Expert Systems With Applications 67 (2017) 296–303

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system

Wathiq Laftah Al-Yaseen[a,b,*], Zulaiha Ali Othman[a], Mohd Zakree Ahmad Nazri[a]

[a] Data Mining and Optimization Research Group (DMO), Centre for Artificial, Intelligence Technology (CAIT), School of Computer Science, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), 43600 Bandar Baru Bangi, Malaysia
[b] Al-Furat Al-Awsat Technical University, Iraq

ARTICLE INFO

Article history:
Received 30 August 2015
Revised 29 September 2016
Accepted 30 September 2016
Available online 30 September 2016

Keywords:
Intrusion detection system
Support vector machine
Extreme learning machine
K-means
Multi-level
KDD Cup 1999

ABSTRACT

Intrusion detection has become essential to network security because of the increasing connectivity between computers. Several intrusion detection systems have been developed to protect networks using different statistical methods and machine learning techniques. This study aims to design a model that deals with real intrusion detection problems in data analysis and classify network data into normal and abnormal behaviors. This study proposes a multi-level hybrid intrusion detection model that uses support vector machine and extreme learning machine to improve the efficiency of detecting known and unknown attacks. A modified K-means algorithm is also proposed to build a high-quality training dataset that contributes significantly to improving the performance of classifiers. The modified K-means is used to build new small training datasets representing the entire original training dataset, significantly reduce the training time of classifiers, and improve the performance of intrusion detection system. The popular KDD Cup 1999 dataset is used to evaluate the proposed model. Compared with other methods based on the same dataset, the proposed model shows high efficiency in attack detection, and its accuracy (95.75%) is the best performance thus far.

1. Introduction

Dependence on convenient security systems to protect computers and networks against intrusions is a crucial issue in computer science because of the significant development of network-based computer services. Intrusion detection system (IDS) is one of the systems used to monitor and analyze the events in a computer or network to identify any deviation from normal behavior. IDSs can be categorized in several ways, but the most common are misuse-based and anomaly-based categories (Lee, Stolfo, & Mok, 1999). Misuse-based IDS can efficiently detect known attacks, such as Snort (Roesch, 1999). This type of IDS has a low false alarm rate, but it fails to identify new attacks that do not embody any rules in the database. Anomaly-based IDS builds a model of normal behavior and then distinguishes any significant deviations from this model as intrusions. This type of IDS can detect new or unknown attacks but features a high false alarm rate.

To reduce the false alarm rate of anomaly-based IDS, many machine learning techniques, including support vector machine (SVM) (Feng, Zhang, Hu, & Huang, 2014; Horng et al., 2011; Kuang, Xu, & Zhang, 2014) and extreme learning machine (ELM) (Cheng, Tay, & Huang, 2012; Singh, Kumar, & Singla, 2015), have been applied, along with models combining several techniques (Hasan, Nasser, Pal, & Ahmad, 2013; Panda, Abraham, & Patra, 2012). Each model offers specific strengths and weaknesses, with overall generic detection rates steadily increasing.

SVMs exhibit good detection performance with IDSs in terms of classifying the flow of a network into normal or abnormal behaviors. Horng et al. (2011) proposed an IDS based on a combination of BIRCH hierarchical clustering and SVM technique. Their proposed method achieved a good accuracy of up to 95.72% with a false alarm rate of 0.7%. Feng et al. (2014) introduced an approach combining SVM with self-organized ant colony network. This approach exhibited a good detection rate of up to 94.86% and a false positive rate of 6.01%. Kuang et al. (2014) proposed an IDS based on a combination of the SVM model with kernel principal component analysis (KPCA) and genetic algorithm (GA). KPCA was used to reduce the dimensions of feature vectors, whereas GA was employed to optimize the SVM parameters. The average detection rate was 95.26%, whereas the average false alarm rate was 1.03%.

ELMs exhibit performance comparable with that of SVMs in terms of classifying instances of IDS. Cheng et al. (2012) applied a kernel-based ELM for multi-class classification of IDS. Their results showed the ELM exhibited high data classification accuracy,

* Corresponding author.
E-mail addresses: wathiqpro@gmail.com (W.L. Al-Yaseen), zao@ukm.edu.my, zakree@ukm.edu.my (M.Z.A. Nazri).

http://dx.doi.org/10.1016/j.eswa.2016.09.041
0957-4174/© 2016 Elsevier Ltd. All rights reserved.

exceeding that of SVM; the accuracy of kernel ELM was 98.44%, whereas that of SVM was 98.19%. Singh et al. (2015) presented an IDS based on online sequential ELM. They used NSL-KDD dataset to evaluate the proposed technique. The OS-ELM achieved 97.67% accuracy with 1.74% false positive rate for multi-class classification. In the present study, ELM outperforms other methods, such as ANN, in classifying attacks under the Probe category. However, its performance in terms of classifying attacks falling under other categories is poor. A combination of SVM and ELM can improve attack-detection performance.

