Journal of King Saud University – Computer and Information Sciences (2017) **29**, 462–472

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

Intrusion detection model using fusion of chi-square feature selection and multi class SVM

Sumaiya Thaseen Ikram [a,*], **Aswani Kumar Cherukuri** [b]

[a] *School of Computing Science and Engineering, VIT University, Chennai, Tamil Nadu, India*
[b] *School of Information Technology and Engineering, VIT University, Vellore, Tamil Nadu, India*

Received 7 July 2015; revised 4 October 2015; accepted 3 December 2015
Available online 31 March 2016

KEYWORDS

Chi square feature selection;
Cross validation;
Intrusion detection;
Radial basis kernel;
Support vector machine;
Variance

Abstract Intrusion detection is a promising area of research in the domain of security with the rapid development of internet in everyday life. Many intrusion detection systems (IDS) employ a sole classifier algorithm for classifying network traffic as normal or abnormal. Due to the large amount of data, these sole classifier models fail to achieve a high attack detection rate with reduced false alarm rate. However by applying dimensionality reduction, data can be efficiently reduced to an optimal set of attributes without loss of information and then classified accurately using a multi class modeling technique for identifying the different network attacks. In this paper, we propose an intrusion detection model using chi-square feature selection and multi class support vector machine (SVM). A parameter tuning technique is adopted for optimization of Radial Basis Function kernel parameter namely gamma represented by 'ϒ' and over fitting constant 'C'. These are the two important parameters required for the SVM model. The main idea behind this model is to construct a multi class SVM which has not been adopted for IDS so far to decrease the training and testing time and increase the individual classification accuracy of the network attacks. The investigational results on NSL-KDD dataset which is an enhanced version of KDDCup 1999 dataset shows that our proposed approach results in a better detection rate and reduced false alarm rate. An experimentation on the computational time required for training and testing is also carried out for usage in time critical applications.

1. Introduction

Intrusion detection identifies computer attacks by observing various records processed on the network. Intrusion detection models are classified into two variants, misuse detection and anomaly detection systems. Misuse detection can discover intrusions based on a known pattern also known as signatures (Ilgun et al., 1995). Anomaly detection can identify the malicious activities by observing the deviation from normal network traffic pattern (Sumaiya Thaseen and Aswani

* Corresponding author.
E-mail addresses: sumaiyathaseen@gmail.com (I. Sumaiya Thaseen), aswanis@gmail.com (C. Aswani Kumar).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksuci.2015.12.004

Kumar, 2014; Amiri et al., 2011). Hence anomaly detection can identify new anomalies. The difficulty with the current developmental techniques is the high false positive rate and low false negative rate (Sarasamma et al., 2005).

Most of the data mining and bio-informatics applications require processing of large data. A large amount of resources have been utilized in Intrusion Detection Systems (IDS) and several machine learning techniques like decision tree (Lee et al., 2008), genetic algorithm (Shafi and Abbass, 2009), Support vector machines (Khan et al., 2007), Artificial Neural Network (Wang et al., 2010) and hybrid intelligent system (Peddabachigari et al., 2007) are explored to build an IDS. However none of the techniques are able to identify all intrusion attempts and result in a higher detection rate and lower false alarm rate (Panda et al., 2011). Hence there is a need to integrate feature selection and classifier techniques to achieve a better performance.

A model can be learned using supervised or unsupervised learning. Supervised learning requires that the target variable is well known and a sufficient number of values are provided. In unsupervised learning either the target variable is unknown or has been observed only for small number of data.

Support vector machine (SVM) is one of the supervised learning models that has a higher classification efficiency in comparison to other classifier models but due to the higher training time for large data sets, the usage is limited. Hence many feature selection techniques are integrated with SVM to obtain reduced dimensional data. This results in less training time for the classifier. Feature selection is used to select an optimal subset of features for model construction. The feature selection process calculates the score of each probable feature based on a specific feature selection technique and then identifies the best '*k*' features. This procedure is carried out by generating a ranked list of features and different selection criteria can be considered to select a subset of features.

One of the common statistical techniques is the chi-squared that estimates discrepancy from the expected distribution if the feature incidence is not dependent on the class value.

In this paper we put forward an intrusion detection model integrating chi-square feature selection and multi class support vector machine for high accuracy and low false positive rate. The kernel parameter is optimized by obtaining the variance for each attribute feature and determining the highest attribute variance. As the result if kernel is inversely dependent to the variance, a high variance will result in a better kernel parameter. We call this technique as the variance tuning technique.

Many intrusion detection models have been developed with feature selection and classification techniques. The uniqueness of the proposed model over existing intrusion detection approaches is that the optimization of SVM parameters is performed using a variance tuning technique. The variance tuning technique results in a better accuracy in the SVM classifier with minimum time complexity which is detailed in Section 5.1. The average accuracy achieved for all the attacks and normal traffic is more than 95% whereas only the U2R attack accuracy is less as the number of samples involved in training the model is less.

The rest of the paper is structured as follows. The review of various machine learning techniques employed for intrusion detection and the importance of SVM technique for classification along with other feature selection techniques integrated with SVM are introduced in Section 2. The background of various techniques used in the model is detailed in Section 3. The proposed methodology is discussed in Section 4. The experiments and results of the model are reported in Section 5. Section 6 contains the conclusion.

2. Related work

Many hybrid intrusion detection models have been developed to overcome the restrictions of anomaly and misuse detection models. We will analyze the literature of traditional intrusion detection techniques, intrusion models using data mining techniques, intrusion models using single SVM classifiers and integrated intrusion models using SVM and feature selection techniques.

The various techniques used by IDS are statistic (Lazarevic et al., 2003), hidden markov model (Ye and Borror, 2004), artificial neural network (Fisch et al., 2010; Novikov , 2006), fuzzy logic (Saniee Abadeh et al., 2007; Toosi and Kahani, 2007) and rule learning (Xuren et al., 2006). Research in the recent years indicate that SVM can be used for building an intrusion detection model effectively. Fisch et al. (2010) and Mukkamala (2005) have observed the performance of support vector machine, multi variate adaptive regression splines(MARS) and artificial neural network(ANN). It is preferable to build an assembly of classifiers like ANN, MARS and SVM to improve the detection accuracy. Zhang and Shen (2005) used SVM for building an intrusion detection. The system employed text processing methods based on occurrence of system call implemented by the program. Horng et al. (2010) developed a network intrusion detection model using SVM and integrated with BRICH hierarchical clustering for preprocessing. The grouping process reduced the data set thereby decreasing the training time and hence SVM classifiers resulted in higher performance. Ilgun et al. (1995) employed rule based techniques to design and develop IDS, where the expert knowledge is considered as a rule set. Lee et al., (1999) used the data mining technique to create association rules instead of human experts as an analytical model. The drawback of such methods is a large number of association rules are defined thus increasing the complexity of the model.

