Enhanced relevant feature selection model for intrusion detection systems

Ayman I. Madbouly*

Research and Consultancy Department,
Deanship of Admission and Registration,
King Abdulaziz University,
Jeddah, Saudi Arabia
Email: amadbouly@kau.edu.sa
and
Building Physics and Environment Research Institute,
Housing and Building National Research Center,
Cairo, Egypt
Email: amadbouly@yahoo.com
*Corresponding author

Tamer M. Barakat

Electrical Engineering Department,
Faculty of Engineering,
Fayoum University,
Fayoum, Egypt
Email: tmb00@fayoum.edu.eg

Abstract: With the increased amount of network threats and intrusions, finding an efficient and reliable defence measure has a great focus as a research field. Intrusion detection systems (IDSs) have been widely deployed as effective defence measure for existing networks. IDSs detect anomalies based on features extracted from network traffic. Network traffic has many features to measure. The problem is that with the huge amount of network traffic we can measure many irrelevant features. These irrelevant features usually affect the performance of detection rate and consume the IDSs resources. In this paper, we proposed an enhanced model to increase attacks detection accuracy and to improve overall system performance. We measured the performance of the proposed model and verified its effectiveness and feasibility by comparing it with nine-different models and with a model that used the 41-features dataset. The results showed that, our enhanced model could efficiently achieves high detection rate, high performance rate, low false alarm rate, and fast and reliable detection process.

Keywords: intrusion detection system; classification algorithms; supervised learning; feature selection; data mining.

Reference to this paper should be made as follows: Madbouly, A.I. and Barakat, T.M. (2016) 'Enhanced relevant feature selection model for intrusion detection systems', *Int. J. Intelligent Engineering Informatics*, Vol. 4, No. 1, pp.21–45.

Biographical notes: Ayman I. Madbouly received his BE, ME and PhD degrees in Electronic and Electrical Communication Engineering from the Cairo University, Egypt, Ain-Shams University, Egypt, Fayoum University, Egypt, respectively. In 1997, he joined the National Housing and Building Research Center, Egypt, as an Assistant Researcher, and in 2003, he joined King Abdulaziz University as a Lecturer. Since May 2005, he has been the Head of Computer and Information Technology at the Jeddah Community College, KAU. In 2007, he worked as a college consultant. In 2012, he joined the Department of Research and Consultancy, Deanship of Admission and Registration, KAU as a research consultant. His current research interests include computer networks administration, management, and security. Intrusion detection and prevention systems, cryptography and key management for wireless and sensor networks, information privacy and security, data mining, and machine learning, e-learning and higher education quality.

Tamer M. Barakat received his BSc in Communications and Computers Engineering from Helwan University, Cairo, Egypt, in 2000. He received his MSc in Cryptography and Network Security Systems from Helwan University, in 2004. He received his PhD in Cryptography and Network Security Systems from Cairo University, in 2008. Currently, he is working as a Lecturer, post-doctor researcher and also joining several network security projects in Egypt. His main interest is cryptography and network security. More specially, he is working on the design of efficient and secure cryptographic algorithms, in particular, security in the wireless sensor networks. Other things that interest him are number theory and the investigation of mathematics for designing secure and efficient cryptographic schemes.

1 Introduction

Intrusion detection systems (IDSs) have been widely deployed in computer networks. Nowadays, there are wide spread use of large and distributed computer networks, especially those used in critical systems such as military and commercial systems. Detecting and preventing malicious activities and unauthorised use of such systems is the main function of IDSs. Mainly, there are two approaches to design IDSs, based on the technique used to detect intrusions: *anomaly detection* and *misuse detection* (Axelsson, 2000). Anomaly approach detects intrusions by identifying significant deviations from the normal behaviour profile. Anomaly detection approach is able to detect not only known intrusions but also unknown intrusions. Misuse approach detects intrusion by probing whether previously defined suspicious misuse signatures are present or not in the auditing trails, and any matched activity is considered an attack. Misuse detection approach rarely fails to detect previously known intrusion signatures, but it fails to detect new intrusions never seen before. Anomaly IDSs usually designed using features extracted from raw network traffic data or system audit data. However, with high traffic volume and large-scale networks, we have large amount of features to observe for attack detection. Therefore, IDSs needs to examine large amount of high dimension data even for small network. Hence, IDSs has to meet the challenges of low detection rate, large computation time and complexity. To optimise IDSs detection accuracy and to improve its computational time we need to select relevant features that best distinguish between normal and attack traffic. An efficient feature selection algorithm reduces the number of

selected features by selecting relevant features. Therefore, feature selection plays a key role in designing and building lightweight and robust IDSs while achieving fast and reliable training and testing processes.

Blum and Langley (1997) showed that feature selection approaches fall in three broad categories named *filter*, *wrapper* and *hybrid* approach. Filter approaches use heuristics based on general characteristics of the data to evaluate the worth of features. Filter approach is independent of classification algorithm. Wrapper approaches evaluate the set of features using machine-learning algorithm that will ultimately be employed for learning. A search algorithm searches for the best set of features through the space of all available features. A predetermined classifier evaluates the worth of the selected feature subset. Hybrid approach combines wrapper and filter approach to achieve best possible performance of wrapper approach while preserving low time complexity of filter approach.

In a recent work (Madbouly et al., 2014), we have proposed a relevant feature selection model that selects a set of relevant features to be used in designing a lightweight, efficient, and reliable intrusion detection system. In this research, we modified our recently proposed model algorithm to enhance its detection rate. Despite the previous algorithm achieved good overall detection result; detection results for PROBE, U2R, R2L attack types were low. By modifying this algorithm, we could select a new set of 12-features. We added new features that replaced previously selected features. Updated algorithm could efficiently select features relevant to these attacks. Results of the new proposed model showed higher detection rates, higher performance rates, lower false alarm rates, faster and more reliable detection process.

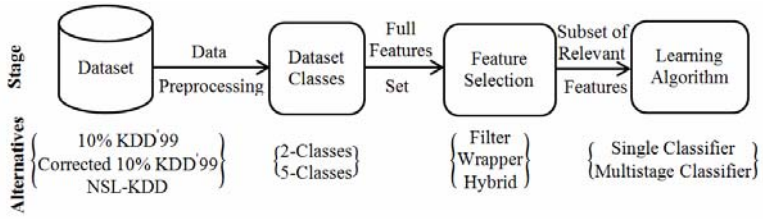
The rest of the paper is organised as follows: Section 2 presents some related researches that cover the topic of using data mining techniques for features selection for IDSs. Section 3 briefly describes the proposed model. Finally, Section 4 describes the experimental results and analysis, followed by the conclusions in the Section 5.

2 Related work

Different researches suggested many algorithms, approaches and methodologies anomaly IDSs. These include machine learning, data mining, statistical, neural networks, information flow analysis, and approaches inspired from human immunology. Many of these approaches and algorithms have been proposed and researched to select the best set of relevant features for IDSs. Effective classification algorithms and mining techniques have been employed including traditional classification (Srinivasulu et al., 2009; Wu and Yen, 2009) and hybrid classification (Srinoy, 2007; Chung and Wahid, 2012; Agarwal and Mittal, 2012; Panda et al., 2011; Singh, 2009). Despite the existence of such different algorithms and approaches, none of them is able to detect all types of intrusion attacks efficiently in terms of the detection accuracy and classifier performance. As a result, recent researches aim to combine the hybrid classification strategy and features selection approaches using data mining to solve many IDSs classification problems and to enhance the detection accuracy of IDSs models and to make smart decisions while detecting intrusions.

Figure 1 shows a block diagram of different alternatives used in each stage of mining approaches for IDSs. Different researches used different combinations of these alternatives.

