Wireless Pers Commun (2018) 98:1853–1869
https://doi.org/10.1007/s11277-017-4949-x

A Hybrid Feature Selection Method for Improved Detection of Wired/Wireless Network Intrusions

J. Rene Beulah[1] · **D. Shalini Punithavathani**[1]

Published online: 8 September 2017

Abstract Internet has become an essential aspect of communication in the day to day life of everyone around the world. With the increased usage of Internet, attacks have also increased and the need for various levels of security is on the rise, both in wired and wireless environments. Intrusion detection system (IDS) has become a mandatory level of security for organizations to protect themselves from intruders. Improving the accuracy of IDS is crucial and it is the present focus of researchers. Feature selection has its role in enhancing accuracy by extracting the most relevant features. This study proposes a hybrid method for feature selection that picks and combines the best features from different feature selection methods. This method can be applied for feature reduction in any application domain. In this work, the proposed hybrid method is employed for intrusion detection and six predominant features are picked from NSL-KDD dataset. An exhaustive performance investigation has proved that the proposed feature selection method increases the detection rate by 5% thereby improving the accuracy of intrusion detection system by 3%.

Keywords Intrusion detection · Attribute selection · Classification · NSL-KDD dataset · Performance analysis · IHFS

1 Introduction

Feature selection is an essential component of classification-based knowledge discovery [1]. Feature selection, also known as attribute selection or variable selection or dimensionality reduction is used as a pre-processing step in many application areas, predominantly in data

✉ J. Rene Beulah
renebeulah@gcetly.ac.in

D. Shalini Punithavathani
shalini329@gmail.com

[1] Department of Computer Science and Engineering, Government College of Engineering, Tirunelveli, Tamilnadu, India

mining. Relevant features are selected by examining the information shared between features and class label. Feature selection methods can be predominantly categorized as Filter Model and Wrapper Model as shown in Fig. 1. Wrapper models select a subset of features using the classifier itself whereas filter models are classifier independent and they rank features on the basis of their pertinence to the class label. Filter models are further classified as feature weighting algorithms and subset search algorithms. Feature weighting algorithms estimate the degree of influence of each feature and rank the features accordingly. Subset search algorithms evaluate a subset of features as a group such that correlation among features within a group is less and correlation between each feature and the class is high.

Intrusion detection systems monitor various activities in a network and investigate them for the presence of intrusions. The prime focus of IDS is to detect malicious traffic. Intrusion detection can be considered as a classification task which classifies whether a particular network connection is normal or an intrusion [2]. The datasets used for intrusion detection are high dimensional with regard to the number of records and the number of attributes in each record. The classification task becomes tough and consumes much time when the number of attributes considered is more. The high dimensionality of dataset not only incurs high computational cost, but also deteriorates the generalization ability of learning algorithms [3]. All the attributes may not contribute equally to the classification process. Some may contribute much; some less and some may not contribute at all. Every attribute has a positive or negative impact on the accuracy of IDS. The purpose of feature selection in IDS is to determine the most pertinent features of the incoming traffic [4]. Feature selection removes irrelevant and redundant features and extracts the core features that dominate the classification task [2]. The quality of the selected features mainly determines the effectiveness of the IDS. The main motive behind minimizing the data dimensionality and having the number of features as low as possible is to decrease the training time and to enhance the classification accuracy [5]. In this work, a hybrid feature selection method is proposed to pick the best features for network intrusion detection.

The rest of the paper is structured as follows. Section 2 lists the related work, Sect. 3 discusses the outline of improved hybrid feature selection (IHFS) which can be adopted for feature selection problem of any application, Section 4 describes the details of dataset used, metrics used for performance analysis and methods used for feature selection and classification. Section 5 details the employment of IHFS for intrusion detection, lists the features selected and validates the worth of the features selected. Section 6 presents the results of an in-depth performance analysis and Sect. 7 arrives at the conclusion.

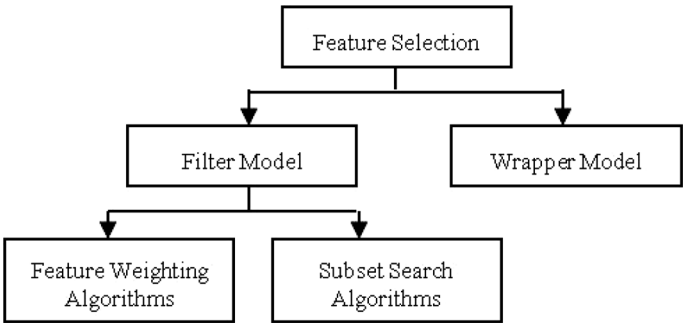

Fig. 1 Classification of feature selection methods

2 Related Work

The emphasis of the work presented in this article is to improve the detection rate of intrusion detection systems by picking the most relevant attributes from the NSL-KDD dataset. Some of the existing works that have performed feature selection using NSL-KDD dataset are recorded in this section. A filtering technique based on Principal Component Analysis is proposed in [6]. Critical Eigenvalue test and screeplot test were used to determine 23 important features. Support Vector Machine (SVM) is used for classification and it shows that training and testing time is reduced by reducing the features. A soft computing approach is used for feature selection in [7] based on Linear Discriminant Analysis (LDA) and Genetic Algorithm (GA). First, LDA is used to convert numeric feature space into linear feature space so as to make classification easier. Then, GA was employed and 11 features were identified as the optimal features. Radial basis function network is used for classification and it is shown that resource utilization and computational cost are minimized and accuracy ratio is increased. In [8], a wrapper feature selection based on Bayesian network classifier is employed to pick 11 features. Sequential search strategy is used to find the feature subset that gives improved classification accuracy to Bayesian Network classifier.

