: GAIN RATIO (GR)

It is a variation of the information gain by taking into account number and size of branches in selecting an attribute/feature to decrease its bias on high-branch attributes. It is able to correct unstable data by considering the intrinsic information of a split. Equation (3) is used to split the information.

$$Split\ Information = -\sum\nolimits_{t=1}^{c} \frac{Si}{S} \log 2 \frac{Si}{S} \tag{3}$$

where Si to Sc are c subsets resulting from solving S using attribute A which has c number of values. Then, the gain ratio is determined by (4).

$$Gain\ Ratio = -\frac{Gain\,(S, A)}{Split\ Information\,(S, A)} \tag{4}$$

Intrusion detection studies that use gain ratio mechanism include [45] and [46].

c: SYMMETRICAL UNCERTAINTY (SU)

This feature selection technique works by measuring the uncertainty of a random variable x to another variable y as given by (5) and (6). $P(xi)$ is the prior probability for all values of x and $P(xi|yi)$ is the posterior probability of x given y [52], [53]. It is claimed to be effective in feature selection for large scale data sets.

$$H\,(x) = -\sum P\,(x_i)\, log_2(P\,(x_i)) \tag{5}$$

$$H(x|y) = -\sum\nolimits_i P(y_i) \sum\nolimits_j P(x_i|y_i) log_2(P\,(x_i\,|\,y_i)) \tag{6}$$

Symmetric Uncertainty (SU) is a correlation measure between the features and the class, and it is calculated by (7).

$$SU = \frac{H\,(x) + H\,(y) - H(x|y)}{H\,(x) + H(y)} \tag{7}$$

where $H(x)$ and $H(y)$ are the entropies based on the probability associated with each feature and class value respectively and $H(x, y)$, the joint probabilities of all combinations of values of x and y [40].

SU has symmetric nature where for two independent variables x and y, $SU(x,y) = SU\ (x,y)$. Thus, the number of required comparison processes decreased. SU is not biased by the number of values of an attribute.

d: RELIEF-F (RF)

This feature selection algorithm evaluates individually features' quality subset by comparing nearby features with the selected features [54]. The algorithm starts with drawing the instances at random, compute their nearest neighbors, and then adjust a feature weighting vector to give more weight to features that discriminate the instance from neighbors of different classes [55]. Thus, it deals with multi-class problems and select k neighboring instances per class for evaluation [56].The Relief-F measure for a given feature 'a' is represented as W_a in (8).

$$W_a = \frac{1}{k}\left[diff_{\,a}(R,H) - \left(\sum\nolimits_{j=1}^{k} \frac{p(M_j)}{1-p(R)} * diff_{\,a}(R,M)\right)\right] \tag{8}$$

where, $diff_{\,a}(RX)$ is difference in value of 'a' for instances R and X, *and* $p(X)$ is probability of X.

e: ONE-R (OR)

This feature selection is a rule-based algorithm that chooses the lowest error rate attributes as its one rule and then ranked them accordingly [57], [58]. It constructs rules and tests a single attribute at a time and branch for every value of that attribute [59].

f: CHI-SQUARE (CS)

This feature selection technique eliminates irrelevant features/attributes based on the value of the dependency weight between features and class. This method evaluates the feature value by calculating the square statistical value with respect to class [47], [48]. Several studies that apply chi-square for feature selection include research [49], [50] and [24]. Chi-square value is calculated using (9).

$$x^2\,(f,c) = \frac{N*(WZ-XY)^2}{(W+x)+(W+Z)+(W+X)+(Y+Z)} \tag{9}$$

where, W = How many times the feature t and the class label c appears, X = How many times is t without a c,

Z = How many times other than c or t is there, and N = Total number of records.

The six feature selection techniques use forward floating search methods: (SFFS). The algorithm starts with a null feature set and, for each step, the best feature that satisfies some criterion function is included with the current feature set, i. e., one step of the sequential forward selection (SFS) is performed. The algorithm also verifies the possibility of improvement of the criterion if some feature is excluded. In this case, the worst feature (concerning the criterion) is eliminated from the set, that is, it is performed one step of sequential backward selection (SBS). Therefore, the SFFS proceeds dynamically increasing and decreasing the number of features until the desired dimension d is reached. The time complexity of these feature selection techniques is $O(d)$ [69].

TABLE 4. Result of ranking process on the new IDS dataset.

No	Features of new IDS dataset	IG	GR	SU	RF	OR	CS
1	Packet_Length_size	1	9	1	6	1	1
2	IP_Header_Length	20	22	20	20	20	20
3	IP_Total Length	2	10	2	7	2	2
4	IP_Fragment_Offset	21	21	21	22	21	22
5	TTL	17	13	17	16	17	17
6	Protocol	5	18	13	10	9	6
7	ID_Protocol	12	4	7	1	12	10
8	TCP_SRC_port	10	11	9	12	10	11
9	TCP_DST_port	11	12	10	14	8	12
10	TCP_Seq_num	6	1	4	8	4	5
11	TCP_Ack_num	8	2	5	9	5	8
12	TCP_offset	13	5	8	3	7	13
13	TCP_Flags	14	6	12	15	14	14
14	TCP_Win	4	7	6	4	3	4
15	TCP_Length	16	16	16	18	16	15
16	Checksum	15	17	15	5	15	16
17	ICMP_Type	19	15	19	19	19	19
18	ICMP_Code	22	20	22	21	22	21
19	ICMP_Checksum	18	8	18	11	18	18
20	UDP_SRC_port	7	14	11	2	11	7
21	UDP_DST_port	9	19	14	13	13	9
22	UDP_Length	3	3	3	17	6	3

This work chooses feature selection techniques to reduce data dimensionality and computational complexity. Overall, computational complexity of filter-based technique is $O(m\cdot n^2)$, where m is the number of training data, and n is number the of attributes/features. It is less as compared to embedded and wrapper-based techniques [60]. The complex nature of wrapper-based techniques creates the high risk of over-fitting. Thus, using feature selection technique that produces significant, relevant, less number of features and less computational complexity will reduce the execution time of classification algorithms used in the anomaly/attack detection process.

For IG, GR, SU, and RF which are entropy-based FSs, the features are given IDs from 1 to 22. The FSs rank and group the features according to the minimum weight values. Thus, groups of features are obtained and each feature sub-group will be having different number of features as shown in Table 4. For OR and SC which are statistical-based FSs, the features are ranked and grouped based on the average and standard deviation.

3) ENSAMBLE, VALIDATION AND COMPARISON

Each best selected number of features that are obtained from feature ranking step for respective feature selection technique will be used to classify attacks via four classification methods that have been used by previous researchers in IDS classification, i.e.: Bayesian Network, Naïve Bayesian (NB), Decision Tree: J48 and SOM. The rationales of choosing these four classification methods mainly based on accuracy, scalability and processing time. Sahu and Mehtre [61] reveals that the ability of Bayesian Network in classifying attacks outperforms other algorithms. Sahu and Mehtre [62] conclude that J48 algorithm has good accuracy in classifying attacks. Naïve Bayes is a classification algorithm that is able to identify class labels faster than other algorithms because it has a low complexity of the model [63]. The SOM is particularly powerful because It never needs to be told what intrusive behavior looks like. It learns to characterize normal behavior then implicitly prepares itself to detect any aberrant network activity [64]. Thus, each feature selection technique with its respective best features is combined with each classifier method to generate ensemble IDSs.

In this study, machine learning Weka tools are used for ranking search and classification processes. Weka is a collection of machine learning algorithms for data mining tasks and has been widely used for classification and clustering of data in various application domains [65]. The results for each classification are compared and analyzed by considering the classifications accuracy represented by (10).

$$Accuracy = (TN + TP)/(TN + TP + FN + FP) \quad (10)$$

where TN is True Negative, TP is True Positive, FN is False Negative, and FP is False Negative. Thus, the accuracy is the percentage of number of correct assessments over the number of all assessments. The validation involves Hold-out and K-Fold approaches to maintain the objectivity of the testing phase. In addition, F-measure and statistical test approaches testing were also carried out in this study, to ensure that the data used for the experiments are valid data. At the final step, an ensemble IDS, consists of selected FSs along with their best features is obtained. Figure 2 illustrates the proposed approach.

IV. EXPERIMENTAL SET UP, RESULTS AND ANALYSIS

The experiment started with sorting the rank of all 22 attributers/features of packet traffic data in the new IDS dataset using the six features ranking methods i.e.: IG, GR, SU, R-F, OR, and CS. The results are listed in Table 4.

As can be seen in Table 4, *Packet_length_size* is ranked as the 1st essential feature by IG, SU, OR, and the CS method, but it is ranked as the 6th and the 9th essential feature by R-F and by CS method, respectively. Four features were found as the lowest rank in five of the feature selection techniques, i.e.: IG, SU, R-F, OR, and CS, they are: *IP_Header_Length*, *IP_Fragment_Offset* ,*ICMP_Type, andICMP_Code*, whilst GR identifies *IP_Header_Length*,*IP_Fragment_Offset*, *UDP_DST_port*, and *ICMP_Code*. The other features also ranked differently by the feature selection techniques.