Figure 1 Data mining approaches for IDSs



Shah and Trivedi (2015) investigated the effectiveness and the feasibility of feature reduction technique on back propagation neural network (BPNN) classifier. They have performed three comparisons: basic, N-fold validation and testing, on reduced dataset with full feature dataset. The three comparisons showed that reduced dataset is better or is equally compatible with no drawback as compared to full dataset. In addition, they showed that usage of such reduced dataset in BPNN could lead to better model in terms of dataset size, complexity, processing time and generalisation ability.

Eesa et al. (2015) presented a new feature-selection approach based on the cuttlefish optimisation algorithm. Their proposed model used cuttlefish algorithm (CFA) as a search strategy and the decision tree (DT) as a classifier. CFA was used to ascertain the optimal subset of features which were judged using DT classifier. They evaluated their proposed model using KDD'99 dataset. The reduced feature subset obtained by using CFA gave a higher detection rate and accuracy rate with a lower false alarm rate, when compared with the obtained results using all features.

Lin et al. (2015) studied the importance of feature representation method on classification process. They proposed cluster centre and nearest neighbour (CANN) approach as a novel feature representation approach. In their approach, they measured and summed two distances. The first distance measured the distance between each data sample and its cluster centre. The second distance measured the distance between the data and its nearest neighbour in the same cluster. They used this new one-dimensional distance to represent each data sample for intrusion detection by a k-nearest neighbour (k-NN) classifier. The proposed approach provided high performance in terms of classification accuracy, detection rates, and false alarms. In addition, it provided high computational efficiency for the time of classifier training and testing.

Zhao et al. (2015) proposed a new model based on immune algorithm (IA) and BPNN. The new developed method is used to improve the detection rate of new intruders in coal mine disaster warning internet of things. IA was used to preprocess network data, extract key features and reduce dimensions of network data by feature analysis. BPNN is adopted to classify the processed data to detect intruders. Experiments' results showed the feasibility and effectiveness of the proposed algorithm with a detection rate above 97%.

Feng et al. (2014) introduced a data classification algorithm based on machine learning. Their proposed approach combined the SVM method with self-organised ant colony network (CSOACN) clustering method. They evaluated their implemented

algorithm using a standard benchmark KDD99 data set. The experimental results showed that combining support vectors with ant colony (CSVAC) outperformed SVM alone or CSOACN alone in terms of both classification rate and run-time efficiency.

De la Hoz et al. (2015) presented a hybrid classification approach based on principal component analysis (PCA) statistical technique and self-organising maps (SOM) machine learning technique. They considered feature selections, noise removal, and low variance features filtering by means of PCA and Fisher discriminant ratio (FDR). The proposed approach modified its classification capabilities by modifying the SOM units' prior activation probabilities to avoid retraining the map. This allowed improving detection accuracy by tuning the detection threshold and enable fast implementations of IDS necessary to cope with current link bandwidths.

Elhag et al. (2015) proposed a new methodology based on genetic fuzzy systems (GFS) with pairwise learning framework for the development of a robust and interpretable IDS. The approach is based on the FARCHD algorithm, a linguistic fuzzy association rule mining classifier, and one-vs.-one (OVO) binarisation methodology in which the binary sub problems are obtained by confronting all possible pair of classes in order to learn a single model for each couple. They tested the goodness and quality of the proposed methodology by means of a complete experimental study versus the state-of-the-art of GFS for IDS. They included C4.5 DT as a baseline rule induction algorithm for comparison. They selected KDD'99 as benchmark dataset. The results showed that the proposed FARCHD-OVO approach has the best tradeoff among all performance measures, especially in the mean F-measure, the average accuracy and the false alarm rate.

Elngar et al. (2013) proposed a hybrid IDS that combines particle swarm optimisation (PSO), information entropy minimisation (IEM) discritisation method, and the hidden naïve Bayes (HNB) classifier. They conducted several experiments using NSL-KDD dataset to evaluate the performance of the proposed IDS. In addition, to validate the proposed IDS they applied a comparative study; such as PCA and gain ratio (GR). They proposed a reduced 11-features subset out of the 41-features. The results showed the adequacy of the proposed network IDS with high intrusion detection accuracy of 98.2% and improved speed of 0.18 sec.

Zhang and Wang (2013) proposed an effective feature selection approach based on Bayesian Network classifier. They compared the proposed approach using the benchmark dataset (NSL-KDD) with other usually used feature selection methods. The empirical results showed that features selected by this approach have decreased the time to detect attacks and increased the classification precision as well as the true positive rates significantly.

Xu et al. (2013) proposed a filter method for unsupervised feature selection based on the geometry properties of L1 graph constructed through sparse coding. Features' local preserving ability was used to evaluate the quality of features. They compared their proposed method with classical Laplace score and Pearson correlation unsupervised methods and with the Fisher score supervised method. The classification results demonstrated the efficiency and effectiveness of the proposed method.

Xu et al. (2012) studied the problem of using imputation quality to search for the meaningful features. They proposed feature selection via sparse imputation (FSSI) method. Sparse representation criterion was utilised to test individual feature. A

comparison with classical feature selection methods Fisher score and Laplacian score was conducted. The results showed the effectiveness of the proposed of FSSI method.

Aziz et al. (2012) and Eid et al. (2013) proposed a genetic algorithm approach (GA) that is used to generate anomalous activities detectors. They addressed the importance of applying discretisation on building network IDS. They proposed to use discretisation for continuous features selected for the intrusion detection. This is used to create homogeneity between data values by replacing values with bin numbers. They explored the impact of the quality of the classification algorithms when combining discretisation with GA. Their proposed detectors generated by GA with smaller population size gave better detection rates, true alarms, and lower false alarms than detectors generated using higher population sizes.

Aziz et al. (2013) proposed an anomaly detectors generation approach using GA in conjunction with several features selection techniques. They applied GA with deterministic crowding niching technique, to generate a set of detectors from a single run. Results showed that sequential-floating techniques used with the GA have higher detection accuracy, especially the sequential floating forward selection technique, compared to others techniques.

Mukherjee and Sharma (2012) investigated the performance of three standard feature selection methods: correlation-based feature selection (CFS), information gain (IG) and GR. They proposed feature vitality-based reduction method (FVBRM) that could identify a subset of 24-important features. They applied naive Bayes classifier on the reduced datasets for intrusion detection. Their empirical results showed that, better performance could be achieved if the selected reduced attributes were used to design efficient and effective IDS.

Chung and Wahid (2012) proposed hybrid intrusion detection systems that use intelligent dynamic swarm-based rough set (IDS-RS) for feature selection and simplified swarm optimisation (SSO) for intrusion data classification. They mentioned 6-features subset out of the 41-features as the most relevant features. For classification, they proposed a new weighted local search (WLS) strategy incorporated in SSO to improve the classification performance. WLS strategy discovered the better solution from the neighbourhood of the current solution produced by SSO. The results showed that the proposed hybrid system could significantly improve the overall performance of the A-NIDS with 93.3% accuracy in average of 20 runs. Furthermore, SSO-WLS managed to outperform the other two most popular benchmark classifiers that are support vector machine (SVM) and naive Bayes.

Li et al. (2012) proposed a gradually feature removal method to choose the critical features that represent various network attacks. They chose a subset of 19-features as the most relevant features. They developed an efficient and reliable classifier to judge a network visit to be normal or not with an accuracy of 98.6249%. The developed classifier was a combination of clustering method, ant colony algorithm and SVM.

Ahmed et al. (2011) proposed a mechanism for optimal features subset selection using PCA, GA and multilayer perceptron (MLP). They used the PCA to project features space to principal feature space and select features corresponding to the highest eigenvalues. However, since the features corresponding to the highest eigenvalues may not have the optimal sensitivity for the classifier due to ignoring many sensitive features. They applied GA to search the principal feature space for genetic eigenvectors that offers a subset of features with optimal sensitivity and the highest discriminatory power. They

proposed a subset of 12-features that increased accuracy, reduced training and computational overheads and simplified the architecture of intrusion analysis engine.