Rough set theory is used for feature reduction in [9]. A rough set tool kit Rosetta is used to construct the discernibility matrix, which is simplified to generate a minimal reduct set or reducts. From the reducts, 27 features were chosen and it is shown that accuracy and sensitivity are increased. Feature selection using Attribute Ratio (AR) is done in [10]. AR is computed using attribute average and class ratio. 22 features having higher AR values are selected and J48 decision tree classifier is used for classification. Correlation-based Feature Subset Selection is applied to select 13 features in [11] and 5 classification algorithms were employed to test the accuracy. It is shown that reducing the features speeds up the classification process and also provides utmost testing accuracy.

Simplified Swarm Optimization, a simplified version of Particle Swarm Optimization and Random Forest are combined to reduce dimensionality to 13 in [12]. Random Forest algorithm is used for classification and it is claimed that feature reduction is essential for improving accuracy. Following this stream, we proposed SHFS [2], a hybrid method for feature selection. Top N features are retrieved using seven well-known feature selection methods. Features that are selected by all the seven methods are chosen as candidate features. All these works have pointed out that feature selection improves accuracy and speeds up training and testing.

In every work mentioned above, a particular subset of features is selected and is assumed to be the optimal subset without any consideration of other possible candidate feature sets. Yet another limitation in some of the works is that only one classifier is used to test the effect of feature reduction.

In the proposed hybrid feature selection method, an extensive study is done by reducing the features gradually and analyzing its impact on detection rate, false alarm rate, classification accuracy, and ROC Area. We have also tested with five different classifiers and made sure that the feature subset is indeed an optimal one.

3 Proposed Method for Feature Selection

This section describes the general structure of Improved Hybrid Feature Selection (IHFS). This feature selection method can be applied to any application domain to select the optimal feature set. There are two main steps in IHFS—Generating Candidate Feature Sets and Finding the Optimal Feature Set.

3.1 Generation of Candidate Feature Sets

First, select x existing best performing feature selection methods suitable for the application by careful performance analysis. Let d be the number of features in the dataset considered. The top N features extracted by different feature selection methods are combined to generate a candidate feature set. This is repeated for different values of N from 1 to d − 1 to get different candidate feature sets as detailed in the proposed algorithm. The procedure is depicted pictorially in Fig. 2. For example, when $N = 2$, the top 2 features extracted by all the feature selection methods are combined to form the candidate feature set CF2.

Algorithm: Generation of Candidate Feature Sets
Input: Dataset D with d features, Feature Selection Methods $FS_1, FS_2, \ldots, FS_x$
Output: Candidate feature sets $CF_1, CF_2, \ldots, CF_{d-1}$
begin
  for $N = 1$ to d-1 do
    $CF_N \leftarrow \{\}$
    for i = 1 to x do
      Apply FS_i for Dataset D
      $TOP_N \leftarrow$ Top N features selected by FS_i
      $CF_N \leftarrow CF_N \cup TOP_N$
    end for
  end for
end

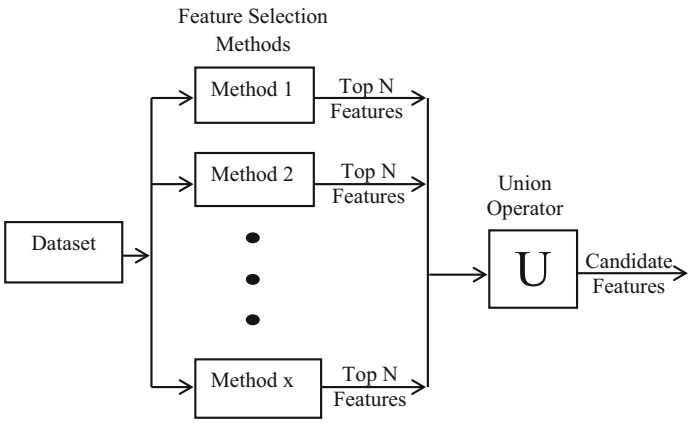

Fig. 2 Candidate feature set generation

3.2 Finding the Optimal Feature Sets

After generating the candidate feature sets, the next step is to detect the best candidate feature set. This is done using the evaluation scheme shown in Fig. 3. Let y denote the number of classifiers chosen for evaluating the candidate feature sets. Usually, a set of features will give good results for a particular kind of classifier. For example, a set of features will be more suitable for tree-based classifiers, whereas another set of features will be more suitable for neural network-based classifiers. So the classifiers chosen should be of different types so that the optimal features selected will work well for any type of classifier. For each candidate feature set, apply the chosen y classifiers and observe the performance of the classifiers for different sets of features. From the results of different classifiers, compute the average classification accuracy and other performance metrics suitable for the application. Pick the candidate feature set that has the overall best performance as the optimal feature set. The optimal features thus selected will be the best representatives of the dataset, since they have been ranked as the best by different methods and have yielded good classification results for different types of classifiers.

4 Dataset, Methods and Metrics Used for Intrusion Detection Problem

This section discusses the employment of IHFS for intrusion detection problem using NSL-KDD dataset.