Due to the large dimensionality of network data, many intrusion models were developed with feature selection considered as a step of preprocessing. Mukkamala (2005) deployed a feature selection technique during preprocessing. At every instance, one input feature is disassociated from the dataset while the residual data set is employed for training and testing. The features are graded based on a set of rules pertaining to the classifiers performance before and after feature selection. Chebrolu et al. (2005) categorized primary features in constructing an IDS that is very crucial for real world detection. Markov model and decision tree has been used in the feature selection process. Bayesian network combined with regression trees were used to build the intrusion detection model. Sung and Mukkamala (2003) eliminated one feature at every time instance to conduct experiments on SVM integrated with neural network. The authors used only 34 significant features rather than all 41 feature sets and obtained a significant performance change in the intrusion detection. Zaman (2009) developed a feature selection technique to construct a lightweight IDS. The proposed approach employed a fuzzy enhanced support vector

decision function (Fuzzy ESVDF) to improve efficiency. The IDS advances in scalability, extendibility resulting in satisfactory system performance. Amiri et al. Amiri et al. (2011) developed a simple and effective feature selection technique according to mutual information technique. The authors investigated both linear correlation and mutual information and the proposed method resulted in better accuracy especially for the minority attacks. Senthilnayaki et al. (2014) built an IDS model with gain ratio as feature selection technique and two classification techniques namely support vector machine and rule based classification were used for identifying the class label. The method however achieved higher accuracy levels only for DoS attacks. Farrahi and Ahmadzadeh (2015) developed an intrusion detection model by using k-means clustering and multiple classifiers such as Naïve Bayes, support vector machine and OneR algorithms. This model resulted in a better accuracy for normal traffic and DoS attack only whereas the false alarm rate was higher for Probe,U2R and R2L attacks. Saxena and Richariya (2014) built an intrusion detection model using gain ratio as the feature selection technique and SVM integrated with particle swarm optimization (PSO) was deployed as the classifier. The resulted accuracy levels were high but the time computation of employing SVM with PSO was not analyzed which is a crucial factor when optimization is performed.

Thus many hybrid models integrating feature selection and classification technique were developed to improve prediction accuracy. Kasliwal et al. (2014) developed a hybrid model by integrating Latent Dirichlet Allocation(LDA) and genetic algorithm(GA). LDA performs the identification of an optimal set of attributes for classification and GA is used for computing the initial score of data items and performs breeding, evaluation of fitness and finally filtering to produce a new generation. Sarasamma et al. (2005) integrated Self Organizing Map with consistency based feature selection for identifying the attacks in the network. Kuang et al. (2014) proposed a novel support vector machine combining kernel principal component analysis (KPCA) with genetic algorithm. A multi layer SVM classifier was adopted to determine whether an action results in an attack. An improved kernel function was proposed by embedding the mean and the difference of mean square values of attributes. Genetic algorithm optimized the punishment factor C, kernel parameter o and tube size ϵ of SVM. This model resulted in higher accuracy, faster speed and good generalization capability. Sumaiya Thaseen and Aswani Kumar (accepted for publication) proposed a novel model for intrusion detection by integrating PCA and support vector machine (SVM) after optimizing the kernel parameter using variance of samples belonging to same and different class. This variance plays a major role in identifying the optimal kernel parameter to be deployed in the model to be trained. Hence this method resulted in a better classification accuracy.

Hence from the literature it is very clear that classifiers along with dimensionality reduction techniques results in good accuracy by improving the classification rate and a shorter detection time. The kernel parameter of SVM also plays an important role in increasing the accuracy. Therefore in this paper we propose a model to reduce the dimensionality and improve the classification rate by combining chi square feature selection technique and optimized kernel SVM. We also analyze the computational time required for training and testing the proposed model.

3. Background

In this section we briefly analyze the feature selection and data mining techniques that are employed in our proposed model.

3.1. Scaling

Network traffic is very huge and contains many features with a different range of values. Processing the data directly is time consuming and classification may not be accurate. Hence data packets undergo a normalization process before dimensionality reduction. Many methods are available for normalization. The commonly used are z-score, min–max normalization and decimal scaling. The z-score technique is chosen for the proposed model as it is the simplest normalization technique. This method preserves the range (maximum and minimum) and introduce dispersion of series (standard deviation / variance). The z-score linearly transforms the data in such a way, that the mean value of the transformed data equals 0 while their standard deviation equals 1. The transformed values themselves do not lie in a particular interval like [0,1] or so. The transformation formula thus is:

$$x^1 = \frac{(x - \bar{E})}{s} \tag{1}$$

where x is the current sample, x^1 is the transformed sample, $\bar{E}$ denotes the mean of the data and 's' represents the standard deviation.

3.2. Feature selection

Feature selection and ranking are very crucial for intrusion detection. Feature selection is the process of obtaining the score for each potential feature and then obtaining the excellent 'k' features. Scoring is done by counting the frequency of a feature in training positive and negative class samples separately and then obtaining a function of both. There are many features that have to be monitored for intrusion detection out of which certain features will be useful and others may be useless. The removal of useless features enhances the accuracy and decreases the computation time thereby achieving higher performance.

The commonly known metrics are chi-squared (CHI), Information Gain, Correlation Coefficient and Odds Ratio (OR). Yang and Pedersen (1997) reported that CHI performed best for multi class data. Hence chi-square feature selection metric is used in our model.

3.2.1. Chi-square feature selection (chi)

Chi-squared is a numerical test that measures deviation from the expected distribution considering the feature event is independent of the class value. The chi square value is calculated from the following metrics such as true positives (tp), false positives (fp), true negatives (tn), false negatives (fn), probability of number of positive cases P_{pos} and probability of number of negative cases P_{neg}.

$$chi-square\_metric = t(t_p,(t_p+f_p)P_{pos}) + t(f_n,(f_n + t_n)P_{pos}) + t(f_p,(t_p+f_p)P_{neg}) + t(t_n,(f_n+t_n)P_{neg}) \quad (2)$$

where t (count, expect) = (count – expect)^2/expect.