Nguyen et al. (2010) proposed an automatic feature selection approach based on a filter method. Their study focused on correlation feature selection (CFS) to obtain the optimal subset of features. Actually, they transformed the CFS optimisation problem into polynomial mixed (0−1) fractional programming problem, then they applied an improved Chang's method to get mixed (0−1) linear programming problem with linear dependence of the number of constraints and variables on the number of features in the full set. A subset of 9-features was selected and evaluated by C4.5 and Bayes net classifiers. Experimental results showed that the selected subset outperforms the best-first-CFS and GA-CFS methods by removing much more redundant features and still keeping the classification accuracies or even getting better performances.

Chen et al. (2010) proposed a simple and quick inconsistency-based feature selection method. Firstly, they found optimal features by using data inconsistency, and then the sequential forward search is utilised to facilitate the selection of subset features. Their proposed feature selection method can directly eliminate irrelevant and redundant features result in a subset of 14-features. The results showed that the proposed approach reduced the features as well as dataset and achieved good model correctness. The proposed method has a little advantageous than that with the general CFS method.

Zaman and Karray (2009) proposed an enhanced simple method based on support vector decision function (ESVDF). They selected features based on two important factors: the feature's rank (weight) calculated using support vector decision function (SVDF), and the correlation between the features determined by either the forward selection ranking (FSR) or backward elimination ranking (BER) algorithm. Of the total number of 41-features (ESVDF/FSR) algorithm selected 6-features, and (ESVDF/BER) selected 9-features. The proposed approach significantly decreases training and testing times without loss in detection accuracy. Moreover, it selects the features set independently of the classifier used.

Sheen and Rajesh (2008) considered three different approaches for feature selection: chi-square, IG and ReliefF which is based on filter approach. In their comparative study of the three approaches, they evaluated the performance of their selected subset of 20-features by a DT (C4.5) classifier. Of the three features filter approaches chosen they found that chi-square and IG gave better performance than ReliefF. Classification accuracy of chi-square, Info Gain and ReliefF are 95.8506%, 95.8506% and 95.6432% respectively.

Chebrolu et al. (2005) investigated the performance of two feature selection techniques: Bayesian networks (BN), and classification and regression trees (CART). They selected the important features using the Markov blanket model. They found that out of the 41-features, Markov blanket model selected 17 and tested by a classifier constructed using BN. In addition, out of the 41-features, DT model selected 12-features and tested using a CART classifier. The empirical results indicated that normal class is classified 100% correctly and the accuracies of classes U2R and R2L have increased by using the 12-features reduced data set. They observed that CART classifies accurately on smaller data sets. They concluded that the ensemble model of BN classifier and the CART detected, Normal, Probe and DOS with 100% accuracy, U2R, and R2L with 84% and 99.47% accuracies, respectively.

3 The proposed model

The proposed model has four phases, as shown in Figure 2:

Phase 1 data pre-processing

Phase 2 best classifier selection

Phase 3 feature reduction

Phase 4 best feature selection.

3.1 Data pre-processing

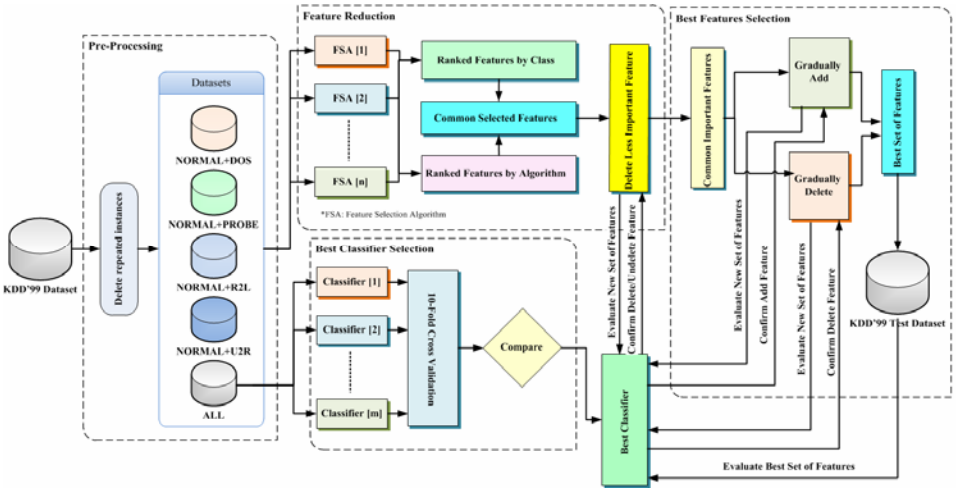
Data mining on huge amounts of data is time-consuming operation, making such analysis impractical or infeasible. Data reduction technique have been used to analyse reduced representation of the dataset without compromising the integrity of the original data and yet producing the quality knowledge. As mentioned by Tavallaee et al. (2009), KDD'99 dataset has some major problems that caused unreliable evaluation results. One major problem is the large number of redundant instances biased learning algorithm to the classes with large repeated instances. While less repeated instances such as U2R and R2L that are usually more harmful to network will have no effect in learning process. We applied data cleansing and data reduction techniques to solve this issue. All repeated instances in the '10% KDD' train dataset and 'Corrected KDD' test set were deleted, and we kept only non-redundant instances.

Table 1 shows the class distribution and statistics of the reduction of repeated records in the KDD'99 dataset. In this phase, we could remove about 70.5% of redundant and repeated records. This large number of redundant and repeated instances (348,435 instances out of 494,021 instances) causes a major problem while training classifiers, and results in biased classification results. Even after removing these records, KDD dataset still has a major problem that affects the classification results. The problem is the unbalanced and inhomogeneous distribution of attacks and normal instances. There are about (*60.33%*) of NORMAL class instances, (*37.48%*) DOS class instances, (*1.46%*) of PROBE class instances, (*0.68%*) of R2L class instances, and (*0.04%*) of U2R class instances. This unbalanced distribution of different classes of KDD'99 dataset biased the classification results to the classes with major instances. This resulted in lower detection performance for classes with low instances, such as U2R and R2L classes. By studying the classification results while using the full 41-features we noticed that most of misclassification occurred between attack classes and Normal class. To solve this issue, we created four class-based datasets: (NORMAL + DOS), (NORMAL + PROBE), (NORMAL + R2L), and (NORMAL + U2R). Each of these dataset contains *all* NORMAL instances plus *all* instances of only one attack type. These four datasets were used along with the original dataset (NORMAL + *all* attack type classes) to search for the best set of most relevant features.

Table 1 10% KDD'99 training dataset preprocessing results

Class	*# of instances before*	*% to all instances*	*# of instances after*	*% to all instances*	*% of reduction*
Normal	97,278	19.69%	87,832	60.33%	9.71%
DOS	391,458	79.24%	54,572	37.48%	86.06%
R2L	1,124	0.23%	997	0.68%	11.30%
U2R	54	0.01%	54	0.04%	0.00%
PORBE	4,107	0.83%	2,131	1.46%	48.11%
Total	*494,021*		*145,586*		*70.53%*

Figure 2 The proposed model framework (see online version for colours)



3.2 Best classifier selection

This phase aimed to find the best classifier that we used in next phases. A comparison between nine different classification algorithms using the 10% KDD'99 training dataset with 41-features was conducted. The selected classifier was used to test the reduced feature sets of the next phase. In addition, this classifier was used to build a lightweight intrusion detection system with the best set of relevant features in the last phase. Figure 3 shows the nine classifiers used in the best classifier comparison. The results showed that ensemble classifier of Adaboost algorithm and C4.5 algorithm (Quinlan, 1993) gives the best performance results while it has the lowest error rate. Figure 4 shows a comparison between different classifiers' root mean squared error (RMSE). Figure 5 shows a comparison between different classifiers' false positive rate (FPR).