Owing to the complexity and diversity of intrusions, few researchers proposed a model that used the same technique many times in a multi-level model to classify these intrusions (Rajeswari & Kannan, 2008; Xiang, Chong, & Zhu, 2004). Other scholars designed models using different techniques in multiple levels (Ibrahim, Badr, & Shaheen, 2012; Selim, Hashem, & Nazmy, 2011; Xiang, Yong, & Meng, 2008). These models exhibited good attack detection performance. This study proposes a new multi-level model for IDS that combines SVM with ELM to reduce the false alarm rate while improving detection accuracy. SVM-based classification models have gained considerable interest and achieved significant success over the past few years. Such SVM separates data into multiple classes (at least two) using a hyperplane. SVM also attempts to identify the maximum margin among different classes to separate the data with high accuracy. ELM can deal with large training and testing data within a short period and avoid the local minima problem. Basic ELM has slightly lower accuracy than SVM. To solve this problem, kernel-based ELM can be implemented to increase detection accuracy (Cheng et al., 2012). The KDD Cup 1999 dataset is used to evaluate the performance of our proposed model. The training dataset of the KDD Cup 1999 is large and imbalanced, such that the training complexity of SVM is high, while both SVM and ELM are biased to large classes. We use modified K-means clustering to build a new training dataset for each category of the KDD Cup 1999. The pre-processing of the KDD Cup 1999 dataset is implemented to convert and normalize the data before clustering.

The rest of this work is organized as follows. Related work of the training SVM phase and multi-level models are presented in Section 2. The backgrounds of ELM and SVM are introduced in Section 3. The proposed multi-level hybrid SVM and ELM based on the modified K-means model is expounded in Section 4. The experimental results are discussed in Section 4. Conclusions and future assignments are offered in Section 5.

2. Related work

Related literature on reducing the training time of SVMs is presented in this section, followed with a discussion of the literature on implementing multi-level IDS.

The disadvantage of reducing the training time of SVMs is that training can take a long time, particularly when the training dataset is extremely large. Khan, Awad, and Thuraisingham (2007) used hierarchical clustering, that is, dynamically growing self-organizing tree (DGSOT), to reduce the training time of SVMs. DGSOT is used to find the boundary points between two classes in a training dataset. The boundary points are the most qualified points to train SVMs. Balcázar, Dai, and Watanabe (2001) also successfully used random sampling or sub-sampling techniques in many applications. Similarly, Shih, Rennie, and Karger (2003) used data reduction technique (i.e., Rocchio algorithm) to speed up the classifier work by randomly removing some training sampling and sub-sampling the remaining training dataset. Yu, Yang, and Han (2003) used hierarchical clustering BIRCH to handle the large dataset in the computer memory and built a tree for training SVMs with its nodes. Horng et al. (2011) also used the BIRCH technique to reduce the training of KDD Cup 1999 dataset and provide SVMs with high quality sampling. Kuang et al. (2014) used the kernel principal component analysis to reduce the dimension of feature vectors, thus further reducing the training time of SVMs. On the contrary, ELM requires short time because it is built on a one-time training using Moore–Penrose pseudo-inverse to solve a least-squares equation (Creech & Jiang, 2012).

With regard to building multi-level IDS, Selim et al. (2011) developed a hybrid multi-level IDS using different neural network and decision tree techniques. Each level is implemented through a highly accurate technique – C5 and MLP techniques showed the best results. Ibrahim et al. (2012) likewise applied a multi-level model with different machine learning techniques, such as C5, MLP, and Naïve Bayes. The study used one of the techniques at each level to classify one category, thereby confirming that multi-level techniques exhibit higher detection accuracy than a single technique. Lu and Xu (2009) proposed a novel IDS by combining supervised classifiers with unsupervised clustering. Three levels were implemented to improve the performance of IDS with C4.5, Naïve Bayes, and Bayesian clustering. Xiang et al. (2008) also designed a multi-level hybrid classifier by combining decision tree and Bayesian clustering to detect intrusions. This proposed approach was efficient in detecting attacks with a false alarm rate of 3.2%. Similarly, Rajeswari and Kannan (2008) presented a multi-level hybrid classifiers using C4.5 at each level to classify data with a false alarm rate of 9.1%. Gogoi, Bhattacharyya, Borah, and Kalita (2014) proposed a multi-level hybrid IDS using a combination of supervised, unsupervised, and outlier methods. This system was evaluated with three datasets, namely, real-time flow dataset, DDoS dataset, and the KDD Cup 1999 with NSL-KDD datasets. The system performance was good with a false alarm rate of 3.4% with the corrected KDD Cup 1999 dataset.

The present study proposes a modified K-means to reduce the size of training dataset as well as balance the training dataset for training SVMs and ELMs. Essentially, modified K-means is used to reduce the training time of techniques and handle the redundancy of training dataset. The study also proposes a new multi-level model for implementing SVMs and ELMs with high performance and low false alarm rate.