A. EXPERIMENTS USING HOLD-OUT VALIDATION

The result of each ranked feature in the new IDS dataset requires to be evaluated for the best feature selection. An independent testing set of data is preferred, to avoid overfitting. A natural approach is to split the available data into two non-overlapped parts: one for training and the other for testing. The testing data is held out and not looked at during the training. Hold-out validation avoids the overlap between the training data and testing data, yielding a more accurate estimation for the generalization of the performance of the algorithms.

Nikhitha and Jabbar [23] have studied that experimental result using 70:30 data portion provides high accuracy. Furthermore, Nikhitha and Jabbar [66] have reported that the use of the 70:30 data portion of training and testing data indicates the same level of accuracy as the portions of 80:20 and 60:40. Therefore, this research follows the two research works. For each dataset, 70% of the data is used for the training set and the other 30% is used for the testing set. Various ranking results of the features in Table 4 will be used by the respective feature selection technique and ensemble with the classifier methods, and then evaluated by four validation approaches to determine the best ensemble IDSs. The graphs in Fig. 3 to Fig. 6 present the accuracy of selected features ranked by different classification method. A feature is selected based on the highest classifier values from the testing set results because the testing set describes the strength and utility of a predictive model.

Fig. 3 until Fig. 6 present the measurement of detection accuracy for each ensemble IDS with varying the number of selected features from 1 to 22, as ITD-UTM has 22 features/attributes. From this measurement, the best, the minimum and the average of the detection accuracy are obtained. Fig. 3 shows the accuracy of detection of the six feature selection techniques ensemble with Bayesian Network Classifier. In the training set, the highest value of classification accuracy among all feature selection techniques is 99.1552 %, with the number of selected features numbers are not more than 17. The average accuracy of feature classifications of the testing set are as follows: 75.7072% for IG, 78.5833 % for GR, 78.4654 % for SU, 71.204 % for RF, 79.2034 % for OR, and 78.9164 % for CS.

Considering the minimum, the maximum, and the average values of classification accuracy on the testing set and ensemble with the Bayesian Network Classifier, two sets of features are selected for feature ranking. The first one is SU with 10 selected features, i.e.: *Packet_Length_size, IP_Total_Length, ID_Protocol, TCP_SRC_port, TCP_DST_port, TCP_Seq_num, TCP_Ack _num, TCP_offset, TCP_Win,* and *UDP_Length*. The second one is CS with 4 selected features, i.e.: *Packet_Length_size, IP_Total_Length, TCP_Win*, and *UDP_Length*.

Fig. 4 shows the accuracy of detection of the six feature selection techniques ensemble with Naïve Bayesian Classifier. In the training dataset, the highest value of classification accuracy was the IG method at 99.1552 %. Other results for all methods are as follows. GR has 63.8306 % accuracy, SU has 71.9768 % accuracy, RF has 91.6003 % accuracy, OR has 88.3538 % accuracy and CS has 88.4504 % accuracy.

Considering the minimum, the maximum, and the average values of classification accuracy on the testing set, two sets are selected for feature ranking as ensemble with a Naïve Bayesian Classifier. The first one is IG with 4 selected features, i.e.: *Packet_Length_size, IP_Total_Length, TCP_Win, UDP_Length.* The second one is RF with 8 selected features, i.e.: *Packet_Length_size,*

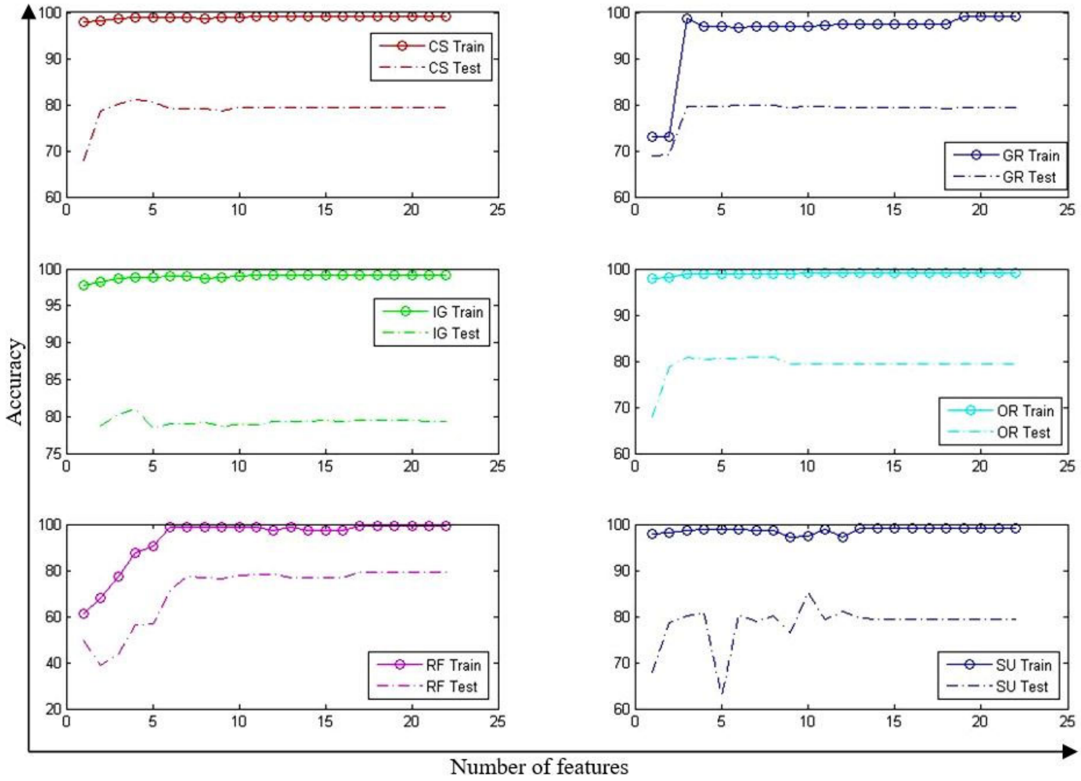

FIGURE 3. Accuracy detection ensemble with Bayesian Network (BN) classifier.

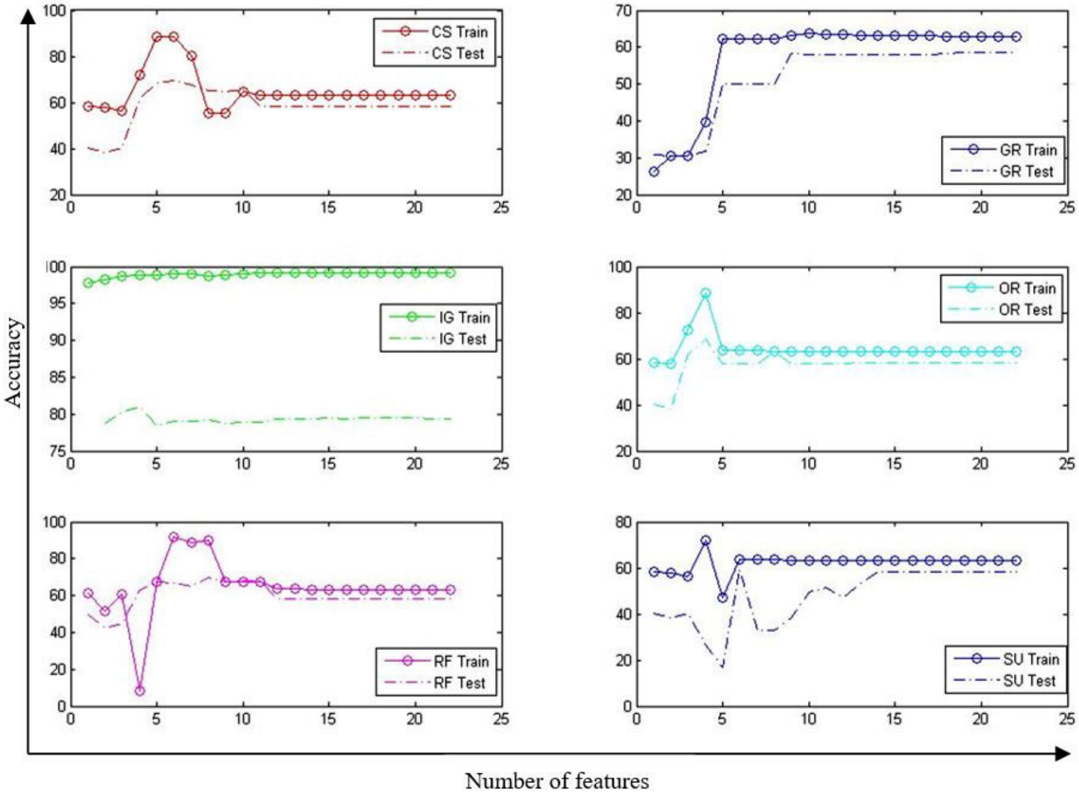

FIGURE 4. Accuracy detection ensemble with Naïve Bayesian (NB) classifier.