From results shown in Figures 4 and 5, we concluded that Lib-SVM, MLP and Bayes net classifiers are not a good choice for our problem domain. These classifiers have lower accuracy and higher error rates when compared with other classifiers. In addition, these classifiers have the worst FPR among all classifiers. The result shows that AdaboostM1-C4.5 ensemble classifiers achieved the best performance among all other

classifiers. However, these classifiers have almost an equivalent or near equivalent performance as AdaboostM1-C4.5. Therefore, we investigated other measures to select the best classifier. We compared the classification performance for the four different best classifiers using 10% KDD'99 41-features dataset and applying ten-fold cross-validation method. Figure 6 shows result of these experiments. From these results, we concluded that AdaboostM1-C4.5 ensemble classifier has the best overall performance compared to other classifiers, especially for detecting U2R and R2L attack classes. Therefore, we select the AdaboostM1-C4.5 ensemble classifier to compare different sets of features that resulted from the next two phases. In addition, this classifier was used to build a lightweight intrusion detection system using the best set of relevant features.

Figure 3 Different classifiers used in the best classifier comparison (see online version for colours)

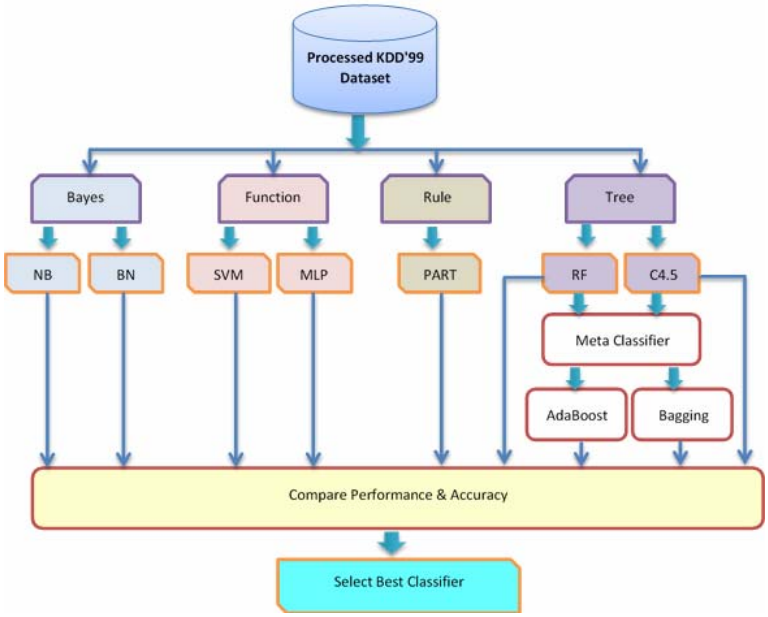


Figure 4 Comparison between different classifiers' RMSE (see online version for colours)

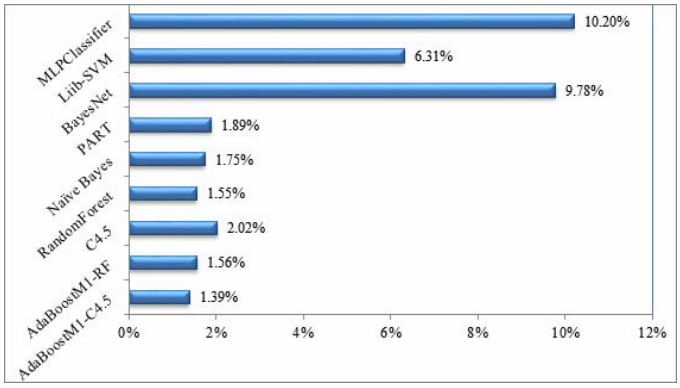


Figure 5 Comparison between different classifiers' FPR (see online version for colours)

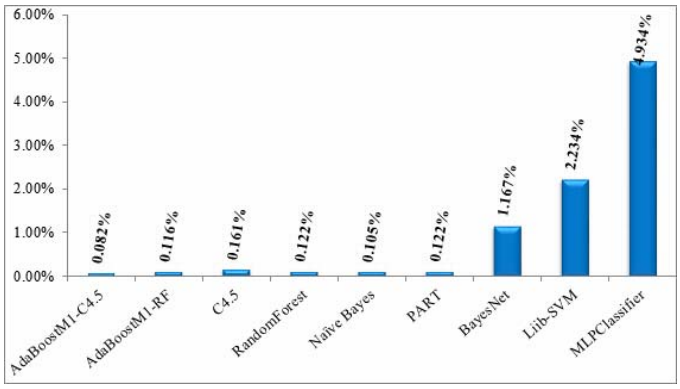
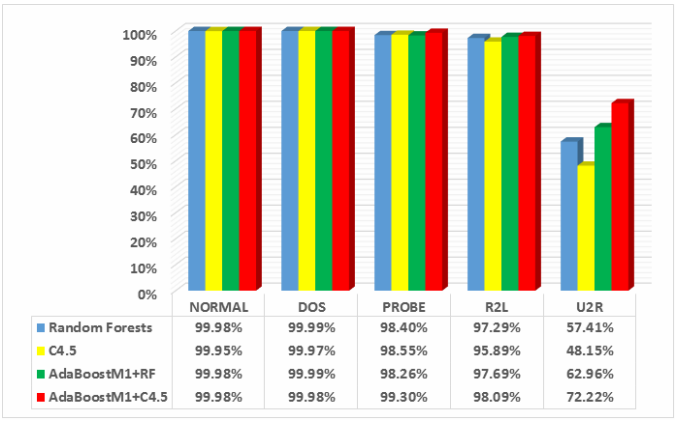


Figure 6 Comparison between different classifiers' true positive rate (see online version for colours)



3.3 Feature reduction

In this phase, irrelevant and less important features were removed. An ensemble for feature evaluation and feature selection algorithms were invoked to select the set of most relevant features. We used correlation-based feature subset selection (CFS) evaluator with seven different search methods as shown in Table 2. Classification performance was measured using ensemble classifier consists of a boosting algorithm, Adaboost M1 method, with C4.5 learning algorithm. The classification was performed using Weka experimenter with ten-fold cross-validation for the testing purposes. Performance measures were calculated by averaging results of a number of ten repetitions.

Table 2 Attribute evaluators and search methods used

Attribute evaluator: correlation-based feature subset selection (CFS)	
Search method	*Description*
Best first	Searches the space of attribute subsets by greedy hill climbing augmented with a backtracking facility.
Evolutionary search	Evolutionary search explores the attribute space using an evolutionary algorithm (EA).
Greedy stepwise	Performs a greedy forward or backward search through the space of attribute subsets.
PSO search	PSO Search explores the attribute space using the particle swarm optimisation (PSO) algorithm
Tabu search	Performs a search through the space of attribute subsets. Evading local maximums by accepting bad and diverse solutions and make further search in the best solutions. Stops when there's not more improvement in n iterations
Rank search (gain ratio)	Evaluates the worth of an attribute by measuring the gain ratio with respect to the class.
Rank search (info gain)	Evaluates the worth of an attribute by measuring the information gain with respect to the class.

Each algorithm evaluated each class dependent dataset created in the previous stage. This resulted in a relevant set of features for each particular class. An average value of feature relevance is calculated as follow.

$$RV_{F_i} = \frac{1}{n}\sum_{j=1}^{n} kA_j \tag{1}$$

where

$$RV_{fi} \equiv Rlevance\ Value\ of\ \ feature\ F_i$$
$$n \equiv number\ of\ \ Evaluation\ \ Algorithms$$
$$k \equiv number\ of\ \ folds\ selected\ F_i\ \ as\ a\ relevant\ \ feature$$

$$A_j = \begin{cases} 1, if\ \ Algorithm\ (j)\ select\ (F_i) \\ \qquad as\ relevant\ \ feature \\ 10\ if\ \ Algorithm\ (j)\ did\ not\ select\ (F_i) \\ \qquad as\ relevant\ \ feature \end{cases}$$

Here we considered only features that are selected by five folds or more (i.e., $k >= 5$). On the other hand, features that not selected by any algorithm were irrelevant and removed from the list. Output of this phase is a reduced set of common relevant features that were ranked by its relevance value for each attack class.