2.1. Extreme learning machine

Huang, Zhu, and Siew (2004)and Huang, Wang, and Lan (2011) proved that single-hidden layer-feed-forward neural network – also termed as ELM – can exactly learn N distinct observations about almost any nonlinear activation function with at most N hidden nodes. The essential difference between ELM and traditional training of a feed-forward network is the hidden layer of ELM does not need tuning, in which the parameters of hidden layer are randomly chosen. The input weights and hidden neurons biases, as well as the output weights of the hidden layer, are assigned randomly to minimize training error. ELM transforms the learning problem into a simple linear system in which the output weights can be analytically determined. The results (Huang, Zhou, Ding, & Zhang, 2012) revealed that ELM performed well and was relatively easy to implement. For N arbitrary distinct instances $\{(x_i, \gamma_i) \ : i = 1, 2, \ldots, N\}$, where $x_i = [x_{i1}, \ x_{i2}, \ \ldots, \ x_{in}]^T \in R^n$ and $\gamma_i = [\gamma_{i1}, \gamma_{i2}, \ldots, \gamma_{im}]^T \in R^m$, an ELM with n inputs, m outputs, k hidden neurons, and an activation function $g(x)$ is modeled as

$$\sum_{i=1}^{k} \beta_i g\left(w_i^T x_j + b_i\right) = o_j, \ \ j = 1, 2, \ldots, N,$$

where $w_i = [w_{i1}, w_{i2}, \ldots, w_{in}]^T$ and $\beta_i = [\beta_{i1}, \ \beta_{i2}, \ \ldots, \beta_{im}]^T$ represent the weight vectors connecting the input neurons to an ith hidden neurons and from the *i*th hidden neurons to the output neurons, respectively, and b_i is a threshold of the *i*th hidden neurons. The ELM with $k = N$ hidden neurons can reliably approximate these N instances with zero error as

$$\sum_{j=1}^{N} \left\| o_j - \gamma_j \right\| = 0$$

$$\sum_{i=1}^{k} \beta_i g\left(w_i^T x_j + b_i\right) = \gamma_j, \;\; j = 1, 2, \ldots, N.$$

The equation above can be re-written as $Y\beta = \Gamma$ where

$$Y = \begin{bmatrix} g\left(w_1^T x_1 + b_1\right) & \cdots & g\left(w_k^T x_1 + b_k\right) \\ \vdots & \cdots & \vdots \\ g\left(w_1^T x_N + b_1\right) & \cdots & g\left(w_k^T x_N + b_k\right) \end{bmatrix}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_k^T \end{bmatrix} \text{ and } \Gamma = \begin{bmatrix} \gamma_1^T \\ \vdots \\ \gamma_N^T \end{bmatrix}.$$

The matrix Υ is the hidden layer output matrix of ELM, where the *i*th column of Υ is the *i*th hidden neuron output with respect to inputs $x_1, x_2, \ldots, x_N$.

In basic ELM, when $k << N$ and Υ is a non-square matrix, the learning of ELM is equal to finding a least-squares solution $\hat{\beta}$ of the linear system $Y\beta = \Gamma$. The norm least-squares solution of the linear system is $\hat{\beta} = Y^{*}\Gamma$, where $Y*$ is the Moore–Penrose generalized inverse of the matrix Υ. For kernel-based ELM, several nonlinear kernel functions can be used to calculate the hidden layer feature mapping of ELM. One of the popular kernel functions is Gaussian radial basis function kernel.

2.2. Support vector machine

SVM classifier is a machine learning technique based on statistical learning theories. This classifier builds a mechanism to separate data into different categories by an *N*-dimensional hyper plane that computes from a given training dataset. The instances of training dataset are labeled as $\{(x_i, y_i)\}$, $i = 1, 2, \ldots, N$, where N represents the number of instances, and y_i is the class of instance x_i in the training dataset. The main issue of SVMs is determining the maximum margin separating hyper plane from the closest points in a high dimensional space, where SVMs compute the sum of distances between points of hyper plane to the closest points of dimensional space (Golmah, 2014). The boundary function of the largest margin can be computed as follows (Hsu, Chang, & Lin, 2003):

$$Minimise\ W(\alpha)\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j k\left(x_i, x_j\right) - \sum_{i=1}^{N}\alpha_i$$

subject to

$$\forall i : 0 \le \alpha_i \le C \text{ and } \sum_{i=1}^{N}\alpha_i y_i = 0,$$

where α is a vector of N variables, C is the soft margin parameter, $C > 0$, and $k(x_i,\ x_j)$ represents the kernel function of SVMs. These sets of kernel functions can be used with SVMs to separate the instances of data into different categories, and these kernel functions are as follows (Hsu et al., 2003):

- Linear kernel: $k(x_i,\ x_j) = x_i^T.x_j$
- Polynomial kernel: $k(x_i,\ x_j) = (\gamma x_i^T.x_j + r)^d,\ \gamma > 0$
- Radial basis function (RBF) kernel: $k(x_i,\ x_j) = exp(-\gamma \|x_i - x_j{}^2\|),\ \gamma > 0$
- Sigmoid kernel: $k(x_i,\ x_j) = tanh(\gamma x_i^T.x_j + r)$

Where γ, r and d are kernel parameters.