The chi-square approach consists of the following steps:

(i) Specify the hypothesis
(ii) Devise an analysis plan
(iii) Examine sample data
(iv) Deduce results.

3.2.1.1. Devise an analysis plan. After the hypothesis is stated, the analysis plan specifies how to utilize model data to accept or reject the hypothesis. The plan must specify the following:

(i) Significance rank: Researchers choose significance level equal to 0.01,0.05 or 0.10 but it can be any value between 0 and 1.
(ii) Test method: The chi-square test is used to test independence level to identify whether there is a considerable relationship between two categorical attributes.

3.2.1.2. Examine sample data. The sample data have to be analyzed to calculate the degrees of freedom, predictable frequencies, test value and the *P*-value associated with the test.

(i) Degreesof freedom : $DF = (r-1)*(c-1)$ (3)

where r is the number of levels of one categorical variable and c is the number of levels for other categorical variable.

(ii) Test Statistic :

$$\chi^2(f,c) = \left[\frac{N*(AD-CB)^2}{(A+C)(B+D)(A+B)(C+D)}\right] \quad (4)$$

where A = No. of times feature 't' and class label 'c' co-occurs.

B = No. of times 't' appears without 'c'
C = No.of times 'c' appears without 't'.
D = No.of times neither 'c' nor 't' appears.
N = Total number of records.

3.2.2. Ranking methodology

At every time instance, one input feature is removed from the sample and the resulting sample is then used for training and testing of the model. The important features are ranked according to a set of rules based on performance. The procedure is specified as follows:

(i) Delete one input attribute from the data (training and testing).
(ii) The resultant data are used for training and testing the classifier.
(iii) The results of the classifier are analyzed using the performance metrics.
(iv) The rules are used to rank the attribute by its importance level.
(v) Repeat the steps 1 to 4 for each of the attributes.

3.3. Support vector machine classification model

Supervised machine learning solves the problem of assigning labels to records where the labels are assigned from a finite element set. This technique is called as multi class learning. Numerous algorithms have been developed for multi class learning constructed upon classification algorithms for binary problems. Many multi class learning algorithms such as decision tree, specialized versions of boosting such as AdaBoost and support vector machines have been used. One of the dominating approaches for the problem of multi class learning is support vector machine wherein a single multi class problem is modified into multiple binary problems. A SVM is a binary classifier, that is, the class labels contain only two values + 1 and -1. Many real world problems have to be assigned in multiple classes. Hence we employ a multi class SVM.

3.3.1. Multi class SVM model

Construct a set of binary classifiers $f^1, f^2 \ldots f^N$ for $1 \ldots N$ classes each trained to differentiate one class from the rest. A multi class categorization can be obtained by combining them according to the maximal output before applying the sgn function.where

$$\text{argmax } g^k(x)$$
$$\text{where } g^k(x) = \sum_{i=1}^{n} y_i \alpha_i^k k(x,x_i) + b^k \quad (5)$$
$$\text{where } k = 1 \ldots N$$

wherein $g^k(x)$ returns a signed real value which is the distance from the hyper plane to the point x. This value is referred as the confidence value. The higher the value, the more confident we are that the point x belongs to positive class. Hence we need to assign x to the class having highest confidence value.

Given normal data $\chi = \{x_1, x_2 \ldots x_m\} \in R^d$ and let r be the radius of the hypersphere and $c \in R^d$ which is the center. The optimization problem can be solved by determining the minimum enclosing hypersphere.

$$\text{Minimize } r^2$$
$$\text{Subject to } ||\Phi(x_j) - c||^2 \leqslant r^2, j = 1, \ldots m \quad (6)$$

$$L(c,r,\alpha) = r^2 + \sum_{j=1}^{m} \alpha_j \{||\Phi(x_i) - c||^2 - r^2\} \quad (7)$$

$$\text{Setting the derivatives } \frac{\partial L(c,r,\alpha)}{\partial c} = 2\sum_{j=1}^{n} \alpha_j (\Phi(x_j) - c) = 0 \quad (8)$$

We can obtain the following equation,

$$\sum_{j=1}^{m} \alpha_j = 1 \text{ and } c = \sum_{j=1}^{m} \alpha_j \Phi(x_j)$$

Hence the Eq. (7) becomes,

$$L(c,\gamma,\alpha) = \sum_{j=1}^{m} \alpha_j k(x_j,x_j) - \sum_{i,j=1}^{m} \alpha_i \alpha_j k(x_i,x_j) \quad (9)$$

which is the dual form of Eq. (7).

The dual form of α can be obtained by solving the optimization problem,

Maximizing,

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i k(x_i, x_i) - \sum_{i,j=1}^{m} \alpha_i \alpha_j k(x_i, x_j) \quad (10)$$

Subject to

$$\sum_{i=1}^{m} \alpha_i = 1 \text{ and } \alpha_i \geqslant 0, i = 1 tom.$$

It should be noted that lagrange multiplier can be non-zero only if the inequality constraint is an equality for the solution.

The complementarity conditions are satisfied by the optimal solutions for $\alpha, (c, \Upsilon)$ given by,

$$\alpha_i \{||\Phi(x_i) - c||^2 - r^2\}, \quad i = 1 \ldots m \quad (11)$$

Hence it implies that the training samples x_i lie on the surface of the optimal hypersphere corresponding to $\alpha_i > 0$.

The decision function becomes,

$$f(x) = \text{sgn } (r^2 - ||\Phi(x) - c||^2)$$

This implies,

$$= \text{sgn}(r^2 - \{\Phi(x).\Phi(x)) - 2\sum_{i=1}^{m} \alpha_i \Phi(x).\Phi(x_i)) + \sum_{i,j=1}^{m} \alpha_i \alpha_j (\Phi(x_i).\Phi(x_j))\})$$

$$= \text{sgn}(r^2 - \{k(x, x) - 2\sum_{i=1}^{m} \alpha_i k(x, x_i) + \sum_{i,j=1}^{m} \alpha_i \alpha_j k(x_i, x_j)\}) \quad (12)$$

Thus the aim of obtaining minimum enclosing hypersphere containing all training samples is satisfied.

3.3.1.2 One-versus-all SVM. This technique is one of the simple multiclass classifiers frequently used in SVMs which has the following properties:

(i) Solve M different binary problems: classify "class k" versus "the rest classes" for $k = 1 \ldots M$
(ii) Assign a test model to the class which is having largest $f_k(x)$ (most positive value), where $f_k(x)$ is the kth problem.