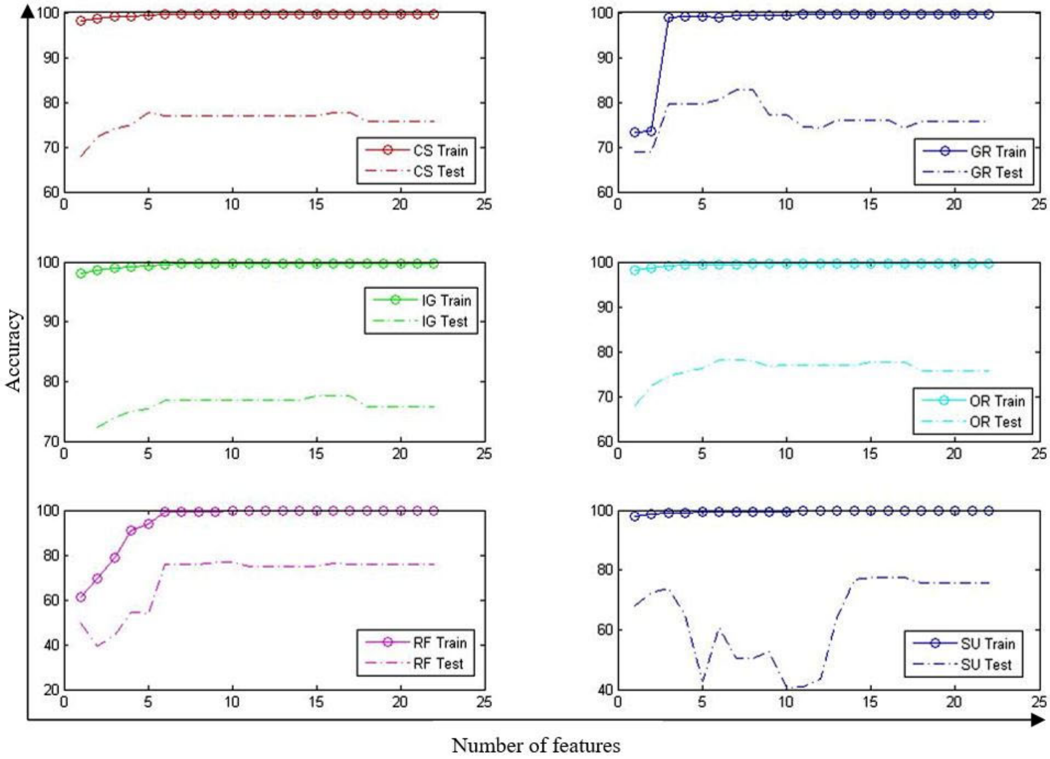

FIGURE 5. Accuracy detection ensemble with J48 classifier.

IP_Total_Length, ID_Protocol, TCP_Seq_num, TCP_offset, TCP_Win, Checksum, UDP_SRC_port.

Meanwhile, Fig. 5 shows the accuracy of detection of the six feature selection techniques ensemble with Decision Tree: J48 Classifier. In the training set, the highest value of classification accuracy achieved by RF method at 99.7707 %. SU achieves 99.7345 % accuracy while 99.7224 % accuracy is obtained by IG, OR, CS and GR.

Considering the minimum, the maximum, and the average values of classification accuracy in the testing set, two sets of the best features are selected as ensemble with J48 classifier. They are: GR with seven selected features, i.e.: *ID_Protocol, TCP_Seq_num, TCP_Ack_num, TCP_offset, TCP_Flags, TCP_Win, UDP_Length*; and One-R with six selected features, i.e.: *Packet_Length_size, IP_Total_Length, TCP_Seq_num, TCP_Ack_num, TCP_Win, UDP_Length*.

Fig. 6 shows the accuracy of detection of the six feature selection techniques ensemble with Self-Organizing Map (SOM) Artificial Neural Network classifier. In the training set, the highest value of classification accuracy is for IG method with 99.1552%. Results for other methods are as follows. OR and CS have 72.9664% accuracy, SU is 71.132% accuracy, RF is 69.1045% accuracy, and GR is 50.5069% accuracy.

Considering the minimum, the maximum, and the average values of classification accuracy on the testing set, and ensemble with SOM classifier, two sets are selected for feature ranking. They are IG with seven selected features, i.e.: *Packet_Length_size, TCP_Win, IP_Total _Length, UDP_Length*; and CS with four selected features, i.e.: *Packet_Length_size, IP_Total_Length, TCP_Win,* UDP_Length.

Fig. 7 shows the detection accuracy for the best selected feature for each feature selection technique ensemble with BN classifier. In the testing set, the highest classification accuracy is 85.2593 % which was obtained by SU with 10 selected feature numbers. The highest classification accuracy of the testing set using other methods is as follows. IG and CS each has 81.0316 % accuracy with 4 features; OR has 80.8625 % accuracy with 7 features; GR has 79.9887 % accuracy with 6 features; and RF has 79.4814 % accuracy with 17 features.

As shown in Fig. 8, for ensemble with NB, the highest classification accuracy in the testing dataset was IG that scores 81.0316 % with 4 features, followed by R-F that has 69.5321 % accuracy with 8 features, CS has 69.3067 % accuracy with 6 features, OR has 68.743 % accuracy with 4 features, SU has 59.2728 % accuracy with 6 features and GR has 58.4555 % accuracy with 6 features. The average accuracy of classification in the testing dataset of each feature is as follows. 79.3122 % for IG, 51.7449 % for GR, 47.7951 % for SU, 59.7135 % for R-F, 57.3454 % for OR, and 58.2517% for CS.

For ensemble with J48 (Fig. 9), the highest classification accuracy of the testing set was achieved by GR that

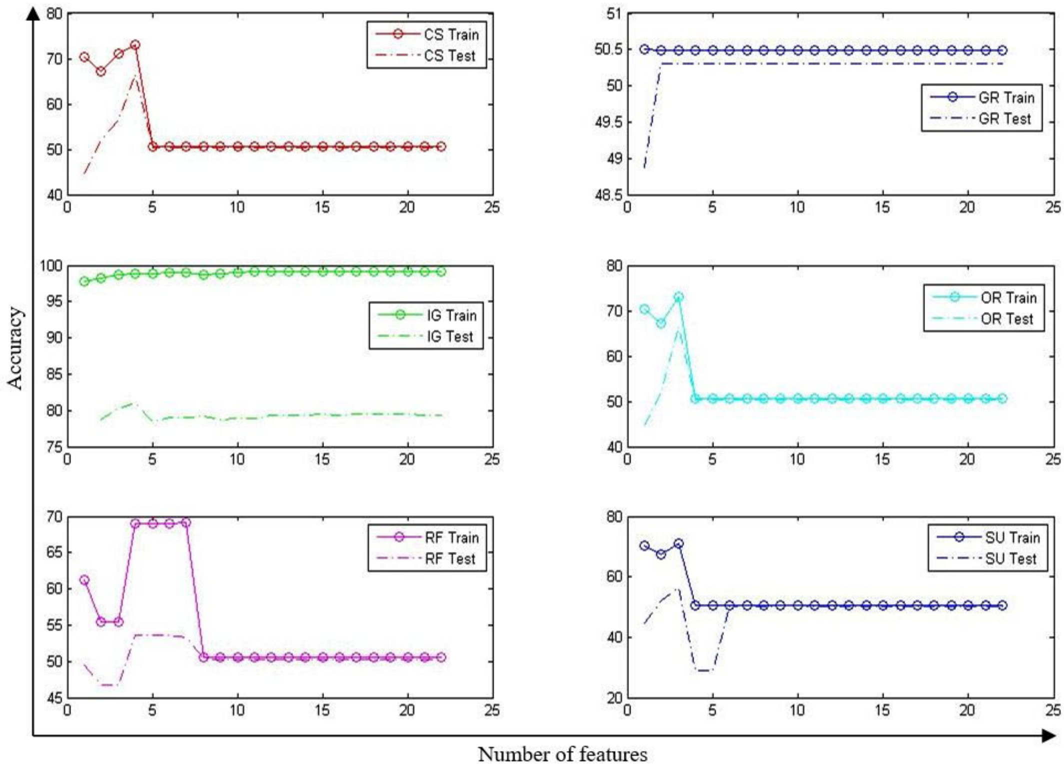

FIGURE 6. Accuracy detection ensemble with SOM.

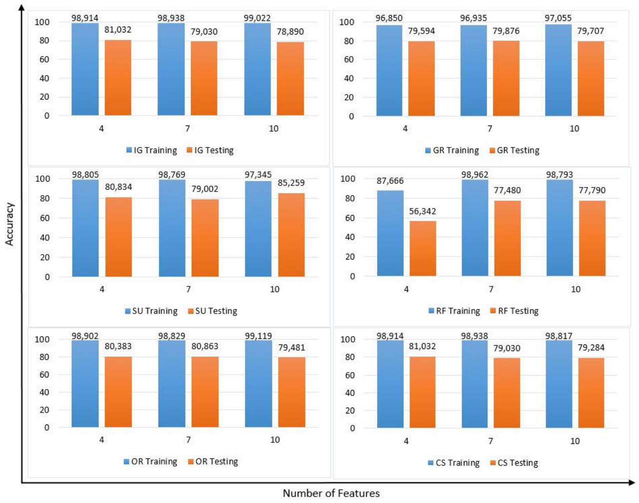

FIGURE 7. Detection accuracy ensemble with BN on the best selected features.