3. Proposed multi-level hybrid SVM and ELM based on modified K-means for IDS

This section describes the proposed method of combining the modified K-means with SVM and ELM as a multi-level model for IDS.

The KDD Cup 1999 benchmark is employed to evaluate the performance of the proposed model. A 10% KDD training dataset is used to train SVM and ELM, and it features a large number of instances equal to 494,021. If the entire dataset is used for training the proposed model, then several problems can occur. The most important ones are the training time of SVMs is rather long, and a system goes to a single point of failure because of memory overflow. The modified K-means is used to reduce the size of dataset and build a new small high-quality training dataset. *High quality* means the instances of resulting dataset should represent all the instances in an original training dataset. This high-quality dataset allows SVM and ELM to exhibit high detection and low false alarm rates. The main goal of separating the original training data into five categories and rebuild new training datasets with a few instances is to reduce the training time of classifiers that may last for more than one hour if training with original training data as well as retain the performance of classifiers. The scenario of the proposed method is described as follows:

Step 1. Convert the symbolic attributes *protocol, service,* and *flag* to numerical ones (Sabhnani & Serpen, 2003).
Step 2. Normalize data to [0,1] as (Sabhnani & Serpen, 2003).
Step 3. Separate the instances of 10% KDD training dataset into five categories: Normal, DoS, Probe, R2L, and U2R.
Step 4. Apply modified K-means on each category and create new training datasets.
Step 5. Train SVM and ELM with these new training datasets.
Step 6. Test multi-level model with corrected KDD dataset.

Additional details about Steps 4 to 6 are provided.

3.1. Proposed modified K-means

Several methods have been proposed to improve the performance of K-means. Most of these methods aim to solve the initial centroids of clusters and identify the advanced number of clusters (Al-daoud, 2007; Arthur, Arthur, Vassilvitskii, & Vassilvitskii, 2007; Erisoglu, Calis, & Sakallioglu, 2011; Katsavounidis, Kuo, & Zhang, 1994). The present study proposes a method for selecting the initial centroids of clusters depending on a distance threshold. The distance threshold represents the maximum distance between centroids of clusters and the instances of dataset. For example, if the distance between an instance and the centroid of the cluster is less than the threshold, then the instance belongs to the cluster; otherwise, the method creates a new cluster with this instance. The process applies to all instances of dataset. The first centroid is chosen as the first instance of the dataset. This scenario identifies the number of clusters dynamically. The following are the steps of modified K-means:

(1) Identify the distance threshold (φ).
(2) Set the first instance of the dataset as the first centroid (C_1) and set $k = 1$.
(3) Read the next instance (S).
(4) $\forall C_i$, $1 \le i \le k$, If $\exists\, C_j \in \{C_i\}$; distance $(S,\ C_j) < \varphi$ then put S in cluster *j*. Go to Step 6.

Table 1
Number of instances before and after applying modified K-means on 10% KDD dataset.

Category	# of instances/before	# of instances/after
Normal	97,278	639
DoS	391,458	140
Probe	4107	134
R2L	1126	51
U2R	52	25
Total	494,021	989

Table 2
Best values of SVM parameters.

Category	nu	gamma (γ)
Normal	0.06	0.09
DoS	0.004	0.5
Probe	0.1	0.3
R2L	0.05	0.008
U2R	0.05	0.008

(5) Create a new cluster with centroid S, set $k = k + 1$.
(6) Repeat Steps 3–5 to reach the end of the dataset.
(7) Continue with the other basic K-means steps.

The modified K-means is used for clustering each category into a set of clusters (e.g., normal results were obtained from Step 3 in the scenario of the proposed method). The centroids of final clusters are calculated, and each centroid is assigned as a new instance in this category. The number of instances before and after applying these steps is shown in Table 1.

Subsequently, five training datasets are constructed from the new instances – one training dataset for every category. For example, the constructed Normal training dataset has 989 instances, assuming 1 for instances of Normal category and 2 for instances from other categories, and so on. We obtain five new training datasets, namely, *Normal-DS, DoS-DS, Probe-DS, R2L-DS*, and *U2R-DS*, that are used to build classifiers of SVM and ELM. This approach is known as one-versus-all for training classifiers. The grid search (Staelin, 2003) is used to identify the best values of SVM parameters (*nu* and *gamma*). Table 2 presents the best values of SVM parameters used in our experiments.

3.2. Proposed multi-level hybrid SVM and ELM

Several models have been proposed to design multi-level IDS. Some models classified DoS and Probe categories at the first level, Normal category at the second level, and both R2L and U2R categories at the third or last level (Gogoi et al., 2014; Rajeswari & Kannan, 2008; Xiang et al., 2008). By contrast, other models classified Normal, DoS, and Probe at the first level and R2L and U2R at the second level (Xiang et al., 2004). Lu and Xu (2009) classified DoS and Probe at the first level, U2R at the second level, and Normal together with R2L at the third level.

From this context, the present study divides the categories into three groups: Group 1, DoS and Probe; Group2, U2R and R2L; and Group 3, Normal.