This approach is very simple to implement and it performs well in practice. Hence in this paper we have followed this approach.

4. Proposed work

In this section we propose a hybrid model for intrusion identification using chi-square feature selection and multi class SVM.

4.1. Proposed methodology

The proposed model integrates rank based chi-square feature selection with multi class SVM optimized by kernel scale. Fig. 1 shows the block diagram of the proposed model. Normalization is performed as the initial preprocessing step followed by feature selection using chi-square based feature selection. The proposed model employs two stages: In the first stage, chi square feature selection finds an optimal subset of all attributes and removes low rank attributes. The ranking plays a major role in identifying the high priority attributes that are crucial for classification. In the second stage, the data are divided into validation, training and test set. The validation set is used to obtain the optimized kernel parameter (gamma) and overfitting constant 'C' which is explained in Section 4.2. The parameters that result in best cross validation accuracy are retrieved as optimal parameters. The optimal parameters are then fed to the SVM classifier to train the model for the training set. The trained model is used to predict the label for the test data set. Algorithm 1 shows the step by step analysis of feature selection performed by chi-square ranking and integration with multilevel SVM. In the next sub section we discuss the methodology for tuning the parameters gamma

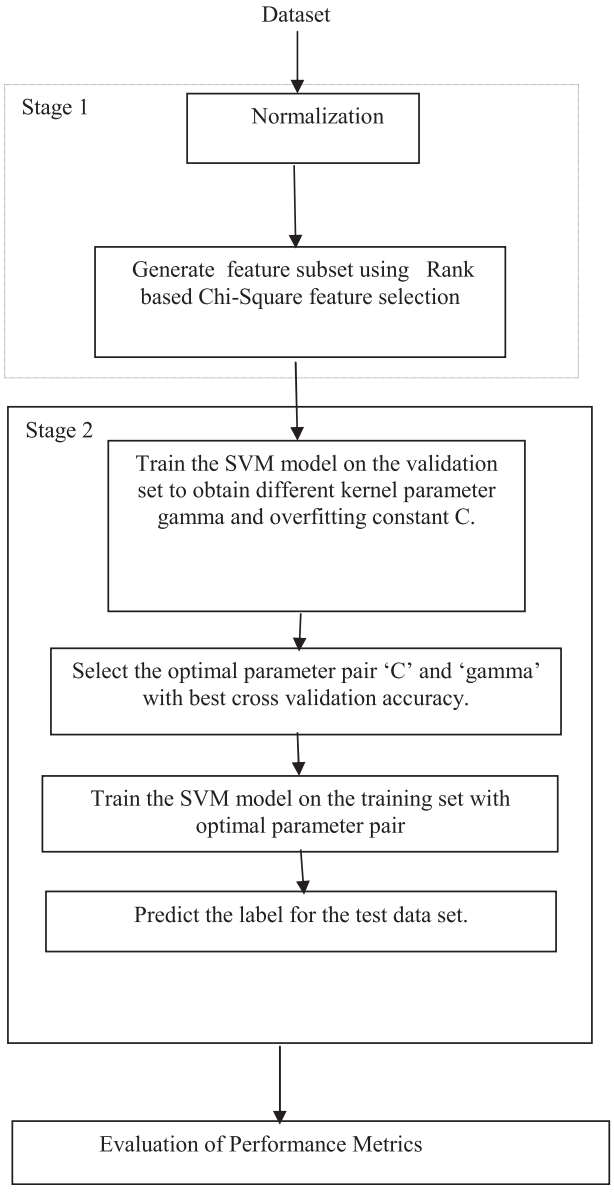

Figure 1 Proposed intrusion detection model using multi class SVM.

and overfitting constant C and predicting the class label using multi class SVM.

4.2. Optimization of Radial Basis Function (RBF) kernel parameter γ and overfitting constant 'C'

A model is usually validated using '*k*' folds of the training data where k is the number of partitions. During cross validation, the SVM model parameters 'C' and gamma are trained on the remaining part of the data considered as a test set to compute the performance measure. This approach is called cross validation.

Algorithm 1. Chi-square Multi Level SVM optimized by variance tuning

Input:
Tc = Training Data of all features of NSL-KDD data set.
C = class labels of training data.
Algorithm
Initialize S = {F1...Fn}
For each feature {f} in the training set, Compute chi-square metric using Eq. (5)
If (($\chi^2 <$ *threshold*)
S = S – {f}
Else
Continue;
End For
Take training data Tc with reduced feature set and randomly split into training set $T_{trn} = \{(x_1^t, y_1^t) \ldots (x_n^t, y_n^t)\}$ validation set $T_{Val} = \{(x_1^v, y_1^v) \ldots (x_n^v, y_n^v)\}$ and test set. $T_{test} = \{(x_1^t, y_1^t) \ldots (x_n^t, y_n^t)\}$
For every validation data T_{val}
  For every f in S
Determine σ^2 of every feature and substitute in kernel using Eq. (13)
Determine C with various margins 's' using Eq. (14)
Train the SVM model with different C and gamma
  End For
End For
Obtain the (C, gamma) with best accuracy
For every training data Ttrn
  Train the SVM model with the optimized model parameters.
End For
For every test data T_{test}
Predict the label y_{crit} for each sample
End For 's'
Display confusion matrix of test data.

The RBF Kernel is obtained as follows

$$k(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2) \quad (13)$$

where $||x_i - x_j||^2$ is the squared Euclidean measure of distance between two feature vectors and gamma is represented by $\gamma = \frac{1}{2\sigma^2}$ and σ^2 is the variance associated with each attribute in the validation data set. This variance is optimized using cross validation.

The objective function of SVM is

$$\min ||w||^2 + C \sum \xi \quad (14)$$

where 'w' is the margin of hyperplanes, ξ is the error rate due to slack variables and C is the overfitting constant. If '*C*' is very large, optimization algorithm will reduce ||w|| leading to generalization loss. If '*C*' is small, it leads to a large training error. Hence it is very crucial to identify optimal value of '*C*'.

The cross validation accuracy before optimization is 95% with a C value of 1 and gamma value of 0.01 obtained after feature selection whereas the optimization accuracy increases to 99% when the guassian value increases to 0.07. This result is achieved nearly after 31 iterations of modifying the kernel parameter value for all data in the validation set.