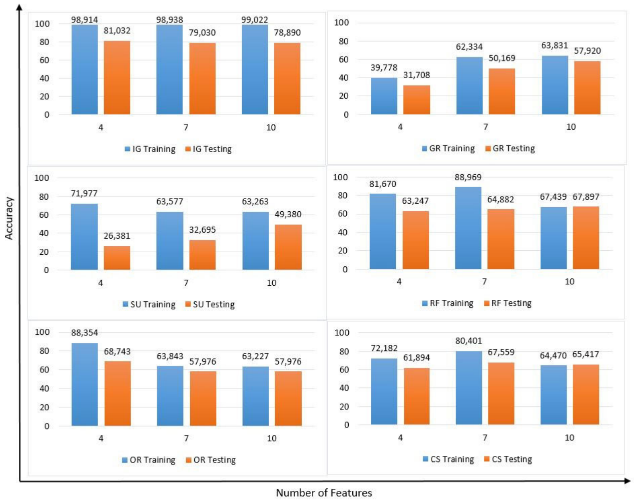

FIGURE 8. Detection accuracy ensemble with NB on the best selected features.

has 82.8918% accuracy with 7 features, followed by OR with 78.2694% accuracy with 6 features. IG, SU and CS obtain 77.6494% of classification accuracy with a number of selected features is not more than 15 features, and lastly, RF scores 76.9448% accuracy with 10 features. The averages of classification accuracy of testing set for each feature are as follows: IG: 72.7426%, GR: 76.491%, SU: 64.238%, R-F: 69.437%, OR: 76.071%, and 75.895 % for CS.

Fig. 10 shows the detection accuracy for the best selected feature for each feature selection technique ensemble with SOM classifier.

The highest classification accuracy of testing set is IG that has 81.0316% accuracy with 7 features, followed by CS that has 66.3472% accuracy with 4 features, OR that has 66.3472% accuracy with 3 features, SU that has 56.4543% accuracy with 3 features, RF that has 53.6359% accuracy with 4 features, and the lowest result is GR that obtains

TABLE 5. Summary of accuracy results for the ensemble IDS (in %).

FS	Classifier Algorithm											
	BN			NB			J48			SOM		
	Best	Feat.#	Ave.	Best	Feat.#	Ave.	Best	Feat.#	Ave.	Best	Feat.#	Ave.
SU	***85.2593***	***10***	***78.4654***	59.2728	6	47.7951	77.6494	≤15	64.238	56.4543	3	48.4857
IG	81.0316	4	75.7072	***81.0316***	***4***	***79.3122***	77.6494	≤15	72.7426	**81.0316**	**7**	**75.7071**
CS	***81.0316***	***4***	***78.9164***	69.3067	6	58.2517	77.6494	≤15	74.895	**66.3472**	**3**	**51.1696**
OR	80.8625	7	79.2034	68.743	4	57.3454	**78.2694**	**6**	**76.071**	66.3472	3	50.867
GR	79.9887	6	78.5833	58.4555	6	51.7449	***82.8918***	***7***	***76.491***	50.31	2	50.867
RF	79.4814	17	71.204	**69.5321**	**8**	**59.7135**	76.9448	10	69.437	53.6359	4	50.5381

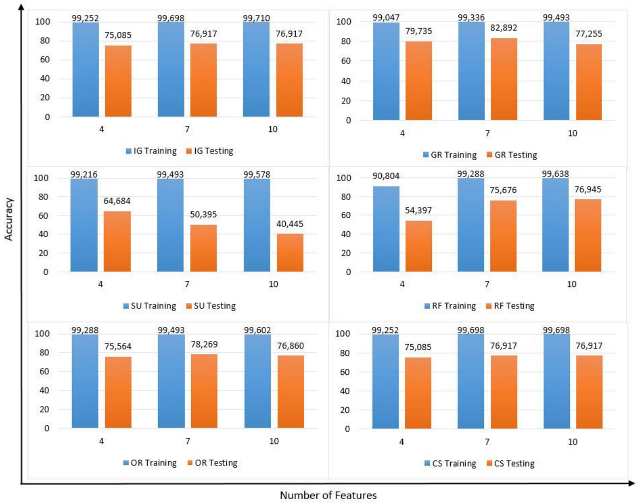

FIGURE 9. Detection accuracy ensemble with J48 on the best selected features.

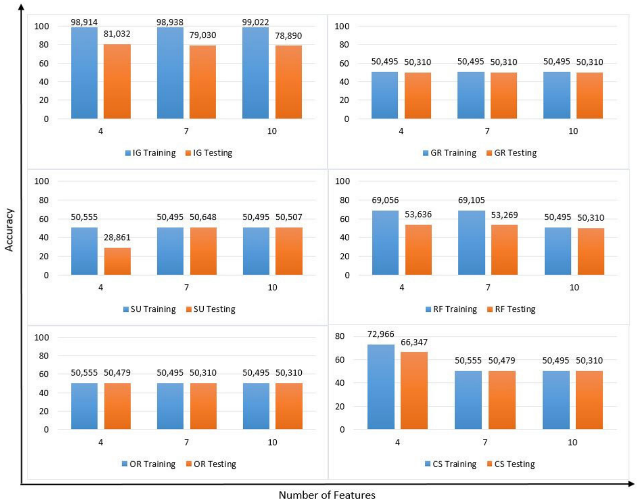

FIGURE 10. Detection accuracy ensemble with J48 on the best selected features.

just 50.31% of classification accuracy with 2 features. The averages of classification accuracy on the testing set for each feature are as follows: 75.7071% for IG, 50.2447% for GR, 48.4857% for SU, 50.5381% for R-F, 50.867% for OR, and 51.1696% for CS.

Table 5 summarizes the experimental results on detection accuracy while Table 6 shows the best ensemble IDSs along with their features.

B. EVALUATION OF THE SELECTED FEATURES USING K-FOLD VALIDATION

In Section IV.A, the ranking search methods were evaluated by hold-out validation. The features are selected based on the best classification accuracy using four machine learning methods. The results are: SU with 10 features, CS with 4 features, IG with 7 features, RF with 8 features, GR with 7 features, and OR with 6 features. CS and IG have the same number of the best selected features, i.e.: 4, so from now on only consider CS.

The hold-out method has two basic drawbacks. In a case where we have a sparse dataset we may not be able to afford the ‘‘luxury’’ of setting aside a portion of the dataset for testing. Since it is a single train-and-test experiment, the hold-out’s error rate estimation may mislead if it happens to get an ‘‘unfortunate’’ split. The limitations of the hold-out can be overcome with a family of re-sampling methods at the expense of more computations using cross validation, i.e.: random sub-sampling, K-Fold cross-validation, leave-one-out cross-validation, and bootstrap. The advantage of K-Fold cross-validation over hold-out validation is that all observations are used for both training and testing, and each observation is used for testing exactly once. In stratified K-Fold cross-validation, the folds are selected so that the mean response value is approximately equal across in all folds.

In this experiment, all selected number of features sets will be re-evaluated with the four classifier methods and verified using 5-fold cross validation. Initially, the new IDS dataset needs to be divided randomly into five sets of equal size. One subset is used as the testing dataset, and the other four subsets are used as the training datasets. The training and testing processes are repeated so that all the subsets are used as a testing dataset. The training set is used to train the parameters

TABLE 6. The best ensemble IDS (Hold-out).

Ensemble IDS	Selected Feature #	Selected Features
SU+BN	10	*Packet_Length_size, IP_Total_Length, ID_Protocol, TCP_SRC_port, TCP_DST_port, TCP_Seq_num, TCP_Ack_num, TCP_offset, TCP_Win,* and *UDP_Length*
IG+NB	4	*Packet_Length_size, IP_Total_Length, TCP_Win*, and *UDP_Length*
CS+BN	4	*Packet_Length_size, IP_Total_Length, TCP_Win, and UDP_Length*
GR+J48	7	*ID_Protocol, TCP_Seq_num, TCP_Ack_num, TCP_offset, TCP_Flags, TCP_Win, UDP_Length*

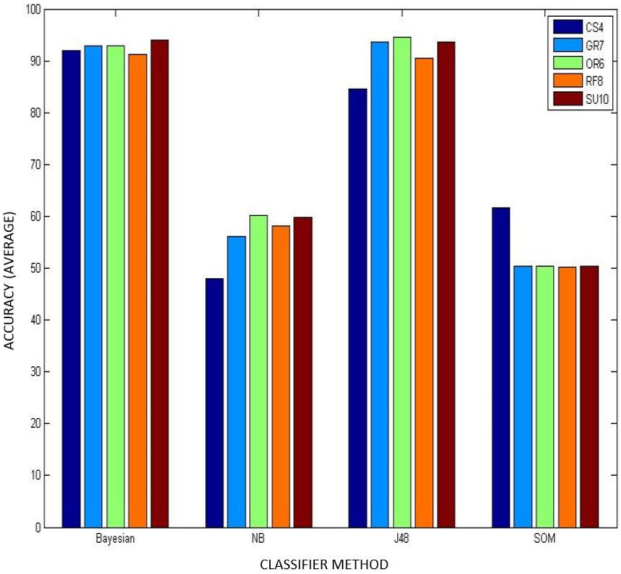

FIGURE 11. Histogram of testing set result from average of 5-Fold validation.

of feature ranking in order to get the optimal solutions, while the testing set is used to test the generalization of the feature ranking parameters performance.