The study proposes a new multi-level IDS model using hybrid SVM and ELM classifiers to analyze the traffic data network. The best multi-level model is chosen through the configuration of two different structures from the groups above, as shown in Fig. 1.

The DoS and Probe are fixed at the first level of the proposed model with respect to the related literature. In general, DoS and Probe categories have minimal similarity with the other categories (Gogoi et al., 2014).

The number of attacks on the network is a small fraction when compared with normal traffic. Sharma and Mukherjee (Sharma & Mukherjee, 2012) demonstrated that minor attacks, such as U2R and R2L, are dangerous for the network, given that these attacks are relatively similar to normal connections (Gogoi et al., 2014). We expect the first structure in Fig. 1(a) to be suitable for designing IDS because the attacks under the U2R and R2L categories are first detected before these attacks are classified as normal connections.

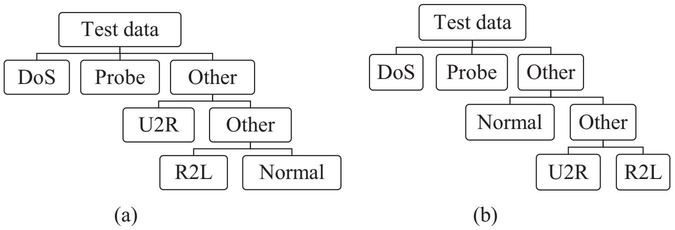

Fig. 1. Multi-level IDS models.

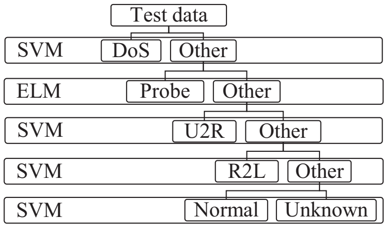

Fig. 2. Proposed multi-level hybrid SVM and ELM.

Similar to other multi-level models, the method used in this study uses one classifier at each level to classify the incoming packet into one of the categories. SVMs or ELMs are used to achieve the goal. Subsequently, we compare the application of three models, namely multi-level SVMs, multi-level ELMs, and multi-level hybrid SVM and ELM. Fig. 2 illustrates the new multi-level hybrid SVM and ELM proposed in this study. This proposed method uses four SVMs to classify instances as DoS, U2R, R2L, or Normal. These SVMs are already trained, as illustrated in the previous section, using the parameters in Table 2. We use an ELM classifier to detect the attacks under the Probe category because this classifier can classify Probe attacks better than SVM. In the case of non-classification of instances into known categories, the model considers these instances as unknown attacks.

3.3. Processing of training and testing data

In this section, a number of training and testing examples are provided to demonstrate the mechanism implemented by the proposed method, namely, modified K-means and multi-level hybrid SVM and ELM. Table 3 presents the details of KDD Cup 1999 categories with the attacks.

Table 4 shows examples of original training and testing dataset before pre-processing operations are applied, which are conversion of symbolic attributes and normalization.

Table 5 lists the results of converting the symbolic attributes to numerical ones.

Then, normalization of data is implemented using the method of Sabhnani and Serpen. The results after normalization are shown in Table 6.

After completing the pre-processing of training dataset, the proposed method constructs five sets from the training dataset, namely, Normal, DoS, Probe, U2R, and R2L. Normal set features only the instances of normal classes, DoS set possesses the instances of attacks included in the DoS category of Table 3, and so on. A modified K-means is then applied on each set to build new five sets, which are described in Table 1. The classifiers of multi-level hybrid model are trained using new binary training datasets

Table 3
Classes and numbers of attacks for each category of KDD Cup 1999 dataset.

Category	10% KDD dataset (Training)		Corrected dataset (Testing)			
			Known attacks		New attacks	
	Attack	#Instances	Attack	#Instances	Attack	#Instances
DoS	Smurf	280,790	Smurf	164,091	Apache2	794
	Neptune	107,201	Neptune	58,001	Mailbomb	5000
	Teardrop	979	Teardrop	12	Udpstorm	2
	Land	21	Land	9	Processtable	759
	Pod	264	Pod	87		
	Back	2203	Back	1098		
Probe	Ipsweep	1247	Ipsweep	306	Mscan	1053
	Portsweep	1040	Portsweep	354	Saint	736
	Satan	1589	Satan	1633		
	Nmap	231	Nmap	84		
R2L	Ftp_write	8	Ftp_write	3	Named	17
	Guess_passwd	53	Guess_passwd	4367	Sendmail	17
	Imap	12	Imap	1	Snmpgetattack	7741
	Multihop	7	Multihop	18	Snmpguess	2406
	Phf	4	Phf	2	Worm	2
	Warezmaster	20	Warezmaster	1602	Xlock	9
	Warezclient	1020			Xsnoop	4
	Spy	2				
U2R	Perl	3	Perl	2	Ps	16
	Loadmodule	9	Loadmodule	2	Sqlattack	2
	Buffer_overflow	30	Buffer_overflow	22	Xterm	13
	Rootkit	10	Rootkit	13	Httptunnel	158

Table 4
Samples of raw training and testing KDD Cup 1999 dataset.