A larger value of σ will result in a smooth decision surface and a systematic decision boundary. Hence our optimization using variance tuning will determine an optimal σ that will result in a better accuracy. The kernel parameter and the overfitting constant C obtained after the cross validation is given to train the SVM model and finally predict the label of the test data set as shown in Fig. 2.

5. Implementation and results

The experiments were conducted on MATLAB R2012A integrated with libSVM package which supports support vector classification (C-SVC, mu-SVC), regression (epsilon SVR, nu-SVR) and distribution estimation (one-class SVM). It also supports multi class classification. Experiments were performed on NSL-KDD (Nsl) dataset. The data sets contain five categories of network traffic namely normal, denial of service (DoS), unauthorized access to local supervisor privileges (User to Root,U2R), Remote to Local (R2L) and probe. A description of NSL- KDD data set and its attacks can be obtained

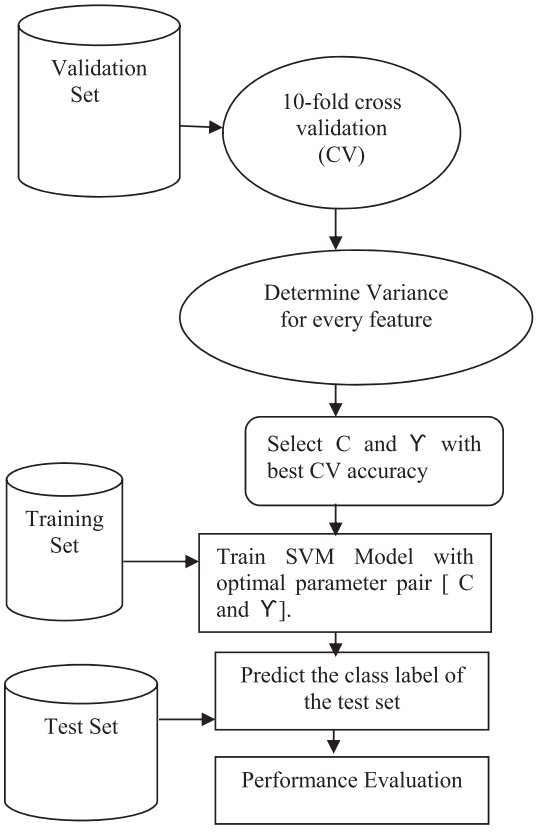

Figure 2 Classification using multi class SVM with parameter tuning technique.

from Nsl. The entire dataset is taken for our analysis containing 33,300 records. Nearly 8325 records are used as training data and 24,975 records are used as testing set. The sample size is proportionately taken so as to have a small training set and train all samples of network traffic and then test a large number of samples. Z-score normalization is performed before the start of the experiments by determining the frequency of the values and converting into numerical attributes and thereby transforming all attributes into the normalized format.

The non-numerical attributes are transformed into numeric by discretization. This is done by grouping categorically to an appropriate integer. Grouping is the process of recoding into a specified number of categories or recoding by interval. The three attributes present in the NSL-KDD data set that are discretized are protocol_type, service and flag. The protocol type such as tcp, udp is transformed into 1 and 2. The service at the destination such as http, telnet is transformed into 1 and 2. Flag values are also transformed into respective numerical category. The correlation between the attributes can influence the classification result. Eliminating crucial features accidentally can reduce the classification result. Hence the 41 attributes of network traffic are carefully examined and 31 attributes are obtained as optimal subset which is shown in Table 1. Table 1 shows the attributes selected by Ranker based chi-square feature selection technique. Only 31 attributes are selected out of the complete 41 attributes based on the ranking search method.

Table 1 shows the attributes selected by Ranker based chi-square feature selection technique. Only 31 attributes are selected out of the complete 41 attributes based on the ranking search method. The reason for selecting the chi-square based feature selection is that it selects a combination of continuous

Table 1 Attributes selected by chi-squared technique.

Rank	Attribute	Attribute number in the KDD set	Description
1	Service	3	Different types of services provided such as http,ftp, smtp, telnet and other
2	Dst_bytes	6	The number of bytes accepted in one connection
3	Dst_host_diff_srv_rate	35	Percentage of connections that exist for different services among connections in dst_host_count
4	Diff_srv_rate	30	Percentage of connections that exist for different services among connections in count
5	Flag	4	Connection status.Possible status are SF,S0,S1,S2,S3,OTH,REJ,RSTO, RSTOSO,SH,RSTRH,SHR
6	Dst_host_serror_rate	38	Percentage of connections that activated the flag s0,s1,s2 or s3 among connections in dst_host_count
7	Dst_host_srv_count	33	Total connections to specific destination port
8	Same_srv_rate	29	Percentage of connections that were exist for the same service among connections in count
9	Count	23	Sum of connections to specific destination
10	Dst_host_same_srv_rate	34	Percentage of connections to the same service
11	Dst_host_srv_serror_rate	39	Percentage of connections that activated the flag.
12	Serror_rate	25	Percentage of connections that activated the flag s0,s1,s2 and s3 among connections in count
13	Src_bytes	5	The number of bytes sent in one connection
14	Dst_host_srv_diff_host_rate	37	Percentage of connections to various destinations
15	Srv_serror_rate	26	Percentage of connections that activated the flag s0,s1,s2 and s3 among connections in srv_count
16	Dst_host_same_src_port_rate	36	Percentage of connections to the same source port
17	Logged_in	12	If login value is correct then assign 1 else 0
18	Dst_host_Count	32	Total connections to specific IP
19	Hot	10	Total number of hot connections
20	Dst_host_rerror_rate	40	Percentage of connections that activated the flag among connections in dst_host_count
21	Srv_count	24	Sum of connections to same destination port
22	Duration	1	Duration of connection
23	Srv_diff_host_rate	31	Percentage of connections that exist to different destinations among connections in srv_count
24	Dst_host_srv_rerror_rate	41	Percentage of connections that activated the flag among connections in dst_host_srv_count
25	R error_rate	27	Percentage of connections that activated the flag REJ among connections in count
26	Protocol_type	2	
27	Srv_r error_rate	28	Percentage of connections that activated the flag among connections in srv_count
28	Is_guest_login	22	If the user is logged in as a guest or visitor
29	Srv_count	24	Sum of connections to a specific destination port
30	Num_compromised	13	Sum of times "not found" fault obtained in a connection
31	Num_failed_logins	11	Total number of incorrect logins in a specific connection

and discrete features whereas dimensionality reduction methods select only continuous features discarding the discrete features.