There is a bias-variance trade-off associated with the choice of k in k-fold cross-validation. Typically, given these considerations, one performs k-fold cross validation using k = 5 or k = 10. These values have been shown empirically to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance [67].

Fig. 11 presents the average of classification accuracy based on 5-fold validation measurements of the testing set. The graph shows that SU method, which has ten features has obtained the best result using Bayesian Network classifier, while OR method has six features that are designated as being superior when verified by NB and J48 classifiers, while CS method has four features that verified by SOM as being better than the others. From these results, it can be concluded that the best features for ensemble IDS can be selected from the alternative results that have the best value, i.e.: SU with ten features, OR with six features, and CS with four features as shown in Table 7.

Thus, the alternative best ensemble IDSs are: SU+BN, CS+BN, CS+SOM, IG+NB, OR+BN.

TABLE 7. The best ensemble IDS (K-fold).

Ensemble IDS	Selected Feature #
SU+BN	10
OR+BN	6
CS+SOM	4

TABLE 8. F-measure results for hold-out validation data.

Classifier	Feature	F-Measure for data Training	Testing
BN	CS4	**0.989**	0.810
	GR7	0.970	0.798
	IG4	**0.989**	0.783
	OR6	0.988	0.807
	RF8	**0.989**	0.768
	SU10	0.974	***0.853***
J48	CS4	0.993	0.743
	GR7	0.995	0.782
	IG4	0.994	0.746
	OR6	0.993	***0.830***
	RF8	0.994	0.757
	SU10	**0.996**	0.392
NB	CS4	0.667	0.613
	GR7	0.594	0.446
	IG4	0.675	0.617
	OR6	**0.902**	***0.692***
	RF8	0.602	0.524
	SU10	0.594	0.445
SOM	CS4	**0.636**	***0.581***
	GR7	0.370	0.352
	IG4	**0.636**	0.576
	OR6	0.370	0.352
	RF8	0.371	0.361
	SU10	0.370	0.364

C. F-MEASURE

F-Measure is performance metric for different types of prediction problems, including binary classification, multi-label classification and certain application of structured output prediction. In this experiment, F-measure is used for measuring the performance of the features that ranked by different features selection methods and verified by four machine learning methods. The selected number of features that chosen are based on the best classification accuracy, i.e.: SU with 10 features, CS and IG with 4 features, RF with 8 features, GR with 7 features, and OR with 6 features to determine the value of its F-measure. The F-measure of classification process listed is in Table 8.

TABLE 9. Descriptive statistics result.

Classifier	N	Mean (%)	Std. Deviation (%)	Std. Error (%)	95% Confidence Interval for Mean (%)		Minimum (%)	Maximum (%)
					Lower Bound	Upper Bound		
1	3	92.9228	1.01037	.58334	90.4129	95.4327	91.90	93.92
2	3	55.9787	6.90542	3.98684	38.8247	73.1327	48.01	60.21
3	3	90.9330	5.51246	3.18262	77.2393	104.6267	84.58	94.50
4	3	54.0560	6.51338	3.76050	37.8759	70.2361	50.30	61.58
Total	12	73.4726	19.86702	5.73511	60.8497	86.0955	48.01	94.50

TABLE 10. Test of homogeneity of variances percentage.

Levene Statistics	df1	df2	Sig.
3.931	3	8	.054

In Bayesian Network classifier, the best result of F-measure for training data is for CS method with four features, IG method with four features, RF method with eight features, i.e.: 0.989, while in testing data is for SU method with 10 features i.e. 0.853. In decision tree-J48 classifier, the best result of F-measure for training data is for SU method with ten features, i.e.: 0.996, while in testing data is for OR method with 6 six features, i.e.: 0.830. In Naïve Bayesian classifier, the best result of F-measure for training data is for OR method with six features, i.e.: 0.902, while in testing data is at 0.692. In SOM classifier, the best result of F-measure for training data is for IG method with four features, i.e.: 0.636, while in testing data is for CS method with four features, i.e.: 0.581.

In Bayesian Network classifier, the best result of F-measure for training data is for OR method with six features, i.e.: 0.975, while in testing data is for IG method with four features, i.e.: 0.978. In decision tree-J48 classifier, the best result of F-measure for training data is for RF method with eight features, i.e.: 0.991, while in testing data is for OR method with six features, i.e.: 0.984. In Naïve Bayesian, the best result of F-measure for training data is for SU method with ten features, i.e.: 0.582, while in testing data is for OR method with six features, i.e.: 0.575. In SOM, the best result of F-measure for training data is for CS method with four features, i.e.: 0.676, while in testing data is for IG method with four features, i.e.: 0.619.

D. STATISTIC TEST

The statistics observation is also carried out to interpret the data as to ensure that the data obtained from experimentations produce a valid conclusion. As mentioned earlier, the best feature selection is selected based on the best accuracy classification that resulted from each classifier method, i.e.: BN, Naïve Bayesian, J48 and SOM. Each classifier method has a group of percentage classifications based on feature ranking. The statistical test is used to analyze whether the different classifier methods had an effect on changes of the percentage classification accuracy values of feature rankings. The 4 classifier methods labeled as BN = 1, NB = 2, J48 = 3, and SOM = 4, and have numerical data types with scaled measurement. The result of classification percentages for each feature ranking is a numerical data type with nominal measurement. Meanwhile, the descriptive statistics that provide simple summaries of the data and observations that have been made about the experiment are listed in Table 9. In this descriptive statistics box, the mean of percentage classification value of Bayesian Network result is 92.9228%; Naïve Bayesian is 55.9787%, Decision Tree: J48 is 90.9330%, and SOM is 54.0560%, while overall mean value is 73.4726%. The number of data in each classifier method, N = 3, they are: the classification percentage value of SU10 (SU with 10 features, OR6 (OR with 6 features) and CS4 (CS with 4 features) for each method. Other data are described in the same row in another column.

Levene's Test is used to test the equality or homogeneity of variances as listed in Table 10. The test determines whether the four classifiers have either similar or different amounts of variability between scores.

The calculated value of the Sig. is 0.054. For $\alpha = 0.05$, the calculated value of Sig. is greater than $\alpha = 0.05$, it means that the variability of the four classifiers is about the same or homogeny. The scores in one classifier do not vary too much when compared with the scores in other classifiers. This measurement means that the variability in the four classifiers is not significantly different. To make a conclusion about whether the different classifier methods have an effect on the value of percentage classifications between groups or within groups of feature rankings, the means can give a head start in interpretation. Since the data in each group is not related in any way and only have one independent variable, 1-Way between subjects, ANOVA test will be used to determine whether the differences between groups' means are significant.

ANOVA hypothesis suggests the null hypothesis that there is no significant difference between the means of the percentage classification results of feature rankings. ANOVA's hypothesis is given as (11),

$$H_0\, false \rightarrow H_1\, true \tag{11}$$

H_0: The classification accuracy result does not depend on the classifier methods.

TABLE 11. ANOVA results.

Percentage

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	4098.649	3	1366.216	44.972	.000
Within Groups	243.034	8	30.379		
Total	4341.683	11			

TABLE 12. Multiple comparisons.

Dependent Variable: Percentage

	(I) 1:"Bayesian" 2:"NB" 3:"J48" 4:"SOM"	(J) 1:"Bayesian" 2:"NB" 3:"J48" 4:"SOM"	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
						Lower Bound	Upper Bound
Dunnett T3	1	2	36.94409*	4.02929	.031	7.5761	66.3121
		3	1.98976	3.23564	.975	-21.0039	24.9834
		4	38.86676*	3.80548	.025	11.2857	66.4478
	2	1	-36.94409*	4.02929	.031	-66.3121	-7.5761
		3	-34.95433*	5.10137	.012	-57.8212	-12.0874
		4	1.92267	5.48054	.999	-22.0495	25.8949
	3	1	-1.98976	3.23564	.975	-24.9834	21.0039
		2	34.95433*	5.10137	.012	12.0874	57.8212
		4	36.87700*	4.92650	.008	15.0503	58.7037
	4	1	-38.86676*	3.80548	.025	-66.4478	-11.2857
		2	-1.92267	5.48054	.999	-25.8949	22.0495
		3	-36.87700*	4.92650	.008	-58.7037	-15.0503

*. The mean difference is significant at 0.05 level.

H_1: Methods will influence the obtained value of classification accuracy.

Thus, to make the decision about whether H_0 is rejected or accepted, the *T-value* and *P-value* must be computed. Table 11 shows the results of 1-Way between subjects of ANOVA. Looking at the Sig. value in the last column, it shows that a significant result has been found F(3, 8) = 44.972, but the significance is given as ".000" so it cannot be concluded that $p < .001$.

A probability of zero means that the result is impossible! What is really meant of course is that the probability rounded to three decimal places is zero. In reality, the probability is really something like.000257 (for example). The most accurate way to report this is by referring to $p < .001$. That is, use the same number of decimal places, change the last digit to 1, and use the < sign. Since the result has a significant F-value, it is known that all the means are not equal (i.e., reject H_o in favor of H_1). However, it does not yet know exactly which means are significantly different to which other means. So it needs to compute a post hoc test. The authors choose Tukey's Test for post-hoc analysis. This test is designed to compare each of four classifiers to every other classifier. Table 12 presents the results of all pair wise comparisons using Tukey's HSD. Looking at the Sig. column in the table, there are some values that are greater than 0.05 and there are some with values that are less than 0.05. If the Sig. value is greater than 0.05, it can be concluded that there is no statistically significant difference between the two conditions being compared. If the Sig. value is less than or equal to 0.05 it is concluded that there is a statistically significant difference between the two conditions being compared. From Table 12, it is observed that BN has statistically significant differences with NB and SOM.