As indicated by Table 3, feature reduction phase reduced the 41-features into 33-features. Features (2, 15, 19, 20, 21, 24, 28, and 41) were not selected as relevant by any algorithm for any attack class. Features (1, 13, 14, 17, and 32) are relevant for U2R

class only. Features (9, 10, 11, 16, 18, and 36) are relevant for R2L class only. Features (27, 40) are relevant for PROBE class only. Features (7, 8, and 31) were selected as relevant while using the ALL class only. Finally, the remaining 17-features are relevant for DOS as well as other Classes.

Table 3 Common important features for each attack class and their importance rank values

U2R		*R2L*		*PROBE*		*DOS*		*ALL*	
F#	*RV***	*F#*	*RV*	*F#*	*RV*	*F#*	*RV*	*F#*	*RV*
14	10.0	*10*	10.0	*25*	10.0	*29*	10.0	*25*	10.0
17	9.8	*26*	8.8	*29*	10.0	*30*	10.0	*29*	10.0
18	8.8	*9*	7.6	*27*	8.6	*12*	9.6	*30*	10.0
29	8.0	*5*	7.4	*37*	7.3	*37*	9.4	*12*	9.8
39	5.9	*16*	6.9	*4*	3.4	*5*	9.3	*3*	8.8
1	4.9	*22*	4.9	*30*	3.4	*26*	7.9	*4*	8.6
13	3.8	*39*	4.9	*38*	3.0	*4*	6.8	*37*	8.0
32	3.0	*11*	3.4	*6*	2.5	*6*	5.9	*6*	5.4
33	2.6	*6*	3.0	*5*	2.1	*25*	5.9	*26*	4.9
3	1.3	*3*	1.3	*33*	1.9	*3*	4.9	*39*	4.8
		33	1.3	*3*	1.3	*38*	4.8	*5*	4.6
		36	1.3	*12*	1.3	*39*	3.3	*35*	4.5
		18	1.1	*23*	1.3	*23*	3.1	*38*	4.4
		37	0.8	*34*	1.3	*34*	1.9	*23*	4.0
				35	1.3	*33*	1.3	*8*	3.8
				40	1.3	*35*	1.3	*10*	3.8
				26	1.0	*22*	0.6	*22*	3.3
								34	3.1
								33	3.0
								14	2.5
								11	1.3
								9	1.1
								13	1.0
								7	0.9
								36	0.9
								31	0.6
								32	0.6

Note: *F#: feature number; **RV: relevance value.

3.4 Best features selection

In this phase, we selected the best set of most relevant features. The 33-Features selected in the previous phase were ranked based on their relevance value to each attack class. This phase consist of two separate stages: *Gradually ADD Feature* and *Gradually DELETE Feature*. The idea is to use two different techniques to select the best features. Two ranked features lists were deduced. One for features that are mostly selected by different algorithms. Where the other one for features that are most important to all attack classes. Common features that came at the end of these two ranked lists excluded and deleted one by one. The rest of features re-evaluated again to make sure that deleting these features did not affect the overall detection accuracy and performance. The algorithm used in this phase is shown below.

Algorithm: Best features selection

Input: Datasets with Common reduced Features
Output: A set of most relevant features
*/*Stage 4.1: Gradually Delete Phase*/*
Starting from the common features set CS[i]
Rank the CS[i], U2R[i], R2L [i], PROBE[i], and DOS[i] based on
The importance of the feature to the attack type (relevance value)
How many attack type the feature can detect
How many algorithms select this feature for each attack type
For j = 1 to i
    If a feature is (used to detect ONLY DOS) AND it is (in the lowest ranked list of DOS)
    Else if a feature is (used to detect ONLY PROBE) AND it is (in the lowest ranked list of PROBE)
    Else if a feature is (used to detect ONLY R2L) AND it is (in the lowest ranked list of R2L)
    Else if a feature is (used to detect ONLY U2R) AND it is (in the lowest ranked list of U2R)
    Else if a feature is (used to detect DOS and PROBE) AND it is (in the lowest ranked list of DOS and PROBE)
        Delete this feature
        Update the CS[j]
        Evaluate performance of the updated CS[j]
        If better performance for U2R, R2L, and PROBE
            Confirm feature deletion
            Update CS[j]
            Update BSA
        Else
            keep this feature
            Update CS[j]
            Update BSA
Next j

```
/*End of Gradually Delete Phase*/
/* Stage 4.2: Gradually Add Phase*/
Start by a common selected set CF(i) of features that are:
Selected as important for all attack types
Selected by all algorithms with high relevance value
Evaluate the performance of CF(i) → BSA
Do until Max BSA
    Add the top ranked feature form the U2R(j) set to CF(i)
    Evaluate the performance of CF(i)
    If performance > BSA
      Confirm adding this feature
      Update CF(i)
      Update U2R(j)
      Update BSA
    Else
      Change the feature importance to lowest rank
    Update U2R(j)
    End if
    Add the top ranked feature form the R2L(j) set to CF(i)
    Evaluate the performance of CF(i)
    If performance > BSA
      Confirm adding this feature
      Update CF(i)
      Update R2L(j)
      Update BSA
    Else
      Change the feature importance to lowest rank
      Update R2L(j)
    End if
    Add the top ranked feature form the PROBE(j) set to CF(i)
    Evaluate the performance of CF(i)
    If performance > BSA
      Confirm adding this feature
      Update CF(i)
      Update PROBE(j)
      Update BSA
    Else
      Change the feature importance to lowest rank
      Update PROBE(j)
    End if
```

```
67:       Add the top ranked feature form the DOS(j) set to CF(i)
68:       Evaluate the performance of CF(i)
69:       If performance > BSA
70:         Confirm adding this feature
71:         Update CF(i)
72:         Update DOS(j)
73:         Update BSA
74:       Else
75:         Change the feature importance to lowest rank
76:         Update DOS(j)
77:       End if
78:   Repeat
79:   Return BSA and CF(i)
80:   /*End of Gradually Add Phase*/
```

The best set of relevant features selected is shown in Table 4.

Table 4 The best set of relevant features

Feature #	*Feature name*
1	duration
3	Service
5	src_bytes
6	dst_bytes
10	Hot
14	root_shell
23	Count
27	rerror_rate
33	dst_host_srv_count
35	dst_host_diff_srv_rate
36	dst_host_same_src_port_rate
38	dst_host_serror_rate

4 Experimental results and analysis

We conduct all our experiments using Windows® 7–32 bits operating system platform with core i7 processor 2.4 GHz, 4.0 GB RAM. Weka 3.7.7 machine learning tool (Hall et al., 2009) was used to evaluate the best subset of most relevant features. Various attribute evaluators available in Weka were used to rank all features according to some metrics. In our experiments, correlation-based feature subset selection (CFS) evaluator

was used with seven different search methods as shown in Table 2. The classification performance is measured by using ensemble classifier consists of a boosting algorithm, Adaboost M1 method, with C4.5 learning algorithm. The classification was performed using Weka experimenter with ten-fold cross-validation for the testing purposes. Performance measures were calculated by averaging results of a number of ten repetitions. To demonstrate the performance of the proposed model and the increase in the detection performance with our set of most relevant features, we compared it with different nine-models with different sizes of feature sets along with the KDD'99 full features dataset. Different performance measures were used to verify the effectiveness and the feasibility of the proposed model. These include detection accuracy, true positive rate (TPR), true negative rate (TNR), FPR, false negative rate (FNR), root mean square error (RMSE), relative absolute error (RAE), training and testing times. The comparison results are presented graphically in Figure 5 to Figure 11, as will be described below (Hint: x F refers to x Features). Table 5 summarises and compares different feature selection algorithms. Table 5 shows the algorithm(s) used along with the number of features selected, selected features, learning algorithm, and the evaluation measure(s) used in each case.