4.1 NSL-KDD Dataset Description

Implementing and evaluating Intrusion Detection techniques in real time on live network traffic is complicated and so research people usually work with benchmark datasets. KDD_Cup'99 is the most commonly used dataset for Intrusion Detection, which is very huge and redundant. Nowadays, the research community has started using NSL-KDD dataset [13], which has selected records from the complete KDD_Cup'99 dataset. The characteristics of KDD_Cup'99 and NSL-KDD datasets are discussed in [14]. The number of instances in the train and test sets of NSL-KDD dataset is reasonable enabling

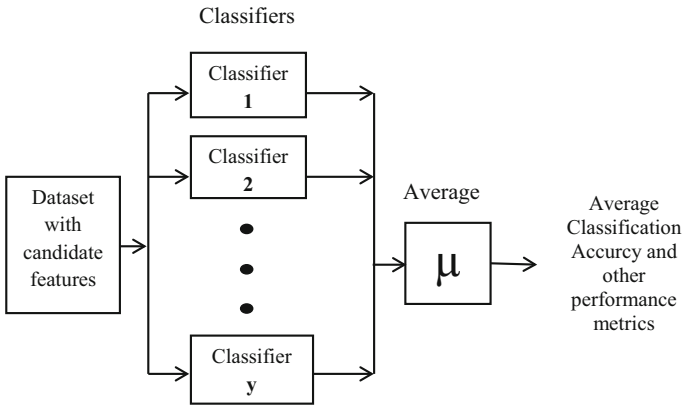

Fig. 3 Scheme for evaluation of candidate features

researchers to work with the complete dataset rather than working on subsets. There are 125,973 instances in the training set and 22,544 instances in the testing set. Each instance is characterized by 41 attributes which are the same as that of KDD_Cup'99 dataset. Table 1 gives a description of these attributes. The training and testing sets have labeled instances representing normal and attack connections. Attacks fall into 4 categories—Denial of Service (DoS) attacks, Probing (Probe) attacks, Remote to Local (R2L) attacks and User to Root (U2R) attacks. The training set contains 22 attacks whereas the testing set contains 37 attacks. The distribution of attacks in the training and testing sets are tabulated in Table 2.

4.2 Evaluation Method

k-fold cross validation is a commonly used method for performance evaluation. When we applied 10-fold cross validation to the training dataset alone, all the classifiers gave good results. Unlike other datasets, the dataset for intrusion detection comes with separate training and testing sets. The testing set is specially designed to contain 17 additional attack types that are not present in the training set so as to check the capability of the intrusion detection techniques to detect new unseen attacks. So, cross validation is not used in our analysis. In our previous work [2], we have discussed about the vast difference in the performance of classifiers when applying cross validation and when separate testing dataset is used. For all the experiments in this study, the training set is used for training the classifier model and the testing set is used for analyzing the performance of the classifier. The experiments were carried out using WEKA [15], a popular machine learning workbench.

4.3 Feature Selection and Classification Methods Used

Many built-in methods are available in Weka for feature selection and classification. The feature selection methods used in this study are described below.

1. CfsSubsetEval (CFS) [16]
   It is a subset search algorithm. It selects a subset of attributes having high correlation with the class and low inter-correlation. The selection of a feature depends on the extent to which it predicts classes in areas of the instance space not already predicted by other features. It imposes a ranking on feature subsets in the search space of all possible feature subsets. Greedy stepwise search is used.
2. GainRatioAttributeEval (GR)
   It is a feature weighting algorithm which assesses the usefulness of an attribute by computing Gain Ratio of the attribute with respect to the class. Gain Ratio is a ratio of information gain or mutual information to the intrinsic information. It takes the number and size of branches into account when choosing an attribute thereby reducing a bias towards multi-valued attributes [17].
3. OneRAttributeEval (OneR)
   It is a wrapper approach for the rule based classifier OneR [18]. OneR is a simple-rule learning system that classifies an object on the basis of a single attribute. It ranks attributes according to error rate on the training set as opposed to entropy-based measures.
4. SymmetricalUncertAttributeEval (SU)
   It is a feature weighting algorithm that measures the usefulness of an attribute by

Table 1 Description of attributes in NSL-KDD dataset

No.	Attribute name	Description
1	Duration	Length (number of seconds) of the connection
2	protocol_type	Type of the connection protocol
3	Service	Network service on the destination
4	flag	Normal or error status of the connection
5	src_bytes	Number of data bytes from source to destination
6	dst_bytes	Number of data bytes from destination to source
7	Land	1 if connection is from/to the same host/port; 0 otherwise
8	wrong_fragment	Number of wrong fragments
9	Urgent	Number of urgent packets
10	hot	Number of "hot" indicators
11	num_failed_logins	Number of failed login attempts
12	logged_in	1 if successfully logged in; 0 otherwise
13	num_compromised	Number of "compromised" conditions
14	root_shell	1 if root shell is obtained; 0 otherwise
15	su_attempted	1 if "su root" command attempted; 0 otherwise
16	num_root	Number of root accesses
17	num_file_creations	Number of file creation operations
18	num_shells	Number of shell prompts
19	num_access_files	Number of operations on access control files
20	num_outbound_cmds	Number of outbound commands in an ftp operation
21	is_hot_login	1 if the login belongs to the "hot" list; 0 otherwise
22	is_guest_login	1 if the login is a "guest" login; 0 otherwise
23	count	Number of connections to the same host as the current connection in the past 2 s
24	srv_count	Number of connections to the same service as the current connection in the past 2 s
25	serror_rate	% of connections that have "SYN" errors (same host connections)
26	srv_serror_rate	% of connections that have "SYN" errors (same service connections)
27	rerror_rate	% of connections that have "REJ" errors (same host connections)
28	srv_rerror_rate	% of connections that have "REJ" errors (same service connections)
29	same_srv_rate	% of connections to the same service (same host connections)
30	diff_srv_rate	% of connections to different services (same host connections)
31	srv_diff_host_rate	% of connections to different hosts (same service connections)
32	dst_host_count	Count of connections having the same destination host
33	dst_host_srv_count	Count of connections having the same destination host and using the same service
34	dst_host_same_srv_rate	% of connections having the same destination host and using the same service
35	dst_host_diff_srv_rate	% of different services on the current host
36	dst_host_same_src_port_rate	% of connections to the current host having the same src port
37	dst_host_srv_diff_host_rate	% of connections to the same service coming from different hosts
38	dst_host_serror_rate	% of connections to the current host that have an S0 error
39	dst_host_srv_serror_rate	% of connections to the current host and specified service that have an S0 error
40	dst_host_rerror_rate	% of connections to the current host that have an RST error