Hence in this paper we have deployed a ranking based feature selection technique that selects a combination of continuous and discrete features.

Table 2 shows the performance metrics such as kappa statistic, mean absolute error and root mean square error. Kappa statistic is a measure of classification in categorical data. A kappa coefficient of 1 means a perfect statistical model whereas a 0 represents every model value is different from the actual value. The higher the value, the more statistic correlation between attributes. Mean absolute error (MAE) is the average of difference between the predicted and actual values. Root mean square error (RMSE) is the average of squared difference between every computed value and its corresponding correct value. MAE and RMSE should be close to 0 as minimum error rate results in better accuracy.

Table 3 shows the performance metrics of each class such as DoS, Probe, U2R, R2L and normal. The false positive rate is very less and the true positive rate is very high which is an important criteria to be achieved for any intrusion detection model. Precision, recall and f-measure are the other performance metrics analyzed. Precision also called as positive predictive value is the fraction of retrieved instances that are relevant. Recall also known as sensitivity is the fraction of relevant instances that are retrieved. F-measure is a common evaluation metric that combines precision and recall. All the three derived metrics should be close to 1 for a good model. Table 4 shows the confusion matrix of each class obtained after feature selection. The rows in the matrix represent true values; columns represent predicted values and entries along the diagonal specify correct predictions (see Table 5).

5.1. Performance analysis

The performance of our model is measured using the following metrics. These values are true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) where TP specifies the normal behavior that is correctly predicted, FP denotes the normal behavior wrongly assumed as abnormal, TN indicates the normal performance that is identified as correct and FN specifies the abnormal performance that is misdetected as normal.

$$\text{(i) Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{15}$$

$$\text{(ii) False Alarm rate(FAR)} = \frac{FN}{TN + FP} \tag{16}$$

Table 2 Performance metrics of the model after ranker + chi-square attribute evaluation.

Correctly classified instances	35,360 (95.492%)
Incorrectly classified instances	1682 (4.5408%)
Kappa statistic	0.9351
Mean absolute error	0.0243
Root mean squared error	0.1102

Table 3 Detailed accuracy by class obtained after feature selection.

Class	TP rate	FP rate	Precision	Recall	F-measure	ROC area
Normal	0.996	0.004	0.995	0.996	0.996	0.999
DoS	0.999	0.001	0.998	0.999	0.998	1
R2L	0.987	0.001	0.992	0.987	0.99	0.999
Probe	0.992	0.001	0.992	0.992	0.992	1
U2R	0.739	0	0.895	0.739	0.81	0.968

These parameters play a crucial role in evaluating the performance of the intrusion detection model. Table 6 shows the accuracy and false positive rate obtained for the selected 31 features with different C and gamma parameters after a series of iterations. The best accuracy results are alone depicted in the table whereas the model was tested on various range values of C:[1,10,100,1000] and gamma[0.01,0.03,0.05,0.07,0.09,0.001,0.0001]. It is evident that high accuracy is obtained when the C value is 1 and gamma value is 0.07 which is indeed a coarse range identified. Hence as the scale increases, accuracy increases and false positive rate decreases.

Fig. 3 shows the accuracy comparison of the proposed model with traditional binary SVM techniques using dimensionality reduction techniques. It is evident from the graph that the proposed model results in higher accuracy and is superior to binary class SVM techniques where the parameters are randomly selected. Feature selection by PCA in existing techniques can result in discriminatory information that may hinder the improvement of classification performance. Hence feature selection by chi-square in the proposed model aims to improve the training, testing time and generalization performance of the classifier. Single SVM has to perform more cross judging and hence results in increased training time. Though N-KPCA-GA-SVM model is better than other three methods KPCA-GA-SVM, PCA-GA-SVM and single-SVM with respect to training time and optimization of the network parameters, the accuracy rate is much improved in the proposed model. Moreover the proposed model does not cause large fluctuations in the detection performance.

Table 4 Confusion matrix obtained after feature selection.

Probe	DoS	U2R	R2L	Normal
2375	13	0	0	165
3	8845	0	0	90
0	0	10	3	22
0	0	0	1619	186
101	88	3	58	11,394

Table 5 Accuracy obtained by parameter tuning technique.

C	Gamma	Mean accuracy (%)	False Alarm rate (%)
1	0.001	88	12
1	0.01	94	6
1	0.03	96	4
1	0.05	96.3	3.7
1	0.07	98.1	1.9
10	0.001	97.7	2.3
10	0.01	96.3	3.7

Table 6 Confusion matrix obtained after multi class SVM.

Normal	DoS	R2L	Probe	U2R
15,543	19	14	23	2
13	11,843	0	3	0
27	1	2377	1	2
21	3	3	3359	0
10	0	2	0	34

Fig. 4 shows the individual class accuracy of the proposed model in comparison with other techniques used for intrusion detection. To detect the normal, probe and DoS classes, the CANN approach performs slightly better. However K-NN can identify some U2R and R2L cases but CANN cannot. These results show that the CANN approach with 19- dimensional dataset does not detect U2R and R2L attacks as efficiently as K-NN. It is also to be noted that a drastic reduction in dimensionality can also result in failure of minority attack identification. Therefore in comparison with other techniques the individual class accuracy for minority attack is greater in our model which depicts that our aim is to focus on all attack categories rather than restricting to individual attacks.

Fig. 5 shows the false alarm rate (in %) comparison of the proposed model with traditional binary SVM techniques and the graph shows the false alarm rate of the proposed model is very less. One versus all (OVA) SVM has the lowest error rate (Madzarov, 2008) in comparison with other techniques such as one-against-one (OVO), Binary Tree of SVM (BTS) and Directed acyclic graph SVM(DAGSVM) based on the experiments conducted in various data sets of UCI repository (Blake et al., 1998). Though the OVO approach has little higher training and testing time it is still preferable as the error rate is very less in comparison with other traditional binary SVMs.

Table 7 shows the results obtained by comparing the run time of the different dimensional data sets used for classification. The data preprocessing time includes the data loading time. As we can observe, a longer run time is needed for the dataset containing high dimensions. Thus the entire data set without feature selection requires 4.09 h whereas a 31 dimensional dataset requires only half of the training and testing time. Comparing the run time with the most recent related works, (ie. testing times) Lin et al. (2015), Kim et al. (2014) and Nadiammai and Hemalatha (2014) obtained times of 13, 11.2 and 8 s which had the complete KDD data set containing about 4,00,000 data packets whereas our experiments deal with 33,300 records only which is an improvised version of KDD data set. The findings suggest that a significant amount of time is reduced for training and testing when dimensionality is reduced and hence this technique proves to be better than other existing techniques with respect to accuracy, false alarm rate and run time analysis.