TABLE 13. Homogeneous subset.

Percentage

	1:"Bayesian" 2:"NB" 3:"J48" 4:"SOM"	N	Subset for alpha = 0.05	
			1	2
Tukey B[a]	4	3	54.0560	
	2	3	55.9787	
	3	3		90.9330
	1	3		92.9228

Means for groups in homogeneous subsets are displayed.

a. Uses Harmonic Mean Sample Size = 3.000.

However, Bayesian Network has no statistically significant difference with J48. NB has statistically significant differences with Bayesian Network and J48. However, it has no statistically significant difference with SOM. J48 has statistically significant differences with NB and SOM and it has no statistically significant difference with Bayesian

TABLE 14. Alternatives of selected features based on the statistical test.

No	Ranking Search Method	Number of Selected Method	Features/Attributes names
1	Symmetrical Uncertainty (SU)	10	*Packet_Length_size, IP_Total_Length, ID_Protocol,TCP_SRC_port, TCP_DST_port, TCP_Seq_num, TCP_Ack_num, TCP_offset, TCP_Win, and UDP_Length*
2	One-R (OR)	6	*Packet_Length_size, IP_Total_Length, TCP_Seq_num, TCP_Ack_num, TCP_Win, UDP_Length.*
3	Chi-Square (CS)	4	*Packet_Length_size,IP_Total_Length, TCP_Win, UDP_Length.*

Network. Meanwhile, SOM has statistically significant difference with Bayesian Network and J48 with no statistically significant differences with NB. For this reason, it can be concluded that the Bayesian Network and J48 classifiers are no significantly different in terms of percentage classification for feature ranking results. Furthermore, NB and SOM are also not significantly different in terms of percentage classification in terms for feature ranking results. The other condition comparisons are significantly different from one another. This means that the percentage value that resulted from Bayesian Network and J48 are significantly different with the percentage value that results from NB and SOM.

In principle, Table 13 reflects the same information as in the previous table. Here Group 2 and Group 4 are grouped together because they do not differ from each other. Group 3 and Group 5 are also grouped together because they do not differ from each other; however, they are different to Group 4 and Group 2. Lastly, Table 14 shows contents alternative selected best features of Symmetrical Uncertainty (SU), One-R and Chi Square classifiers as results from statistical test.

E. DISCUSSION ON THE GENERATED ENSEMBLE IDS

Having done performing four validation procedures, the proposed approach generates the best ensemble IDS. The Hold-out validation approach provides potential best numbers of features; they are 10, 7, 6, and 4 (Table 5). With these selected features, the approach generates 6 ensemble IDSs, i.e.: (SU10+BN), (CS4+BN), (IG4+NB), (OR6+J48), (GR7+J48), and (CS4+SOM). However, after performing the 5-Fold validation test procedure, only 10, 6, and 4 numbers of features are selected (Table 7), thus, the approach generates 3 ensemble IDS, i.e.: (SU10+BN), (OR6+BN), and (CS4+ SOM). Then from the F-measures computation, displayed in Table 8, the approach generates 4 ensemble IDS, i.e.: (SU10+BN), (OR6+J48), (OR6+NB), and (CS4+SOM).

Finally, the statistics tests draw a conclusion that accuracy detection of SU10 and OR6 are not significantly different. Likewise, OR6 and CS4 have the same results, because SU10 is equal to OR6 and OR6 is equal to CS4. So all feature rankings from SU10, OR6, and CS4 can be chosen as the best features for the ITD-UTM IDS dataset. Thus, this ensemble IDS generation is illustrated in Fig. 12.

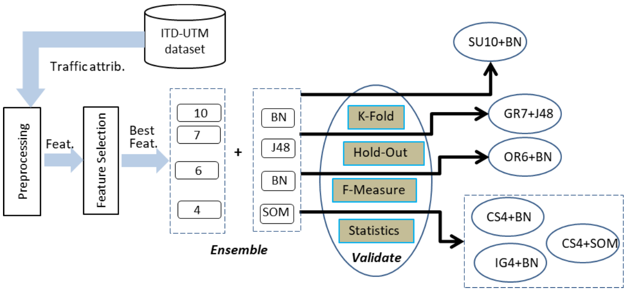

FIGURE 12. The generated ensemble IDS structures.

F. DISCUSSION ON ITD-UTM DATASET

Researchers in [68] have carried out experiment on CICIDS-2017 dataset using ensemble IDS combining IG feature selection technique with Random Forest classifier. They reported the best detection accuracy was 99.86% for 22 selected features. Even more, when ensemble with J48 classifier, the detection accuracy reached the best performance, i.e.: 99.87%. Nevertheless, the computational time is significantly longer, which is trivial. Meanwhile, in this work, the accuracy detection of ensemble IDS (IG+NB) is 81.0316% with number of the best selected features is four. The generated ensemble IDS in this work performs faster in detecting the attack as it involves only 4 selected features, however should sacrifice the accuracy. As this work uses four validation procedures that also considers F-measure and statistics tests, the ensemble IDS (IG+NB) is not considered as the best generated ensemble IDS.

Due to the nature of the CICIDS2017 dataset, that has 78 features [39] while ITD-UTM dataset only has 22, it is observed, that the largest number of selected features was 17. This dataset's characteristic affects the performance of the ensemble IDSs in term of accuracy detection during the testing phase. It is also observed that the highest accuracy detection of the generated ensemble IDSs was 82.8918% (Table 5).

Moreover, the random mechanism applied in dataset division into training dataset and testing dataset also impacts the accuracy performance during the testing. It may happen that numbers of important features are not in the training dataset.

V. CONCLUSION

This study has introduced an approach for constructing ensemble IDS using six ranked feature selection techniques, i.e.: IG, GR, SU, and RF (entropy-based methods); OR, CS (statistical-based methods). These feature selection techniques were ensemble with four feature classifiers, i.e.: Bayes Network, J48, Naïve Bayesian, and SOM. The detection accuracy of these classifiers has been compared.

Considering results from four validation methods, it is concluded that overall, SU feature selection technique with 10 selected features, OR feature selection technique with 6 selected features, and CS feature selection technique with 4 selected features are the best feature selection techniques for the ensemble IDSs on ITD-UTM dataset.

On the other hand, OR feature selection technique with six selected features is superior when it is ensemble with NB or J48 classifiers as they achieved the best F-measure value.

In general, the ITD-UTM dataset is representative enough as new benchmark dataset for conducting researches on IDS.

For future work, authors plan to develop new approach for generating ensemble IDS with considering other FS techniques combine with more than one classifier, and utilizing multiple benchmark datasets.

REFERENCES

[1] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, 2013, doi: 10.1016/j.jnca.2012.09.004.

[2] S. Dua, P. Chowriappa, S. Dua, and P. Chowriappa, "Feature selection and extraction in data mining," in *Proc. Int. Conf. Data Min. Bioinforma.*, 2013, pp. 219–244, doi: 10.1201/b13091-7.

[3] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," in *Proc. Sci. Inf. Conf.*, 2014, pp. 372–378, doi: 10.1109/SAI.2014.6918213.

[4] K. Nahiyan, S. Kaiser, K. Ferens, and R. Mcleod, "A multi-agent based cognitive approach to unsupervised feature extraction and classification for network intrusion detection," in *Proc. Int. Conf. Appl. Cogn. Comput.*, 2017, pp. 25–30.

[5] O. Isaiah, A. Olutola, and O. Olayemi, "Feature or attribute extraction for intrusion detection system using gain ratio and principal component analysis (PCA)," *Commun. Appl. Electron.*, vol. 4, no. 3, pp. 1–4, 2016, doi: 10.5120/cae2016652032.

[6] G. Serpen and E. Aghaei, "Host-based misuse intrusion detection using PCA feature extraction and kNN classification algorithms," *Intell. Data Anal.*, vol. 22, no. 5, pp. 1101–1114, 2018, doi: 10.3233/IDA-173493.

[7] T. A. Alhaj, M. M. Siraj, A. Zainal, H. T. Elshoush, and F. Elhaj, "Feature selection using information gain for improved structural-based alert correlation," *PLoS ONE*, vol. 11, no. 11, 2016, Art. no. e0166017, doi: 10.1371/journal.pone.0166017.

[8] B. Azhagusundari and A. S. Thanamani, "Feature selection based on information gain," *Int. J. Innov. Technol. Exploring Eng.*, vol. 2, no. 2, pp. 18–21, 2013.

[9] H. EzzatIbrahim, S. M. Badr, and M. A. Shaheen, "Adaptive layered approach using machine learning techniques with gain ratio for intrusion detection systems," *Int. J. Comput. Appl.*, vol. 56, no. 7, pp. 10–16, 2012, doi: 10.5120/8901-2928.

[10] N. Farnaaz and M. A. Jabbar, "Random forest modeling for network intrusion detection system," *Procedia Comput. Sci.*, vol. 89, pp. 213–217, May 2016, doi: 10.1016/j.procs.2016.06.047.