Table 1 continued

No.	Attribute name	Description
41	dst_host_srv_rerror_rate	% of connections to the current host and specified service that have an RST error

computing the symmetrical uncertainty with respect to the class. Symmetric uncertainty measures the correlation between two nominal attributes. This measure helps in finding the smallest subset that perfectly correlates with the class [19].

After careful analysis of the performance of many existing feature selection methods, these methods were identified to be more suitable for the intrusion detection problem. The classifiers described below are used for evaluating the effectiveness of the feature selection algorithms.

1. BayesNet [20]
   It learns a Bayesian Network using a hill climbing search algorithm not restricted by an order on the variables. Bayesian network learning is a two stage process: learning a network structure and learning the probability tables. A Bayesian network is a probabilistic graphical model that represents a set of features and their conditional dependencies using a directed acyclic graph in which nodes represent attributes and edges indicate conditional dependencies.
2. Logistic [21]
   It is a multinominal logistic regression model with a ridge estimator. Logistic regression is a popular method to model binary data. Ridge regression is a good method to estimate stable parameters for the logistic regression model.
3. IB1 [22]
   It is a nearest neighbor classifier. It uses normalized Euclidean distance to find the training instance closest to the given test instance, and predicts the same class as this training instance. If multiple instances have the same smallest distance to the test instance, the first one found is used.
4. NBTree [23]
   It is a tree-based classifier that generates a decision tree with naïve Bayes classifiers at the leaves. NBTree is a hybrid of decision tree classifiers and naïve Bayes classifiers. NBTree induces highly accurate classifiers and is suitable for applications where many attributes are likely to be relevant for a classification task, yet the attributes are not necessarily conditionally independent.
5. SGD with SVM
   It implements Stochastic Gradient Descent (SGD) for learning binary class Support Vector Machine (SVM) with Hinge loss function. SGD is a popular algorithm for training SVM. SGD is an efficient approach to discriminative learning of linear classifiers under convex loss functions like SVM and logistic regression. SGD optimizes an objective function by iteration. It converges almost surely to a global minimum when the objective function is convex.

These classifiers are carefully chosen to be of different types such as Bayesian-based, Regression-based, nearest neighbor based, tree based and SVM based so that the final features picked will work well for any type of classifier.

Table 2 Distribution of attacks in NSL-KDD training and testing sets

S. no.	Name of the attack	Number of instances		Attack type and total No. of instances
		Training set	Testing set	
1	neptune	41,214	4657	DoS
2	teardrop	892	12	Training: 45,927
3	Smurf	2646	665	Testing: 7456
4	Pod	201	41	
5	Back	956	359	
6	Land	18	7	
7	apache2	–	737	
8	processtable	–	685	
9	mailbomb	–	293	
10	udpstorm	–	2	
11	Worm	–	2	
12	ipsweep	3599	141	Probe
13	portsweep	2931	157	Training: 11,656
14	Nmap	1493	73	Testing: 2421
15	Satan	3633	735	
16	Saint	–	319	
17	Mscan	–	996	
18	warezclient	890	–	R2L
19	guess_passwd	53	1231	Training: 995
				Testing: 2756
20	ftp_write	8	3	
21	multihop	7	18	
22	Imap	11	1	
23	warezmaster	20	944	
24	Phf	4	2	
25	Spy	2	–	
26	snmpgetattack	–	178	
27	snmpguess	–	331	
28	Named	–	17	
29	sendmail	–	14	
30	Xlock	–	9	
31	Xsnoop	–	4	
32	Rootkit	10	13	U2R
33	buffer_overflow	30	20	Training: 52
34	loadmodule	9	2	Testing: 200
35	Perl	3	2	
36	Ps	–	15	
37	Xterm	–	13	
38	sqlattack	–	2	
39	httptunnel	–	133	
	Total no. of attacks	58,630	12,833	–
	No. of normal instances	67,343	9711	–

Table 2 continued

S. no.	Name of the attack	Number of instances		Attack type and total No. of instances
		Training set	Testing set	
	Total no. of instances	125,973	22,544	–

4.4 Evaluation Metrics

The metrics used for evaluation are Detection Rate (DR), False Positive Rate (FPR) or False Alarm Rate, Precision, Recall, F-Measure, Area under ROC curve (AUC) and classification accuracy (Acc). Detection rate is the ratio of intrusions identified by the system to the actual number of intrusions in the dataset. DR is also called True Positive Rate (TPR). False Alarm Rate is the ratio of the number of normal events misclassified as attacks to the actual number of normal connections in the dataset. Precision indicates the number of relevant records retrieved whereas Recall signifies the number of relevant records present among the records retrieved. F-Measure is the weighted harmonic mean of precision and recall. Receiver Operator Characteristic (ROC) curve is a graph that demonstrates the performance of the classifier when the threshold is varied. It plots False Alarm Rate (FAR) on the x-axis and Detection Rate (DR) on the y-axis. Classification accuracy is the percentage of instances that are correctly classified. Formulas for all these metrics are given in [2].