Applications with large number of data sets find multi class SVM's computationally more expensive and hence deal with binary SVM's (Hsu and Lin, 2002). But in our model we are

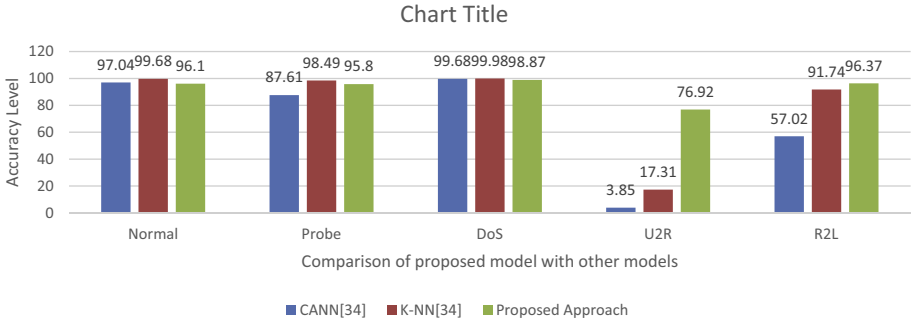

Figure 3 Performance comparison of proposed model with other intrusion models over 5 classes.

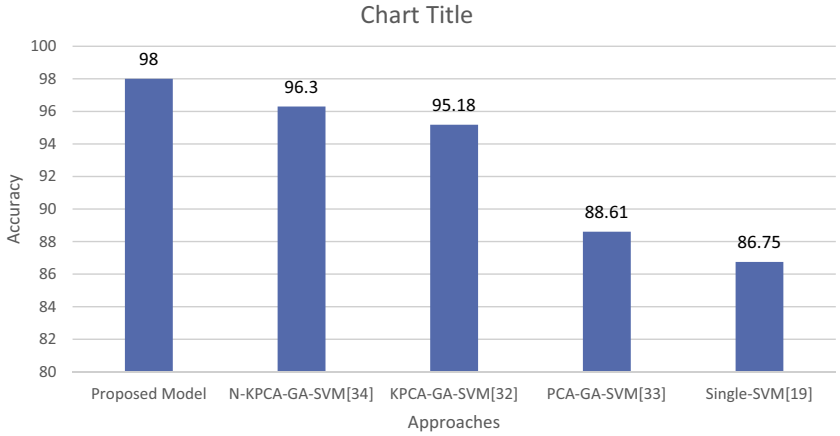

Figure 4 Comparison of proposed model based on accuracy.

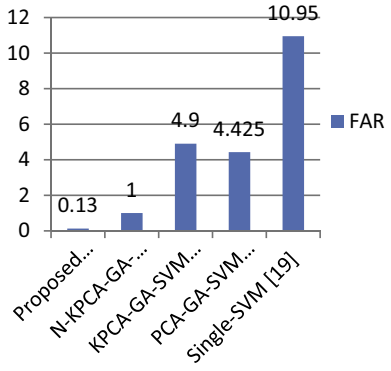

Figure 5 Performance comparison of proposed model based on false alarm rate.

Table 7 Run time of different dimensional datasets.

	Data preparation	Training and testing
41 Dimensional data set applied to SVM with parameter tuning	600 s (10 min)	4.09 h (14,729 s)
31 Dimensional data obtained through chi-square feature selection applied to SVM with parameter tuning.	420 s (7 min)	2.84 h (10,235 s)

dealing with one-against-all SVM which is one of the earliest SVM multi class classification available. In this model the multi class problem is solved by decomposing into several binary classes which results in better accuracy.

5.2. Discussions

The proposed intrusion detection model is an integration of chi square feature selection and multi class support vector machine optimized by parameter tuning technique. This approach is different from the traditional approaches as the curse of dimensionality is high in large data sets. Hence an integration of feature selection and classification results in a better classification accuracy of the individual attacks in comparison to other approaches discussed above. SVM is one of the generalized learning algorithms and many variants of SVM such as One-versus-All and One-versus-One are applicable to this domain. The reason for selecting One-versus-All SVM is because the unknown pattern is determined according to the maximum result obtained from all SVMs which results in a negligible error rate. The SVM model parameters are tuned by the parameter tuning technique discussed in Section 4.2 which is an additional optimization task performed to yield a better prediction. The training of the classifier with optimized parameters assures the prediction label is accurate for the test set. The novelty of this approach is this kind of optimization tuning has not been performed on a multi class SVM classifier for improving the accuracy of attack detection rate in network traffic.

6. Conclusion

This paper proposes an intrusion detection model using chi-square feature selection and multi class support vector machine. A parameter tuning technique is adopted for optimization of RBF kernel parameter gamma and overfitting constant 'C'. The reason for employing multi class SVM is that it has not been used for intrusion detection and the accuracy of individual attack types have not been analyzed in detail. The other advantage is multi class SVM reduces training and testing time. The investigational results on NSL-KDD dataset which is an enhanced version of KDDCup 1999 dataset shows that our proposed model results in high detection rate and low false alarm rates in comparison to other traditional approaches.

For future enhancements, we may develop some algorithms combining kernel methods with other classification methods for pattern analysis and optimization techniques for SVM parameter optimization.

References

Amiri, Fatemeh, Mahdi, Mohammad, Yousefi, Rezaei, 2011. Mutual information based feature selection for intrusion detection systems. J. Network Comput. Appl. 34, 1184–1199.

Blake, C., Keogh, E., Merz, C., 1998. UCI repository of machine learning databases. Statlog Data Set, <http://www.ics.uci.edu/mlearn/MLRepository.html[online]>.

Chebrolu, S., Abraham, A., Thomas, P., 2005. Feature deduction and ensemble design of intrusion detection systems. Comput. Soc. 24 (4), 295–307.

Farrahi, Vahid S., Ahmadzadeh, Marzieh, 2015. KCMC: a hybrid learning approach for network intrusion detection using K-means clustering and multiple classifiers. Int. J. Comput. Appl. 124 (9), pp. 18–23. Published by Foundation of Computer Science (FCS), NY, USA.

Fisch, D., Hofmann, A., Sick, B., 2010. On the versatility of radial basis function neural networks: a case study in the field of intrusion detection. Inf. Sci. 180, 2421–2439.