[11] A. Panigrahi and M. R. Patra, "An evolutionary computation based classification model for network intrusion detection," in *Proc. Int. Conf. Distrib. Comput. Internet Technol.*, vol. 8956, 2015, pp. 318–324, doi: 10.1007/978-3-319-14977-6_31.

[12] I. S. Thaseen and C. A. Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 29, no. 4, pp. 462–472, 2016, doi: 10.1016/j.jksuci.2015.12.004.

[13] S. Solorio-Fernández, J. F. Martínez-Trinidad, and J. A. Carrasco-Ochoa, "A new unsupervised spectral feature selection method for mixed data: A filter approach," *Pattern Recognit.*, vol. 72, pp. 314–326, Dec. 2017, doi: 10.1016/j.patcog.2017.07.020.

[14] D. Stiawan, "Penetration test & cyber attacks scenario: An overview operating system," Repository Files, Univ. Sriwijaya, Palembang, Indonesia. [Online]. Available: https://repository.unsri.ac.id/39267/

[15] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in cloud," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 42–57, 2013, doi: 10.1016/j.jnca.2012.05.003.

[16] R. Zuech, T. M. Khoshgoftaar, and R. Wald, "Intrusion detection and big heterogeneous data: A survey," *J. Big Data*, vol. 2, no. 1, p. 3, 2015, doi: 10.1186/s40537-015-0013-4.

[17] P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula, "Intrusion detection techniques in cloud environment: A survey," *J. Netw. Comput. Appl.*, vol. 77, pp. 18–47, Jan. 2017, doi: 10.1016/j.jnca.2016.10.015.

[18] F. Kuang, W. Xu, and S. Zhang, "A novel hybrid KPCA and SVM with GA model for intrusion detection," *Appl. Soft Comput.*, vol. 18, pp. 178–184, May 2014, doi: 10.1016/j.asoc.2014.01.028.

[19] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowl.-Based Syst.*, vol. 78, pp. 13–21, Apr. 2015, doi: 10.1016/j.knosys.2015.01.009.

[20] C. Guo, Y. Ping, N. Liu, and S.-S. Luo, "A two-level hybrid approach for intrusion detection," *Neurocomputing*, vol. 214, pp. 391–400, Nov. 2016, doi: 10.1016/j.neucom.2016.06.021.

[21] K. Rai, M. S. Devi, and A. Guleria, "Decision tree based algorithm for intrusion detection," *Int. J. Adv. Netw. Appl.*, vol. 7, no. 4, pp. 2828–2834, 2016. [Online]. Available: https://www.researchgate.net/publication/298175900

[22] R. A. Jamadar, "Network intrusion detection system using machine learning," *Indian J. Sci. Technol.*, vol. 11, no. 48, pp. 1–6, 2018, doi: 10.17485/ijst/2018/v11i48/139802.

[23] M. Nikhitha and M. A. Jabbar, "Nearest neighbor based model for intrusion detection system," *Int. J. Recent Technol. Eng.*, vol. 8, no. 2, pp. 2258–2262, 2019, doi: 10.35940/ijrte.b2458.078219.

[24] I. S. Thaseen, C. A. Kumar, and A. Ahmad, "Integrated intrusion detection model using chi-square feature selection and ensemble of classifiers," *Arab. J. Sci. Eng.*, vol. 44, no. 4, pp. 3357–3368, 2019, doi: 10.1007/s13369-018-3507-5.

[25] V. Bolón-Canedo and A. Alonso-Betanzos, "Ensembles for feature selection: A review and future trends," *Inf. Fusion*, vol. 52, pp. 1–12, Nov. 2019, doi: 10.1016/j.inffus.2018.11.008.

[26] S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, and H. Karimipour, "Cyber intrusion detection by combined feature selection algorithm," *J. Inf. Secur. Appl.*, vol. 44, pp. 80–88, Feb. 2019, doi: 10.1016/j.jisa.2018.11.007.

[27] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Comput. Netw.*, vol. 174, no. Oct. 2019, 2020, doi: 10.1016/j.comnet.2020.107247.

[28] J. J. Davis and A. J. Clark, "Data preprocessing for anomaly based network intrusion detection?: A review," *Comput. Secur.*, vol. 30, nos. 6–7, pp. 353–375, 2011, doi: 10.1016/j.cose.2011.05.008.

[29] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, "Building an intrusion detection system using a filter-based feature selection algorithm," *IEEE Trans. Comput.*, vol. 65, no. 10, pp. 2986–2998, Oct. 2016, doi: 10.1109/TC.2016.2519914.

[30] A. Abd and A. Hadi, "Performance analysis of big data intrusion detection system over random forest algorithm," *Int. J. Appl. Eng. Res.*, vol. 13, no. 2, pp. 1520–1527, 2018. [Online]. Available: http://www.ripublication.com

[31] R. Thomas and D. Pavithran, "A survey of intrusion detection models based on NSL-KDD data set," in *Proc. Inf. Technol. Trends Emerg. Technol. Artif. Intell.*, 2019, pp. 286–291, doi: 10.1109/CTIT.2018.8649498.

[32] A. Tesfahun and D. L. Bhaskari, "Effective hybrid intrusion detection system: A layered approach," *Int. J. Comput. Netw. Inf. Secur.*, vol. 7, no. 3, pp. 35–41, 2015, doi: 10.5815/ijcnis.2015.03.05.

[33] V. Vijayakumar and V. Neelanarayanan, "Intrusion detection model using chi square feature selection and modified Naïve Bayes classifier," in *Smart Innovation, Systems and Technologies*, vol. 49. Cham, Switzerland: Springer, 2016, p. 15, doi: 10.1007/978-3-319-30348-2.

[34] M. El Boujnouni and M. Jedra, "New Intrusion detection system based on support vector domain description with information gain metric," *Int. J. Netw. Secur.*, vol. 20, no. 1, pp. 25–34, 2018, doi: 10.6633/IJNS.201801.20(1).04.

[35] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *J. Comput. Sci.*, vol. 25, pp. 152–160, Mar. 2018, doi: 10.1016/j.jocs.2017.03.006.

[36] K. A. Taher, B. M. Yasin Jisan, and M. M. Rahman, "Network intrusion detection using supervised machine learning technique with feature selection," in *Proc. Int. Conf. Robot. Signal Process. Tech.*, 2019, pp. 643–646.

[37] A. Chandra, S. K. Khatri, and R. Simon, "Filter-based attribute selection approach for intrusion detection using k-means clustering and sequential minimal optimization techniqs," in *Proc. Amity Int. Conf. Artif. Intell. (AICAI)*, 2019, pp. 740–745, doi: 10.1109/AICAI.2019.8701373.

[38] D. Stiawan, M. Y. Idris, and A. H. Abdullah, "Attack and vulnerability penetration testing: FreeBSD," *Telkomnika*, vol. 11, no. 2, pp. 399–408, 2013, doi: 10.12928/TELKOMNIKA.v11i2.886.

[39] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Intl. Conf. Inf. Syst. Secur. Privacy (ICISSP)*, Portugal, Lisbon, Jan. 2018, pp. 108–116.

[40] Z. Karimi, M. Mansour Riahi Kashani, and A. Harounabadi, "Feature ranking in intrusion detection dataset using combination of filtering methods," *Int. J. Comput. Appl.*, vol. 78, no. 4, pp. 21–27, 2013.

[41] S. Bhattacharya and S. Selvakumar, "Multi-measure multi-weight ranking approach for the identification of the network features for the detection of DoS and probe attacks," *Comput. J.*, vol. 59, no. 6, pp. 923–943, 2016, doi: 10.5120/13478-1164.

[42] M. Y. Su, "Real-time anomaly detection systems for Denial-of-Service attacks by weighted k-nearest-neighbor classifiers," *Expert Syst. Appl.*, vol. 38, no. 4, pp. 3492–3498, 2011, doi: 10.1016/j.eswa.2010.08.137.

[43] M. R. Kabir, A. R. Onik, and T. Samad, "A network intrusion detection framework based on Bayesian network using wrapper approach," *Int. J. Comput. Appl.*, vol. 166, no. 4, pp. 13–17, 2017.doi: 10.5120/ijca2017913992.

[44] P. Bereziński, B. Jasiul, and M. Szpyrka, "An entropy-based network anomaly detection method," *Entropy*, vol. 17, no. 4, pp. 2367–2408, Apr. 2015, doi: 10.3390/e17042367.

[45] N. Araájo, "Identifying important characteristics in the KDD99 intrusion detection dataset by feature selection using a hybrid approach," in *Proc. 17th Int. Conf. Telecommun.*, Doha, Qatar, 2010, pp. 552–558, doi: 10.1109/ICTEL.2010.5478852.5.

[46] P. Kushwaha, H. Buckchash, and B. Raman, "Anomaly based intrusion detection using filter based feature selection on KDD-CUP 99," in *Proc. IEEE Region Annu. Int. Conf.*, Dec. 2017, pp. 839–844, doi: 10.1109/TENCON.2017.8227975.

[47] J. Novakovi, P. Strbac, and D. Bulatovi, "Toward optimal feature selection using ranking methods and classification algorithms," *Yugoslav J. Oper. Res.*, vol. 21, no. 1, pp. 119–135, 2016, doi: 10.2298/YJOR1101119N.