5 Proposed IHFS for Intrusion Detection

This section discusses the steps involved in employing IHFS for intrusion detection and explains how the optimal feature set is chosen. The significance of the features selected is also discussed.

5.1 Steps Involved in IHFS

1. Retrieve the top N features using the individual feature selection methods CFS, GR, OneR and SU to get the sets FS_{CFS}, FS_{GR}, FS_{OneR} and FS_{SU} respectively.
2. The candidate set of features selected

$$CF_N = FS_{CFS} \cup FS_{GR} \cup FS_{OneR} \cup FS_{SU}$$

3. Repeat steps 1 and 2 for different values of N to get different sets of candidate features CF_1, CF_2,…, $CF_{40.}$
4. For each candidate feature set CF_i, the 5 classifiers BayesNet, Logistic, IB1, NBTree and SGD with SVM are applied and the average DR, average Acc, average FPR, average F-Measure and average AUC are calculated.
5. The candidate feature set CF_i which yields higher values for DR, Acc, F-Measure and AUC and lower FPR is selected as the optimal feature set.

5.2 Identification of the Optimal Feature Set

The sample candidate feature sets generated are listed in Table 3.

Table 3 Candidate feature sets for different values of N

N	CF_N	No. of attributes
6	{3, 4, 5, 6, 12, 25, 26, 29, 30, 39}	10
5	{3, 4, 5, 6, 12, 25, 26, 30, 39}	9
4	{3, 4, 5, 6, 12, 25, 26, 30}	8
3	{3, 4, 5, 6, 12, 26}	6
2	{3, 4, 5, 12, 26}	5
1	{5, 12}	2

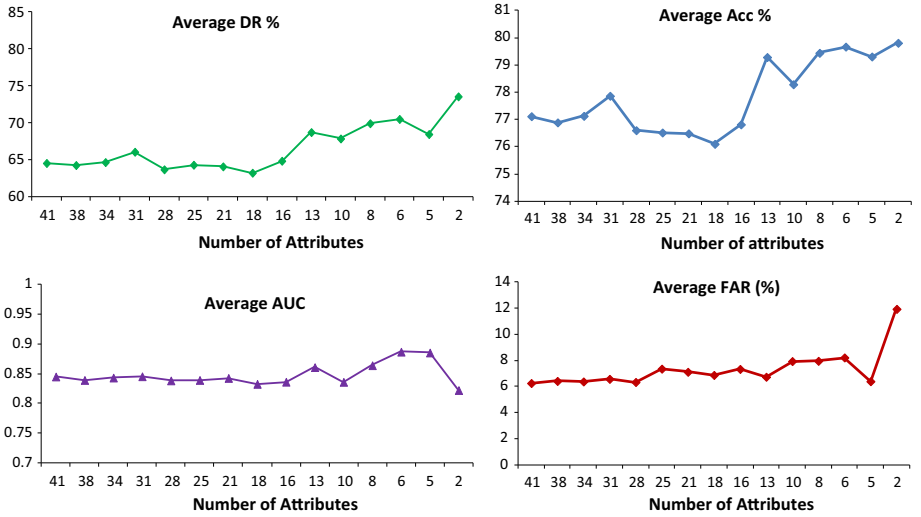

Fig. 4 Average DR, Acc, AUC and FAR on varying the number of features

Figure 4 graphically represents the average values of DR, Acc, AUC and FAR for the 5 classifiers when the number of attributes is reduced gradually. From the graphs, it can be inferred that DR and Acc are the highest with 2 attributes. But FAR for 2 attributes is much higher and AUC is much lower, which makes it unsuitable. The next highest values of DR and Acc can be observed for 6 attributes, for which, FAR is reasonable and AUC is also the highest. Based on rigorous examination of all these parameters, the 6 attributes listed in Table 4 are chosen as the optimal feature set.

Table 4 Optimal feature set selected by IHFS

S. no.	Attribute no.	Attribute name
1	3	service
2	4	flag
3	5	src_bytes
4	6	dst_bytes
5	12	logged_in
6	26	srv_serror_rate

5.3 Significance of the Features Selected

The attribute ***service*** indicates the network service on the destination. It is a discrete valued attribute and it takes 70 discrete values. Some of the examples are auth, courier, http, telnet, ftp, login, name and private. From analysis, it is seen that out of the 70 network services, 44 services are used only in attack connections. So, it is an important attribute to distinguish normal and attack connections.

The attribute ***flag*** specifies the normal or error status of the connection. It is a discrete valued attribute that takes up the values OTH, REJ, RSTO, RSTOSO, RSTR, S0, S1, S2, S3, SF and SH. This attribute indicates whether an attempt to make a connection is successful or not, whether a connection is established and terminated properly, whether a connection is aborted by the originator or responder and other status of the connection. It is also an essential attribute to identify attacks.