No	Instance
1	0,tcp,http,SF,181,5450,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,9,9,1.00,0.00,0.11,0.00,0.00,0.00,0.00,0.00,normal.
2	0,icmp,ecr_i,SF,1032,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,511,511,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,119,0.47,0.02,0.47,0.00,0.02,0.00,0.00,0.00,smurf.
3	0,tcp,private,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,288,14,1.00,1.00,0.00,0.00,0.05,0.06,0.00,255,14,0.05,0.07,0.00,0.00,1.00,1.00,0.00,0.00,neptune.
4	0,icmp,eco_i,SF,8,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,34,0.00,0.00,0.00,0.00,1.00,0.00,1.00,2,10,1.00,0.00,1.00,0.50,0.00,0.00,0.00,0.00,ipsweep.
5	26,tcp,ftp,SF,116,451,0,0,0,2,0,1,0,0,0,0,1,0,1,0,0,1,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,1,1,1.00,0.00,1.00,0.00,0.00,0.00,0.00,0.00,ftp_write.
6	0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,254,1.00,0.01,0.01,0.00,0.00,0.00,0.00,0.00,snmpgetattack.

Table 5
Results of converting the samples in Table 4..

No	Instance
1	0,0,0,0,181,5450,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,9,9,1.00,0.00,0.11,0.00,0.00,0.00,0.00,0.00,normal.
2	0,2,1,0,1032,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,511,511,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,119,0.47,0.02,0.47,0.00,0.02,0.00,0.00,0.00,smurf.
3	0,0,2,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,288,14,1.00,1.00,0.00,0.00,0.05,0.06,0.00,255,14,0.05,0.07,0.00,0.00,1.00,1.00,0.00,0.00,neptune.
4	0,2,3,0,8,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,34,0.00,0.00,0.00,0.00,1.00,0.00,1.00,2,10,1.00,0.00,1.00,0.50,0.00,0.00,0.00,0.00,ipsweep.
5	26,0,4,0,116,451,0,0,0,2,0,1,0,0,0,0,1,0,1,0,0,1,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,1,1,1.00,0.00,1.00,0.00,0.00,0.00,0.00,0.00,ftp_write
6	0,1,2,0,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,254,1.00,0.01,0.01,0.00,0.00,0.00,0.00,0.00,snmpgetattack.

Table 6
Results of normalizing the samples in Table 5.

No	Instance
1	0,0.5,0,0,0.000002,0.000028,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.001957,0.001957,0,0,0,0,1,0,0,1,0.996078,1,0.01,0,0,0,0,0,0,normal.
2	0,1,0.333333,0,0.000001,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.618395,0.618395,0,0,0,0,1,0,0,0.580392,0.011765,0.02,0.02,0.02,0,0,0,0,0,smurf.
3	0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.215264,0.015656,1,1,0,0,0.07,0.06,0,0.043137,0.031373,0.73,0.27,0.09,0,1,1,0,0,neptune.
4	0,1,0.090909,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.001957,0.099804,0,0,0,0,1,0,1,0.003922,0.035294,1,0,1,0.56,0,0,0,0,ipsweep.
5	0.000446,0,0.075758,0,0,0.000087,0,0,0,0.066667,0,1,0,0,0,0,0.035714,0,0.125,0,0,1,0.001957,0.001957,0,0,0,0,1,0,0,0.003922,0.003922,1,0,1,0,0,0,0,0,ftp_write.
6	0,0.5,0,0,0.000002,0.000028,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.003914,0.003914,0,0,0,0,1,0,0,1,0.996078,1,0.01,0,0,0,0,0,0,snmpgetattack.

obtained from combining the new sets. The multi-level hybrid SVM and ELM can analyze any testing data.

In the testing phase, the normalized testing dataset can be analyzed using the obtained model from training phase (Fig 2). Table 7 shows a number of testing examples with their classification results. The first normal instance is classified correctly as normal class at level 5 by obtaining false results from all the previous levels. The second normal instance is classified incorrectly in level 3 as an U2R attack. The same goes for the remaining instances. The last instance in Table 7, which is a named attack, is classified as unknown attack because of the absence of a positive result from all the classifiers with this instance.

4. Experimental results

In this section, we evaluate the performance of the proposed multi-level hybrid SVM and ELM model. All experiments were conducted on a Windows 8.1 PC with Intel Core i5 CPU @2.60 GHz and 12 GB RAM. The implementation was coded using the Java language, and nu-SVC with RBF kernel of LIBSVM (version 3.20, Java

Table 7
Samples of testing dataset with their classification results using the trained multi-level hybrid SVM and ELM model.