[48] I. S. Thaseen and C. A. Kumar, "Intrusion detection model using Chi Square feature selection and modified Naïve Bayes classifier," in *Smart Innovation, Systems and Technologies*, vol. 49, V. Vijayakumar and V. Neelanarayanan, Eds. Cham, Switzerland: Springer, 2016, doi: 10.1007/978-3-319-30348-2_7.

[49] M. A. Salama, H. F. Eid, R. A. Ramadan, and A. Hassanien, "Hybrid intelligent intrusion detection scheme," in *Proc. Soft Comput. Ind. Appl.*, 2011, pp. 293–303.

[50] H. Om and T. Hazra, "Statistical techniques in anomaly intrusion detection system," *Int. J. Adv. Eng. Technol.*, vol. 5, no. 1, pp. 387–398, 2012. [Online]. Available: http://www.archives-ijaet.org/media/37I11-IJAET1111199-Intrusion-detection-system.pdf

[51] A. Moayedikia, K. L. Ong, Y. L. Boo, W. G. Yeoh, and R. Jensen, "Feature selection for high dimensional imbalanced class data using harmony search," *Eng. Appl. Artif. Intell.*, vol. 57, pp. 38–49, Oct. 2016, doi: 10.1016/j.engappai.2016.10.008.

[52] Y. Liu, Y. Wang, X. Ren, H. Zhou, and X. Diao, "A classification method based on feature selection for imbalanced data," *IEEE Access*, vol. 7, pp. 81794–81807, doi: 10.1109/ACCESS.2019.2923846.

[53] K. Anusha and E. Sathiyamoorthy, "Comparative study for feature selection algorithms in intrusion detection system," *Autom. Control Comput. Sci.*, vol. 50, no. 1, pp. 1–9, 2016, doi: 10.3103/S0146411616010028.

[54] T. Zar Phyu and N. N. Oo, "Performance comparison of feature selection methods," in *Proc. MATEC Web Conf.*, vol. 42, 2016, pp. 2–5, doi: 10.1051/matecconf20164206002.

[55] K. K. Vasan and B. Surendiran, "Feature subset selection for intrusion detection using various rank-based algorithms," *Int. J. Comput. Appl. Technol.*, vol. 55, no. 4, pp. 298–307, 2017, doi: 10.1504/IJCAT.2017.086017.

[56] Z. Muda, W. Yassin, M. N. Sulaiman, and N. I. Udzir, "Intrusion detection based on K-means clustering and OneR classification," in *Proc. 7th Int. Conf. Inf. Assurance Secur.*, 2011, vol. 7, pp. 192–197, doi: 10.1109/ISIAS.2011.6122818.

[57] T. Garg and Y. Kumar, "Combinational feature selection approach for network intrusion detection system," in *Proc. 3rd Int. Conf. Parallel, Distrib. Grid Comput.*, 2014, pp. 82–87, doi: 10.1109/PDGC.2014.7030720.

[58] R. A. Ghazy, E. S. M. El-Rabaie, M. I. Dessouky, N. A. El-Fishawy, and F. E. A. El-Samie, "Feature selection ranking and subset-based techniques with different classifiers for intrusion detection," *Wireless Pers. Commun.*, vol. 111, no. 1, pp. 375–393, 2020, doi: 10.1007/s11277-019-06864-3.

[59] K. Shah and D. K. Singh, "A survey on data mining approaches for dynamic analysis of malwares," in *Proc. Int. Conf. Green Comput. Internet Things*, 2015, pp. 495–499, doi: 10.1109/ICGCIoT.2015.7380515.

[60] A. Niranjan, D. H. Nutan, A. Nitish, P. D. Shenoy, and K. R. Venugopal, "ERCR TV: Ensemble of random committee and random tree for efficient anomaly classification using voting," in *Proc. 3rd Int. Conf. Converg. Technol.*, vol. 2018, pp. 1–5, 2018.

[61] S. Sahu and B. M. Mehtre, "Network intrusion detection system using J48 decision tree," in *Proc. Int. Conf. Adv. Comput. Commun. Informat.*, 2015, pp. 2023–2026.

[62] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd Ed. New York, NY, USA: Springer, 2017.

[63] V. K. Pachghare, P. Kulkarni, and D. M. Nikam, "Intrusion detection system using self organizing maps," in *Proc. Int. Conf. Intell. Agent Multi-Agent Syst.*, Chennai, India, 2009, pp. 1–5, doi: 10.1109/IAMA.2009.5228074.

[64] T. Garg and S. S. Khurana, "Comparison of classification techniques for intrusion detection dataset using WEKA," in *Proc. Int. Conf. Recent Adv. Innov. Eng.*, 2014, pp. 1–5, doi: 10.1109/ICRAIE.2014.6909184.

[65] M. Abualkibash, "Machine learning in network security using KNIME analytics," *Int. J. Netw. Secur. Appl.*, vol. 11, no. 5, pp. 1–14, 2019.

[66] S. Russell and P. Norvig *Artificial Intelligence: A Modern Approach*, 3rd ed. London, U.K.: Pearson, 2009.

[67] D. Kurniabudi, D. Stiawan, M. Y. Bin Idris, A. M. Bamhdi and R. Budiarto, "CICIDS-2017 dataset feature analysis with information gain for anomaly detection," *IEEE Access*, vol. 8, pp. 132911–132921, 2020, doi: 10.1109/ACCESS.2020.3009843.

[68] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognit. Lett.*, vol. 15, no. 11, pp. 1119–1125, 1994.

[69] S. I. Ali and W. Shahzad, "A feature subset selection method based on symmetric uncertainty and ant colony optimization," *Int. J. Comput. Appl.*, vol. 60, no. 11, pp. 5–10, 2012.

DERIS STIAWAN received the Ph.D. degree in computer engineering from Universiti Teknologi Malaysia, in 2014. He is currently an Associate Professor with the Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya. His research interests include computer networks, intrusion/ prevention systems, and heterogeneous networks.

AHMAD HERYANTO received the M.Eng. degree in electrical engineering from the Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia, in 2014. He is currently pursuing the Ph.D. degree with Sriwijaya University, Indonesia. His research interests include parallel processing, distributed computing, software security, and network intrusion detection.

ALI BARDADI received the M.Com. degree in information system from Universitas Diponegoro (UNDIP), Semarang, in 2016. He is currently a Lecturer with the Department of Information System, Faculty of Computer Science, Universitas Sriwijaya. His research interests include data warehouse, business intelligence, database, and information systems.

DIAN PALUPI RINI received the Ph.D. degree from Universiti Teknologi Malaysia, in 2017. She is currently a Senior Lecturer with the Informatics Department, Faculty of Computer Science, Universitas Sriwijaya. Her current research interests include soft computing and swarm intelligence.

IMAM MUCH IBNU SUBROTO (Member, IEEE) received the Ph.D. degree from Universiti Teknologi Malaysia, in 2015. He is currently a Senior Lecturer with the Department of Electrical Engineering, Universitas Islam Sultan Agung. His research interests include computer science, artificial intelligence, machine learning, data mining, and education technology.

KURNIABUDI received the master's degree in computer science from Universitas Putra Indonesia YPTK Padang, West Sumatera, Indonesia. He is currently pursuing the Ph.D. degree with the Faculty of Engineering, Universitas Sriwijaya. He is currently a Senior Lecturer with the Faculty of Computer Science, Universitas Dinamika Bangsa, Indonesia. His research interests include technology adoption, information technology, information security, and network security.

MOHD YAZID BIN IDRIS (Member, IEEE) received the M.Sc. degree in software engineering and the Ph.D. degree in information technology (IT) security in 1998 and 2008, respectively. He is currently an Associate Professor with the Faculty of Engineering, School of Computing, Universiti Teknologi Malaysia. In software engineering, he focuses on the research of designing and development of mobile and telecommunication software. His main research interest in IT security includes intrusion prevention and detection (IPD).

ABDUL HANAN ABDULLAH (Member, IEEE) received the Ph.D. degree from Aston University, Birmingham, U.K., in 1995. From 2004 to 2011, he was the Dean of the Faculty of Computer Science and Information Systems. He is currently a Professor with Universiti Teknologi Malaysia (UTM), Johor Bahru, Malaysia. He is also the Head of the Pervasive Computing Research Group under KEconomy Research Alliances, UTM. His research interests include wireless sensor networks, mobile ad hoc networks, network security, and next-generation networks.

BEDINE KERIM received the Ph.D. degree in computer science from the University of Le Havre, Le Havre, France. He is currently an Assistant Professor with the College of Computer Science and Information Technology, Albaha University, Saudi Arabia. He is also a curious Researcher and rigorous academic with good background in artificial intelligence, mathematical modeling, game theory, machine learning, cloud computing, and fuzzy logic.

RAHMAT BUDIARTO received the B.Sc. degree from the Bandung Institute of Technology, in 1986, the M.Eng. and Dr.Eng. degrees in computer science from the Nagoya Institute of Technology, in 1995 and 1998, respectively. He is currently a Full Professor with the College of Computer Science and IT, Albaha University, Saudi Arabia. His research interests include intelligent systems, brain modeling, IPv6, network security, wireless sensor networks, and MANETs.