The attribute ***src_bytes*** denotes the number of data bytes transferred from source to destination and ***dst_bytes*** denotes the number of data bytes transferred from destination to source. There is a normal range of values for src_bytes and dst_bytes for a particular service. If these values do not fall within the range, it may indicate an attack.

The attribute ***logged_in*** specifies whether a user has successfully logged in or not. It is a binary attribute. For most of the intrusions, logged_in = 0. There are some occurrences of normal cases with logged_in = 0 and some intrusive cases with logged_in = 1. If only this attribute is used for classifying normal and intrusive connections, 79.6% of intrusions can be detected, but it has a high false alarm rate of 24.3%. So this attribute alone cannot be used for classification, but when used with other attributes, this gives valuable information.

The attribute ***srv_serror_rate*** implies the percentage of same service connections that have "SYN" errors. For most of the normal connections, this value is 0.

Thus it is justified that all the 6 attributes extracted by this study provide significant information to classify normal and attack connections.

6 Performance Analysis

The 6 attributes selected by IHFS are compared with the top 6 attributes selected by the individual feature selection methods CFS, GR, OneR and SU and the results are tabulated in Table 5.

From the table it is clear that the attributes selected by the proposed IHFS method have given the highest detection rate, recall and AUC for all the 5 classifiers. This is because, the proposed method has picked the three top ranked attributes from 4 different feature selection methods. The highest recall values for all the classifiers indicate that the features selected helped to retrieve most of the relevant results. F-Measure for IHFS is the highest for BayesNet, IB1, NBTree and SGD with SVM classifiers and is closer to the highest value for Logistic classifier. This implies that the tradeoff between Precision and Recall is acceptable. False Alarm Rate for IHFS is unfortunately higher, which in turn resulted in a little lower precision, but those methods which produced lower False Alarm Rate have exhibited very low Detection Rate which is unacceptable. As the AUC is the highest for IHFS, the tradeoff between DR and FAR is acceptable.

Performance of classifiers with all attributes and with the 6 attributes picked by IHFS is graphically depicted in Fig. 5. From the graph, it is evident that Detection Rate, Recall,

Table 5 Performance comparison of IHFS with other methods

Evaluation measure	Dataset	Classifiers				
		Bayes net	Logistic	IB1	NBTree	SGD with SVM
Detection rate %	IHFS	65.4	67.7	73.3	75.8	68.1
	CFS	52.5	56.8	60.9	74.4	56.9
	GR	28.1	62.8	53.6	56.6	56.9
	OneR	64.1	67.7	72.4	74.3	67.2
	SU	51.8	55.8	64.4	74.3	56.9
False alarm rate %	IHFS	2.9	10.5	9.2	8.6	9.9
	CFS	2.1	1.5	2.4	7.6	0.8
	GR	0.6	1.7	0.9	0.8	0.8
	OneR	2.7	9.8	9.1	8.6	10.0
	SU	0.7	1.2	2.4	7.8	0.8
Precision	IHFS	0.968	0.895	0.913	0.921	0.901
	CFS	0.971	0.980	0.971	0.929	0.989
	GR	0.985	0.980	0.988	0.99	0.989
	OneR	0.969	0.902	0.913	0.92	0.899
	SU	0.989	0.984	0.973	0.927	0.989
Recall	IHFS	0.654	0.677	0.733	0.758	0.681
	CFS	0.525	0.568	0.609	0.744	0.569
	GR	0.281	0.628	0.536	0.566	0.569
	OneR	0.641	0.677	0.724	0.743	0.672
	SU	0.518	0.558	0.644	0.743	0.569
F-measure	IHFS	0.781	0.771	0.813	0.832	0.776
	CFS	0.682	0.719	0.749	0.826	0.722
	GR	0.437	0.765	0.695	0.72	0.722
	OneR	0.772	0.774	0.807	0.822	0.769
	SU	0.68	0.712	0.775	0.825	0.722
Area under ROC curve	IHFS	0.943	0.924	0.821	0.946	0.791
	CFS	0.901	0.893	0.793	0.888	0.780
	GR	0.863	0.875	0.763	0.832	0.780
	OneR	0.95	0.867	0.816	0.943	0.786
	SU	0.9	0.820	0.81	0.87	0.780

F-Measure and AUC have significant improvement with the reduced attributes than with all attributes. The increase in FAR can be compromised by the higher detection rate achieved.

To further analyze the performance of IHFS, the six attributes selected by IHFS are compared with the features selected by four existing methods mentioned in literature survey. Among the eight methods [2, 6–12] mentioned in the literature survey, only four methods [2, 7–9] have mentioned the list of attributes selected. The others have just mentioned the number of attributes selected but not the name of the attributes. So, only these four methods are used for comparison and these existing methods will be hereafter referred to as SHFS [2], GeneticAlg [7], WrapperBayes [8] and RoughSet [9]. The works chosen for comparison, [2, 7–9], have also applied their feature selection methods on the