Test data	L-1 DoS	L-2 Probe	L-3 U2R	L-4 R2L	L-5 Normal	Unknown	Remark
0,0.5,0,0,0.000002,0.000028,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.001957,0.001957,0,0,0,0,1,0,0,1,0.996078, 1,0.01,0,0,0,0,0,0,normal.	no	no	no	no	yes		Right
0,0.5,0,0,0.000002,0.000028,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.001957,0.001957,0,0,0,0,1,0,0,1,0.996078, 1,0.01,0,0,0,0,0,0,normal.	no	no	yes				Wrong
0,0.5,0,0,0.000002,0.000028,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.003914,0.003914,0,0,0,0,1,0,0,1,0.996078, 1,0.01,0,0,0,0,0,0,snmpgetattack.	no	no	no	no	yes		Wrong
0,0,0,0.3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.272016,0.021526,0,0,1,1,0.08,0.06,0, 1,0.043137,0.04,0.06,0,0,0,0,1,1,neptune.	yes						Right
0.001299,0,0.203125,0,0.000004,0.000209,0,0,0,0.019802,0,1,0.001256,1,0,0,0.04,0.4,0,0,0,0,0,0.001957, 0.001957,0,0,0,0,1,0,0,0.537255,0.396078,0.74,0.04,0.01,0,0.73,0.99,0,0,loadmodule.	no	no	yes				Right
0,1,0.09375,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.001957,0.001957,0,0,0,0,1,0,0,0.003922, 0.290196, 1,0,1,1,0,0,0,0,ipsweep.	no	yes					Right
0,0,1,0,0.000025,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.001957,0.001957, 0,0,0,0,1,0,0,0.019608,0.003922,0.2,0.4,0.2,0,0,0,0,0,0,named.	no	no	no	no	no	yes	

Table 8
Number instances of training and testing datasets for KDD Cup 1999.

Category	10% KDD	Corrected
Normal	97,278 (19.69%)	60,593 (19.48%)
DoS	391,458 (79.24%)	229,853 (73.90%)
Probe	4107 (0.83%)	4166 (1.34%)
U2R	52 (0.01%)	228 (0.07%)
R2L	1126 (0.23%)	16,189 (5.20%)
Total	494,021	311,029

Table 9
Comparison of the performance between modified K-means and basic K-means based on multi-level hybrid SVM and ELM.

	Basic K-means	Modified K-means
Acc	91.88	**95.75**
DR	92.13	**95.17**
FAR	9.16	**1.87**

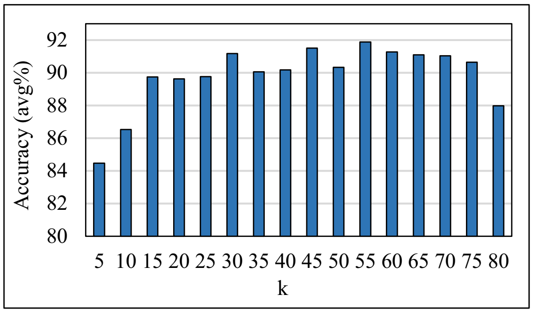

Fig. 3. Best number of clusters for basic K-means.

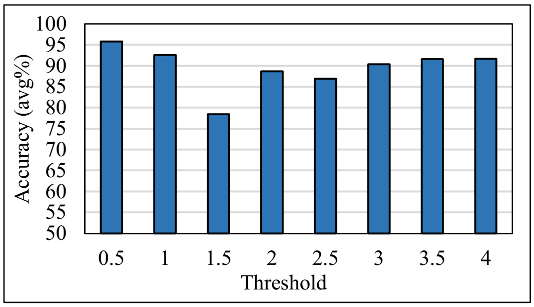

Fig. 4. Best distance threshold for modified K-means.

version) (http://www.csie.ntu.edu.tw/~cjlin/libsvm/) was applied. The ELM Java version of Dong Li (http://www.ntu.edu.sg/home/egbhuang/elm_codes.html) was also implemented. The model was evaluated using the KDD Cup 1999 dataset.

4.1. Training and testing datasets

The KDD Cup 1999 dataset was used for the Third International Knowledge Discovery and Data Mining Tools Competition. Each connection instance is described by 41 attributes (38 continuous or discrete numerical attributes and 3 symbolic attributes). Each instance is labeled as either normal or a specific type of attack. These attacks fall under one of the four categories: DoS, Probe, U2R, and R2L. KDD Cup 1999 provided both the training and testing datasets, which are called 10% KDD and corrected dataset, respectively. The 10% KDD dataset contains 22 types of attacks, whereas the corrected dataset features the same 22 types of attacks, along with 17 additional attack types. The details of the training and testing datasets are shown in Table 8.

4.2. Experiments

The first experiment evaluated the performance of modified K-means. The objective of modified K-means is to improve the performance of SVM and ELM classifiers. This experiment compared the performance of modified K-means with that of basic K-means by implementing them with the multi-level hybrid SVM and ELM. In doing so, we identified the best k for basic K-means and the best distance threshold for modified K-means. Fig. 3 shows that the best number of clusters for the basic K-means algorithm is 55, through which the highest average accuracy of 91.88% was achieved. Fig. 4 indicates that the best distance threshold for the modified K-means is 0.5. This study used $k = 55$ for basic K-means and $threshold = 0.5$ for modified K-means for all comparisons.