Horng, S.-J., Su, M.-Y., Chen, Y.-H., Kao, T.-W., Chen, R.-J., Lai, J.-L., Perkasa, C.D., 2010. A novel intrusion detection system using support vector machines. Expert Syst. Appl.

Hsu, Chih-Wei, Lin, Chih-Jen, 2002. A comparison of methods for multi-class support vector machines. IEEE Trans. Neural Networks, 415–425.

Ilgun, K., Kemmerer, R.A., Porras, P.A., 1995. State transition analysis: a rule-based intrusion detection approach. IEEE Trans. Software Eng. 21 (3), 181–199.

Kasliwal, Bhavesh, Bhatia, Shraey, Saini, Shubham, Sumaiya Thaseen, I., Aswani Kumar, Ch., 2014. A hybrid anomaly detection model using G-LDA. In: 2014 IEEE International Advance Computing Conference, pp. 288–293.

Khan, L., Awad, M., Thuraisingham, B., 2007. A new intrusion detection system using support vector machines and hierarchial clustering. J. Very Large Databases 16 (4), 507–521.

Kim, G., Lee, S., Kim, S., 2014. A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. Expert Syst. Appl. 41 (4), 1690–1700.

Kuang, Fangjun, Xu, Weihong, Zhang, Siyang, 2014. A novel hybrid KPCA and SVM with GA model for intrusion detection. In: Appl. Soft Comput.

Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A., Srivastava, J., 2003. A comparative study of anomaly detection schemes in network intrusion detection, Proceedings of the third SIAM Conference on Data Mining.

Lee, W., Stolfo, S., Mok, K., 1999. A data mining framework for building intrusion detection model. In: Proc. IEEE Symposium on Security and Privacy, pp. 120–132.

Lee, J.H., Lee, J.H., Sohn, S.G., 2008. Effective value of decision tree with KDD 99 intrusion detection datasets for intrusion detection

system, 10th International Conference on Advanced Communication Technology (ICACT'08), pp. 1170–1175.

Lin, W.-C., Ke, Shih-Wen, Tsai, Chih, 2015. CANN: an intrusion detection system based on combining cluster centers and nearest neighbors. Knowledge-Based Syst. 78, 13–21.

Madzarov, Gjorgji, Gjorgjevikj, Dejan, Chorbev, Ivan, 2008. Multi class classification using support vector machines in binary tree architecture. In: International Scientific Conference.

Mukkamala, S., 2005. Sung, intelligent paradigms. J. Network Comput. Appl. 28 (2), 167–182.

Nadiammai, G.V., Hemalatha, M., 2014. Effective approach toward intrusion detection system using data mining techniques. Egypt Inf. J. 15 (1), 37–50.

Novikov, D., Yampolskiy, R.V., Reznik, L., 2006. Anomaly detection based intrusion detection. In: Proceedings of the Third International Conference on Information Technology: New Generations (Itng'06).

<http://nsl.cs.unb.ca/NSL-KDD/>.

Panda, Mrutyunjaya, Abraham, Ajith, Patra, Manas Ranjan, 2011. A hybrid intelligent approach for network intrusion detection. In: International Conference on Communication Technology and System Design, pp. 1–9.

Peddabachigari, S., Abraham, A., Grosan, C., Thomas, J., 2007. Modelling intrusion detection system using hybrid intelligent systems. J. Comput. Network Appl. 30 (1), 114–132.

Saniee Abadeh, M., Habibi, J., Lucas, C., 2007. Intrusion detection using a fuzzy genetics based learning algorithm. J. Network Comput. Appl. 30 (1), 414–428.

Sarasamma, S.T., Zhu, Q.A., Huff, J., 2005. Hierarchical Kohonen net for anomaly detection in network security. IEEE Trans. Syst. Man Cybernetics –Part B Cybernetics 35, 302–312.

Saxena, Harshit, Richariya, Vineet, 2014. Intrusion detection in KDD99 dataset using SVM-PSO and feature reduction with information gain. Int. J. Comput. Appl. 98 (6), 25–29.

Senthilnayaki, Balakrishnan, Venkatalakshmi, K., Kannan, A., 2014. Intrusion detection system using feature selection and classification technique. Int. J. Comput. Sci. Appl. 3 (4), 145–151. http://dx.doi.org/10.14355/ijcsa.2014.0304.02.

Shafi, K., Abbass, H.A., 2009. An adaptive genetic based signature learning system intrusion detection. Expert Syst. Appl. 36 (10), 12036–12043.

Sumaiya Thaseen, I., Aswani Kumar, Ch., 2014. Intrusion detection model using fusion of PCA and optimized SVM. In: Proceedings of 2014 International Conference on Computing and Informatics (IC3I) held on 27–29 Mysore, India, pp. 879–884.

Sumaiya Thaseen, I., Aswani Kumar, Ch., accepted for publication. Improving accuracy of intrusion detection model using PCA and optimized SVM. CIT J. Comput. Inf. Technol.

Sung, A., Mukkamala, S., 2003. Identifying important features for intrusion detection using support vector machines and neural networks. In: Proceedings of the International Symposium on Applications and the Internet (SAINT 2003), pp. 209–217.

Toosi, A.N., Kahani, M., 2007. A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers. Comput. Commun. 30, 2201–2212.

Wang, G., Hao, J.X., Ma, J., Huang, L.H., 2010. A new approach to intrusion detection using artificial neural networks and fuzzy clustering. Expert Syst. Appl. 37, 6225–6232.

Xuren, W., Famei, H., Rongsheng, X., 2006. Modeling intrusion detection system by discovering association rule in rough set theory framework. In: Proceedings of the International Conference on Computational Intelligence for Modeling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIM CA- IAWTIC06).

Yang, Yiming, Pedersen, Jan O., 1997. A comparative study on feature selection in text categorization. In: Proceedings of the 14th International Conference on Machine Learning (ICML), pp. 412–420.

Ye, N., Borror, Y.Z.C.M., 2004. Robustness of the Markov chain model for cyber attack detection. IEEE Trans. Reliab. 116, 12353.

Zaman, 2009. Lightweight IDS based on feature selection and IDS classification scheme, Proceedings of 2009 International Conference on Computational Science and Engineering, pp. 365–370.

Zhang, Z., Shen, H., 2005. Application of online-training SVMs for real-time intrusion detection with different considerations. Comput. Commun. 28 (12), 1428–1442.