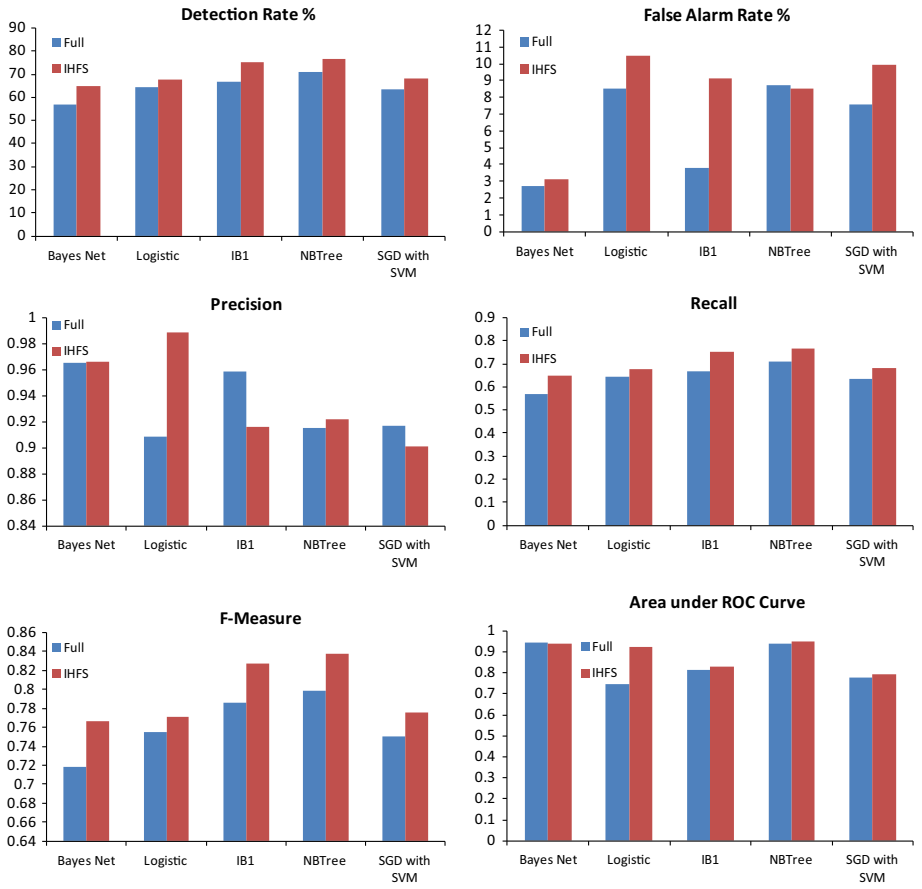

Fig. 5 Performance comparison of IDS with 6 attributes chosen by IHFS and with all attributes

same NSL-KDD dataset and have listed the features that are more relevant. To compare the performance of the features selected by those existing methods with our proposed method, we created five different subsets of NSL-KDD, each having all the records of the NSL-KDD dataset but with only the features suggested by the respective methods SHFS, GeneticAlg, WraperBayes, RoughSet and the proposed IHFS. Five classifiers namely BayesNet, NBTree, Logistic, IB1 and SGD with SVM were applied for all the subsets in the same desktop and the parameters DetectionRate, False Alarm Rate, Precision, Recall, F-Measure and ROC Area are computed.

Figure 6 graphically represents the performance of five classifiers in terms of detection rate, false alarm rate, classification accuracy and area under ROC curve for four existing feature selection methods and the proposed method. From the graphs it is evident that features selected by the proposed method have yielded high detection rate, classification accuracy and AUC. The average performance of different classifiers with all features, features selected by the proposed method and features selected by four existing methods

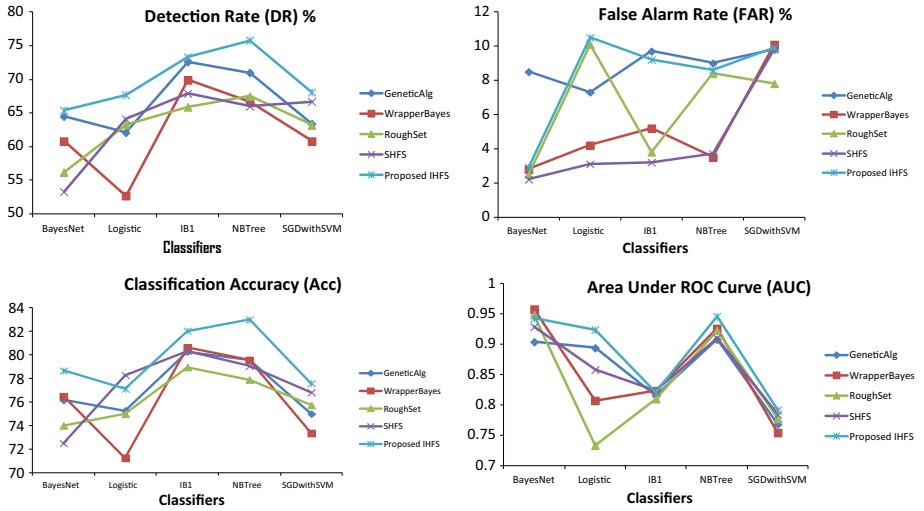

Fig. 6 Performance of different classifiers for IHFS and existing methods

Fig. 7 Average performance analysis of IHFS with existing methods

are graphically depicted in Fig. 7. The proposed IHFS has produced better results for all the metrics except FAR. On an overall comparison with the existing methods, the proposed method shows a significant improvement of 5% in detection rate and 3% in classification accuracy.