Figure 7 shows a comparison between detection accuracy. It is clear that our selected set of 12-features achieved the same performance (99.95%) as KDD'99-41-features (99.95%). Algorithms with lower number of features (Zulaiha-11 features) and (Nguyen-9 features) achieved lower detection accuracy (99.90% and 99.41%) respectively. While other algorithms with higher number of features (Chen-14 features, Sindhu-16 features, Shina-20 features, Xiao-21 features, Gong-21 features, Tamilarasan-25 features) have a detection accuracy (99.94%). The algorithm of (Li-19 features) has lower detection accuracy of (99.78%) while it used larger number of features than some other algorithms. We expected this to happen because of the addition of the two features (feature # 15 *'su_attempted*' and feature # 19 *'num_access_files* ') that are important only for U2R attack class.

Another important performance measures are shown in Figure 8 (TPR and TNR) and Figure 9 (FPR and FNR). As shown in Figures 8 and 9, our model has the same TPR (99.97%) and TNR (99.92%) compared to other models that used larger number of features and compared to the original KDD'99 with 41 features (99.98% and 99.92%) respectively.

The proposed model could efficiently select the set of most relevant features for IDSs. A small number of most relevant features selected (12 out of 41, i.e., 71% reduction of the size of original KDD'99 dataset). By selecting this reduced set of feature we could built a lightweight IDS with fast and reliable training and testing process. This is clear form Figure 10 (Training Times) and Figure 11 (Testing Times). From Figures 10 and 11, it is clear that our model has the lowest training and testing times even when compared with algorithms that used same number of features (Zulaiha-11 F); or less number of features (Nguyen-9 F).

Finally, Figure 12 (RAE) and Figure 13 (RMSE) shows graphical representation of the classification errors. The results shows that our model has lower classification errors compared to others algorithms investigated.

Table 5 Summary and comparison of different feature selection algorithms

Selection method	*Author/ref/year*	*Feature selection algorithm(s)*	*# of selected features*	*Selected features*	*Learning method/classifier(s)*	*Evaluation measure(s)*
Filter	Nguyen et al. (2010)	M01LP from CFS	9	5, 6, 10,12, 14, 22, 29, 37, 41	C4.5 Bayes net	DA1
Wrapper	Othman et al. (2010)	Features selection based on customised features	11	5, 6, 13, 23, 24, 25, 26, 33, 36, 37, 38	JRip, Ridor, PART and decision tree	DR2, FAR
Filter	Chen et al. (2010)	Inconsistency-based feature selection method	14	1, 3, 4, 5, 10, 12, 23, 25, 32, 34, 35, 36, 40, 41	C4.5	TTBM3, DA
Wrapper	Sindhu et al. (2012)	A combined GA and neuro tree method	16	2, 3, 4, 5, 6, 8, 10, 12, 24, 25, 29, 35, 36, 37, 38, 40	Neuro tree	DR
Wrapper	Li et al. (2012)	Gradually feature removal (GFR)	19	2, 4, 8, 10, 14, 15, 19, 25, 27, 29, 31, 32, 33, 34, 35, 36, 37, 38, 40	SVM	DA, training time, test time, MCCavg
Filter	Sheen and Rajesh (2008)	Chi-square, IG, ReliefF	20	2, 3, 4, 5, 12, 22, 23, 24, 27, 28, 30, 31, 32, 33, 34, 35, 37, 38, 40, 41	C4.5	DA
Filter	Xiao and Liu (2009)	Mutual information based algorithm	21	1, 3, 4, 5, 6, 8, 11, 12, 13, 23, 25, 26, 27, 28, 29, 30, 32, 33, 34, 36, 39	C4.5 and SVM	DR, FAR process time
Wrapper	Gong et al. (2011)	Genetic quantum particle swarm optimisation (GQPSO)	21	2, 3, 5, 6, 10, 12, 17, 21, 22, 23, 25, 26, 28, 29, 30, 31, 32, 33, 34, 35, 36	SVM	DR, training time, test time
Filter	Tamilarasan et al. (2006)	Artificial neural network (ANN) and statistical methods	25	1, 2, 3, 5, 8, 10, 12, 13, 22, 24, 25, 26, 27, 28, 29, 30, 33, 34, 35, 36, 37, 38, 39, 40, 41	RBP neural network	DA, FPR FNR
Hybrid	Ayman et al. (2014)	CFS and AdaboostM1-C4.5	12	1, 3, 5, 6, 10, 14, 23, 27, 33, 35, 36, 38	AdaboostM1-C4.5	DA, FPR, FNR, TPR, TNR, training time, testing time, RAE, RMSE

Notes: 1: detection accuracy; 2: detection rate; 3: time taken to build model.

Figure 7 Accuracy comparison (see online version for colours)

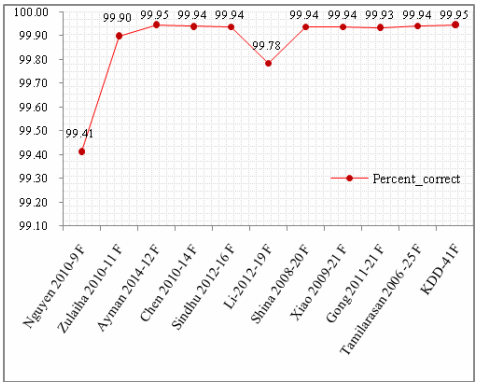


Figure 8 TPR and TNR comparisons (see online version for colours)

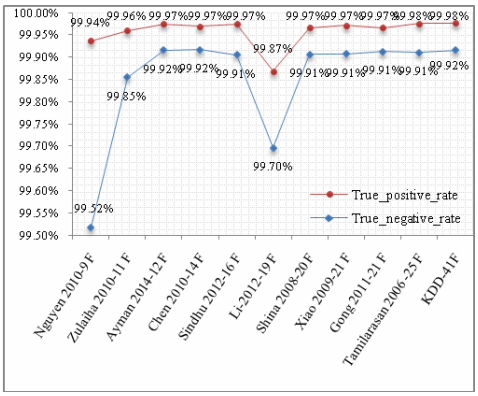


Figure 9 FPR and FNR comparisons (see online version for colours)

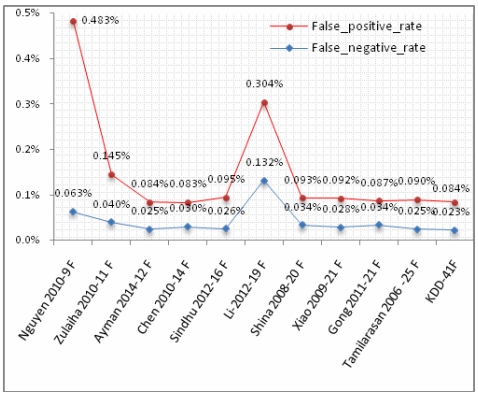


Figure 10 Training times comparisons (see online version for colours)

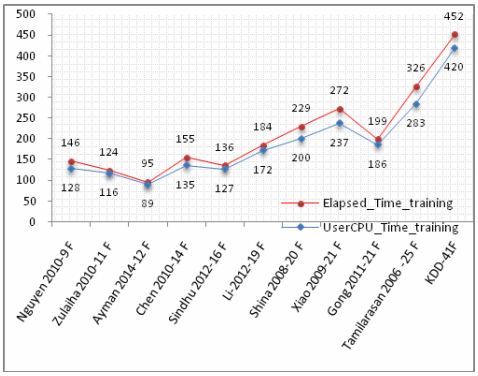


Figure 11 Testing times comparisons (see online version for colours)