7 Conclusion

Feature Selection plays a vital role in increasing the detection rate of an IDS in wired as well as wireless environments and picking the most important attributes needs much analysis. The hybrid method IHFS has picked the top best attributes from 4 different best feature selection methods thereby resulting in enhanced detection of intrusions than with the features retrieved by individual feature selection methods. From the experimental results, we conclude that these 6 attributes (service, flag, src_bytes, dst_bytes, logged_in & srv_serror_rate) contribute the most to the detection of intrusions. We verified the effectiveness of these features with different types of classifiers such as Bayesian network-based, regression-based, nearest neighbor-based, tree-based and SVM-based classifiers. The results demonstrate that our hybrid approach has significantly improved the detection rate and accuracy. Research people who work with NSL-KDD dataset for intrusion detection can use these six attributes instead of all attributes to yield improved results.

References

1. Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2016). A multi-step outlier-based anomaly detection approach to network-wide traffic. *Information Sciences, 348,* 243–271.
2. Beulah, J. R., & Punithavathani, D. S. (2015). Simple hybrid feature selection (SHFS) for enhancing network intrusion detection with NSL-KDD dataset. *International Journal of Applied Engineering Research, 10*(19), 40498–40505.
3. Gu, S., Cheng, R., & Jin, Y. (2016). Feature selection for high-dimensional classification using a competitive swarm optimizer. *Soft Computing*, 1–12.
4. Nguyen, H. T., Petrovic, S., & Franke, K. (2010). A Comparison of feature-selection methods for intrusion detection. In I. Kotenko & V. Skormin (Eds.), *Computer Network Security Lecture Notes in Computer Science* (pp. 242–255). Berlin: Springer.
5. Bennasar, M., Hicks, Y., & Setchi, R. (2015). Feature selection using joint mutual information maximisation. *Expert Systems with Applications, 42*(22), 8520–8532.
6. Heba, F. E., Darwish, A., Hassanien, A. E., & Abraham, A. (2010). Principle components analysis and support vector machine based intrusion detection system. In *2010 10th international conference on intelligent systems design and applications* (pp. 363–367). IEEE.
7. Imran, H. M., Abdullah, A. B., Hussain, M., Palaniappan, S., & Ahmad, I. (2012). Intrusions detection based on optimum features subset and efficient dataset selection. *International Journal of Engineering and Innovative Technology, 2*(6), 265–270.
8. 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 (NAS)* (pp. 307–311). IEEE.
9. Gupta, N., Singh, N., Sharma, V., Sharma, T., & Bhandari, A. S. (2013). Feature selection and classification of intrusion detection system using rough set. *International Journal of Communication Network Security, 2,* 20–23.
10. Chae, H. S., Jo, B. O., Choi, S. H., & Park, T. (2015). Feature selection for intrusion detection using NSL-KDD. *Recent Advances in Computer Science, ISBN*, 978–960.
11. Revathi, S., & Malathi, A. (2013). A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. *International Journal of Engineering Research and Technology. ESRSA Publications.*
12. Revathi, S., & Malathi, A. (2014). Network intrusion detection using hybrid simplified swarm optimization and random forest algorithm on Nsl-Kdd dataset. *IJECS, 3,* 3873–3876.
13. NSL-KDD Dataset. Available on https://web.archive.org/web/20150205070216/http://nsl.cs.unb.ca/NSL-KDD/. Accessed February 2016.
14. Hasan, M. A. M., Nasser, M., Ahmad, S., & Molla, K. I. (2016). Feature selection for intrusion detection using random forest. *Journal of Information Security, 7*(03), 129.
15. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter, 11*(1), 10–18.

16. Hall, M. A. (1999). *Correlation-based feature selection for machine learning*. Doctoral dissertation, The University of Waikato.
17. Han, J., Pei, J., & Kamber, M. (2011). *Data mining: Concepts and techniques*. Upper Saddle River: Elsevier.
18. Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning, 11*(1), 63–90.
19. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data mining: Practical machine learning tools and techniques*. Burlington: Morgan Kaufmann.
20. Bouckaert, R. R. (2008). Bayesian network classifiers in weka for version 3-5-7. *Artificial Intelligence Tools, 11*(3), 369–387.
21. Le Cessie, S., & Van Houwelingen, J. C. (1992). Ridge estimators in logistic regression. *Applied Statistics*, 191–201.
22. Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning, 6*(1), 37–66.
23. Kohavi, R. (1996). Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *KDD* (Vol. 96, pp. 202–207).

J. Rene Beulah is a Full Time Research Scholar of Anna University, Chennai, India pursuing her research in the Department of Computer Science and Engineering, Government College of Engineering, Tirunelveli, affiliated to Anna University, Chennai, India. She received her B.E. in Computer Science and Engineering from Manonmaniam Sundaranar University, Tirunelveli, India in 2004 and M.E. in Computer Science and Engineering from Anna University, Chennai, India in 2006. She was working as a Lecturer in St. Peter's Engineering College, Chennai, India from 2006 to 2011. She is currently working in Network Intrusion Detection. Her research interest includes Network Security and Data Mining.

D. Shalini Punithavathani is the Principal of Government College of Engineering, Tirunelveli, India. She received her B.Sc. in 1979 from Sarah Tucker College, affiliated to Madurai Kamarajar University, India, B.Tech. in Electronics in 1982 from Madras Institute of Technology, affiliated to Anna University, Chennai, India and M.E. in Computer Science and Engineering in 1990 from Government College of Technology, affiliated to Bharathiar University, Coimbatore, India. She got her Ph.D. entitled "Study and Implementation of IPv4 to IPv6 translation techniques" in 2010 from Anna University, Chennai, India. She has 33 years of academic experience. Her research interests include Computer Networks, Mobile Computing, Network Security and Data Mining. She has published many articles in International Journals.