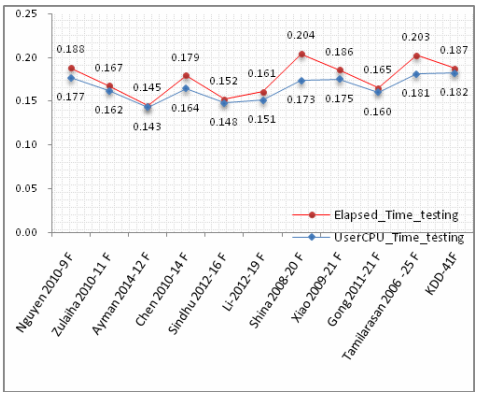


Figure 12 RMSE comparison (see online version for colours)

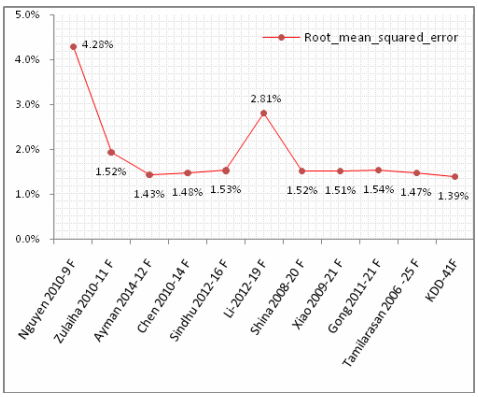


Figure 13 RAE comparison (see online version for colours)

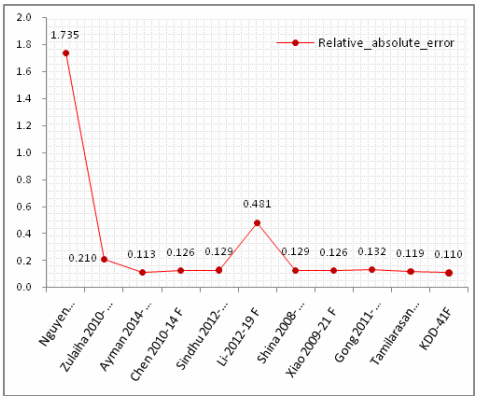


Table 6 Detection confusion matrix – using our most relevant set of 12-features

		Classified as				
		NORMAL	*DOS*	*PROBE*	*R2L*	*U2R*
Actual class	NORMAL	87,811	7	7	4	3
	DOS	4	54,562	6	0	0
	PROBE	10	4	2,117	0	0
	R2L	12	1	0	979	5
	U2R	11	0	0	4	39

Table 7 Detection confusion matrix – using KDD'99 full set with 41-features

		Classified as				
		NORMAL	*DOS*	*PROBE*	*R2L*	*U2R*
Actual class	NORMAL	87,811	2	9	7	3
	DOS	7	54,563	1	1	0
	PROBE	15	0	2,116	0	0
	R2L	14	1	0	978	4
	U2R	14	0	0	1	39

Table 6 and Table 7 show the confusion matrices of detection results using our 12-features set and KDD'99 41-features set respectively. It is clear that with our 12-features set we could achieve same detection accuracy with higher TPR, lower FNR and lower FPR.

5 Results discussion

The proposed model used for feature evaluation and feature selection methods could select a set of 12-best relevant features out of the 41-full features set. Which means that the size of the KDD'99 workbench dataset was reduced by more than 70%. The results showed that features (15, 19, 20, and 21) are not relevant to any intrusion attack type.

While on the other hand, features (1, 14) are highly relevant to detect U2R attacks. In addition, features (10, 36) are highly relevant to detect R2L attacks, where features (27, 38) are highly relevant to detect PROBE attacks. Moreover, features (3, 5, 6, 23, 33, and 35) are highly relevant to detect more than one attack classes, specifically DOS, PROBE, and R2L. The proposed model was able to correctly detect (99.97%) of Normal traffic instances, (99.98%) of DOS traffic instances, (99.3%) of PROBE traffic instances, (98.1%) of R2L traffic instances, and (72.22%) of U2R traffic instances. These results indicated that the selected 12-features achieved almost the same results as the 41-full features set.

6 Conclusions

In this paper, an enhanced model to select a set of most relevant features was proposed. Features relevance analysis using KDD'99 dataset was performed. An ensemble for feature evaluation and feature selection methods was proposed to select a set of best relevant features containing only 12-features out of the 41-full features set. Which reduces the size of the KDD'99 workbench dataset by more than 70%. The proposed model performance was evaluated by comparing its performance measures with recently proposed models using KDD'99 dataset. Results showed that our proposed model could assist in building lightweight IDS that maintains high detection rates with a fast and reliable training and testing while consuming less system resource. The effectiveness and feasibility of the proposed model was verified by several experiments using KDD'99 dataset. The experimental results showed that our enhanced model is not only able to yield high detection rates but also able to speed up the detection process.

Finally, regarding to research limitations, the dataset used is one of the important limitations faced. Although the KDD'99 dataset suffers from some problems discussed above. Moreover, it may not be a perfect representative of existing real networks. However, the lack of public datasets for network-based IDSs, KDD'99 still used as an effective benchmark dataset to help researchers compare different intrusion detection approaches. In future work, we propose to build a new dataset that best represents new and recent real network attacks. We need to have this new dataset as a dynamic dataset open for any updates.

References

Agarwal, B. and Mittal, N. (2012) 'Hybrid approach for detection of anomaly network traffic using data mining techniques', *2nd International Conference on Communication, Computing and Security [ICCCS-2012]*, *Procedia Technol.*, January, Vol. 6, pp.996–1003, doi:10.1016/j.protcy.2012.10.121 [online] http://www.sciencedirect.com/science/article/pii S2212017312006664.

Ahmad, I., Abdulah, A.B., Alghamdi, A.S., Alnfajan, K. and Hussain, M. (2011) 'Feature subset selection for network intrusion detection mechanism using genetic eigen vectors', *Proceedings of 2011 International Conference on Telecommunication Technology and Applications (ICTTA 2011)*, May, pp.75–79 [online] http://s3.amazonaws.com/academia.edu.documents/31038051/13-ICCCM2011-A042.pdf?AWSAccessKeyId=AKIAJ56TQJRTWSMTNPEA&Expires=1439508082&Signature=J9Z4q7swag2JqSxlWgkbRWNEW24%3D&response-content-disposition=inline.

Axelsson, S. (2000) *Intrusion Detection Systems: A Survey and Taxonomy*, Technical Report, Sweden.

Aziz, A.S.A., Azar, A.T., Hassanien, A.E. and Hanafy, S.E. (2012) 'Continuous features discretizaion for anomaly intrusion detectors generation', *The 17th Online World Conference on Soft Computing in Industrial Applications (WSC17)*, 10–21 December.

Aziz, A.S.A., Hassanien, A.E., Azar, A.T. and Hanafy, S.E. (2013) 'Genetic algorithm with different feature selection techniques for anomaly detectors generation', *2013 Federated Conference on Computer Science and Information Systems (FedCSIS)*, Kraków, Poland, 8–11 September.

Blum, A.L. and Langley, P. (1997) 'Selection of relevant features and examples in machine learning', *Artificial Intelligence*, December, Vol. 97, Nos. 1–2, Special Issue on Relevance, pp.245–271, doi:10.1016/S0004-3702(97)00063-5 [online] http://www.sciencedirect.com/science/article/pii/S0004370297000635.

Chebrolu, S., Abraham, A. and Thomas, J.P. (2005) 'Feature deduction and ensemble design of intrusion detection systems', *Comput. Secur.*, June, Vol. 24, No. 4, pp.295–307.

Chen, T., Pan, X. and Xuan, Y. (2010) 'A naive feature selection method and its application in network intrusion detection', *2010 International Conference on Computational Intelligence and Security (CIS)*, pp.416–420.

Chung, Y.Y. and Wahid, N. (2012) 'A hybrid network intrusion detection system using simplified swarm optimization (SSO)', *Appl. Soft Comput.*, September, Vol. 12, No. 9, pp.3014–3022.

De La Hoz, E., Ortiz, A., Ortega, J. and Prieto, B. (2015) 'PCA filtering and probabilistic SOM for network intrusion detection', *Neurocomputing*, 21 September 2015, Vol. 164, pp.71–81, doi:10.1016/j.neucom.2014.09.083 [online] http://www.sciencedirect.com/science/article/pii/S0925231215002982.

Eesa, A.S., Orman, Z. and Brifcani, A.M.A. (2015) 'A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems', *Expert Syst. Appl.*, Vol. 42, No. 5, pp.2670–2679.

Eid, H.F., Azar, A.T. and Hassanien, A.E. (2013) 'Improved real-time discretize network intrusion detection system', *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012) Advances in Intelligent Systems and Computing*, Vol. 201, pp.99–109, doi: 10.1007/978-81-322-1038-2_9.

Elhag, S., Fernández, A., Bawakid, A., Alshomrani, S. and Herrera, F. (2015) 'On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on intrusion detection systems', *Expert Syst. Appl.*, Vol. 42, No. 1, pp.193–202.

Elngar, A.A., Mohamed, D.A.E.A. and Ghaleb, F.F.M. (2013) 'A real-time network intrusion detection system with high accuracy', *The Egyptian Mathematical Society, International Conference on Mathematics, Trends and Development ICMTD12*, 27–29 December 2012, Cairo, Egypt, pp.49–56.

Feng, W., Zhang, Q., Hu, G. and Huang, J.X. (2014) 'Mining network data for intrusion detection through combining SVMs with ant colony networks', *Future Generation Computer Systems*, July, Vol. 37, Special Section: Innovative Methods and Algorithms for Advanced Data-Intensive Computing, pp.127–140.

Gong, S., Gong, X. and Bi, X. (2011) 'Feature selection method for network intrusion based on GQPSO attribute reduction', *2011 Int. Conf. Multimed. Technol.*, July, No. 4, pp.6365–6368.

Hall, M., Frank, E. and Holmes, G. (2009) 'The WEKA data mining software: an update', *ACM SIGKDD*.

Li, Y., Xia, J., Zhang, S., Yan, J., Ai, X. and Dai, K. (2012) 'An efficient intrusion detection system based on support vector machines and gradually feature removal method', *Expert Syst. Appl.*, January, Vol. 39, No. 1, pp.424–430.

Lin, W-C., Ke, S-W. and Tsai, C-F. (2015) 'CANN: an intrusion detection system based on combining cluster centers and nearest neighbors', *Knowledge-Based Systems*, April, Vol. 78, pp.13–21, doi:10.1016/j.knosys.2015.01.009 [online] http://www.sciencedirect.com/science/article/pii/S0950705115000167.

Madbouly, A.I., Gody, A.M. and Barakat, T.M. (2014) 'Relevant feature selection model using data mining for intrusion detection system', *Int. J. Eng. Trends Technol.*, March, Vol. 9, No. 10, pp.501–512.

Mukherjee, S. and Sharma, N. (2012) 'Intrusion detection using naive Bayes classifier with feature reduction', *2nd International Conference on Computer, Communication, Control and Information Technology (C3IT-2012)*, *Procedia Technol.*, 25–26 February 2012, Vol. 4, pp.119–128, doi:10.1016/j.protcy.2012.05.017 [online] http://www.sciencedirect.com/science/article/pii/S2212017312002964.

Nguyen, H., Franke, K. and Petrovic, S. (2010) 'Improving effectiveness of intrusion detection by correlation feature selection', *Availability, Reliab. ..., 2010*.

Othman, Z.A., Bakar, A.A. and Etubal, I. (2010) 'Improving signature detection classification model using features selection based on customized features', *2010 10th Int. Conf. Intell. Syst. Des. Appl.*, November, pp.1026–1031.

Panda, M., Abraham, A. and Patra, M.R. (2011) 'A hybrid intelligent approach for network intrusion detection', *Procedia Eng.*, January 2012, Vol. 30, No. 2011, pp.1–9.

Quinlan, J.R. (1993) *C4.5: Programs for Machine Learning*, 302pp, Morgan Kaufmann Series in Machine Learning, Elsevier, ISBN: 0080500587, 9780080500584.

Shah, B. and Trivedi, B.H. (2015) 'Reducing features of KDD CUP 1999 dataset for anomaly detection using back propagation neural network', *Advanced Computing & Communication Technologies (ACCT), 2015 Fifth International Conference on*, pp.247–251.

Sheen, S. and Rajesh, R. (2008) 'Network intrusion detection using feature selection and decision tree classifier', *TENCON 2008 – 2008 IEEE Reg. 10 Conf.*, pp.1–4.

Sindhu, S.S., Geetha, S. and Kannan, A. (2012) 'Decision tree based light weight intrusion detection using a wrapper approach', *Expert Syst. Appl.*, January, Vol. 39, No. 1, pp.129–141.

Singh, S. (2009) 'An ensemble approach for feature selection of cyber attack dataset', *International Journal of Computer Science and Information Security, IJCSIS*, November 2009, Vol. 6, No. 2, pp.297–302, USA, ISSN 1947-5500 [online] http://arxiv.org/abs/0912.1014.

Srinivasulu, P., Nagaraju, D., Kumar, P.R. and Rao, K.N. (2009) 'Classifying the network intrusion attacks using data mining classification methods and their performance comparison', *Int. J. Comput. Sci. Netw. Secur.*, Vol. 9, No. 6, pp.11–18.

Srinoy, S. (2007) 'Intrusion detection model based on particle swarm optimization and support vector machine', *Computational Intelligence in Security and Defense Applications, 2007: CISDA 2007: IEEE Symposium on*, pp.186–192.

Tamilarasan, A., Mukkamala, S., Sung, A.H. and Yendrapalli, K. (2006) 'Feature ranking and selection for intrusion detection using artificial neural networks and statistical methods', *International Joint Conference on Neural Networks, 2006, IJCNN '06*, July, pp.4754–4761, IEEE, 10.1109/IJCNN.2006.247131 [online] http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1716760.

Tavallaee, M., Bagheri, E., Lu, W. and Ghorbani, A.A. (2009) 'A detailed analysis of the KDD CUP 99 data set', *2009 IEEE Symp. Comput. Intell. Secur. Def. Appl., No. Cisda*, July, pp.1–6.

Wu, S-Y. and Yen, E. (2009) 'Data mining-based intrusion detectors', *Expert Syst. Appl.*, April, Vol. 36, No. 3, pp.5605–5612.

Xiao, L. and Liu, Y. (2009) 'A two-step feature selection algorithm adapting to intrusion detection', *International Joint Conference on Artificial Intelligence, 2009, JCAI '09*, April, pp.618–622, IEEE, 10.1109/JCAI.2009.214 [online] http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5159080.

Xu, J., Yang, G., Man, H. and He, H. (2013) 'L 1 graph based on sparse coding for feature selection', *Advances in Neural Networks–ISNN 2013*, pp.594–601, Springer.

Xu, J., Yin, Y., Man, H. and He, H. (2012) 'Feature selection based on sparse imputation', *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pp.1–7.

Zaman, S. and Karray, F. (2009) 'Features selection for intrusion detection systems based on support vector machines', *2009 6th IEEE Consum. Commun. Netw. Conf.*, January, pp.1–8.

Zhang, F. and Wang, D. (2013) 'An effective feature selection approach for network intrusion detection', *2013 IEEE Eighth Int. Conf. Networking, Archit. Storage*, July, pp.307–311.

Zhao, J-W., Hu, Y., Sun, L-M., Yu, S-C., Huang, J-L., Wang, X-J. and Guo, H. (2015) 'Method of choosing optimal features used to intrusion detection system in coal mine disaster warning internet of things based on immunity algorithm', *Vet. Clin. Pathol.: A Case-Based Approach*, p.157.