Table 9 compares the performance of modified K-means with that of basic K-means, whereas Fig. 5 illustrates the ROC curve of both methods, which reveals that the performance of modified K-means with multi-level hybrid SVM and ELM is significantly better than that of basic K-means with multi-level hybrid SVM and ELM.

The performance of the proposed multi-level hybrid SVM and ELM model is better than those of the multi-level SVM model and multi-level ELM model, separately. Consequently, the proposed hybrid model can better classify the traffic data than a single model. The detection rates for the five categories are shown in Table 10, whereas the overall accuracies (Acc), detection rates (DR), and false alarm rates (FAR) are depicted in Table 11. The ROC curve of comparison between these models is shown in Fig. 6.

The proposed multi-level hybrid SVM and ELM achieves the best performance (Table 11). The detection rate of R2L is minimal compared with that of the other categories because some of the

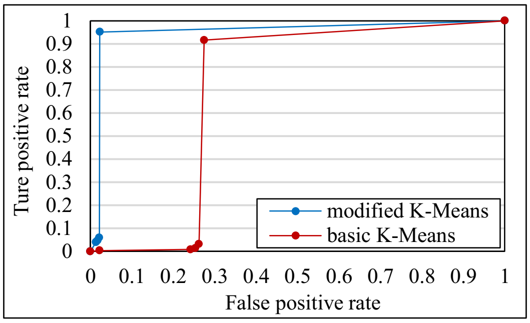

Fig. 5. ROC curve for comparing performance of modified K-means with that of basic K-means.

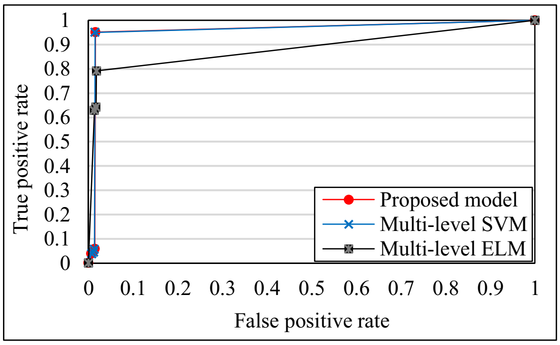

Fig. 6. ROC curves for comparing performance of proposed model with that of two other models.

Table 10
Comparison of detection rates.

Category	Multi-level SVM	Multi-level ELM	Proposed model
Normal	97.83	96.64	**98.13**
DoS	**99.57**	96.83	99.54
Probe	80.94	84.93	**87.22**
U2R	16.23	**23.68**	21.93
R2L	**31.60**	10.14	31.39

Table 11
Comparison of overall accuracies, detection rates, and false alarm rates.

	Multi-level SVM	Multi-level ELM	Proposed model
Acc	95.57	93.83	**95.75**
DR	95.02	93.15	**95.17**
FAR	2.17	3.36	**1.87**

attacks included in R2L, such as *snmpgetattack* (7741/16189) and *snmpguess* (2406/16189), exhibit features that are highly similar to those of Normal and may match these features 100% such that they cannot be 100% classified as attacks. The low detection rate for U2R can be attributed to the very small number of instances (288) compared with that of other categories. Some of the attacks in U2R also show features that are similar to instances of the Normal category.

To compare accurately, Table 12 compares the proposed method with approaches that use only the entire corrected KDD Cup 1999 dataset as a testing dataset because several researchers used only part of the corrected KDD Cup 1999 dataset at random to evaluate their works, whereas others used 10-fold cross validation with 10% KDD dataset.

The proposed model provides balanced detection rates among categories, with the multi-class SVM (2003) having the highest detection rate for Normal but poor detection rates for other categories. The proposed method also has the best detection rate for DoS. The SVM and BIRCH clustering model has the best detection rate for Probe, whereas the proposed model has better detection rates for U2R and R2L. Furthermore, the detection rate of the AIS and SOM model (2008) for U2R is higher, but other detection rates are poor, particularly for Probe and R2L. Overall, in terms of accuracy, this system achieves a top performance of up to 95.75% with a reasonable false alarm rate of 1.87%.

Many new attacks in the corrected dataset have not appeared in the training dataset, and they account for approximately 10% of the KDD. They cripple the performance of classifiers, and allocating the attacks becomes difficult. A total of 18,729 instances of new attacks are classified by proposing a model with a detection rate of 40.02% compared to that of Horng et al. (2011), which has a new attack detection rate of 39.04%.

The above results show that multi-level hybrid SVM and ELM enhances the performance of IDS. Multi-level hybrid SVM and ELM is more reliable than state-of-the-art methods, and it does not cause large fluctuations in detection performance. Multi-level hybrid SVM and ELM can also isolate the unknown attacks effectively and continuously, improving its detection performance.

The strengths of the proposed method are the highly improved detection accuracy compared with other methods as well as the short training time because of the high reduction of original training dataset size. The proposed method used more than one classifier, resulting in longer testing time compared with methods using only one classifier.

5. Conclusion and future work

This study proposes a new multi-level hybrid SVM and ELM model-based network IDS. The proposed model is marked by a significantly better performance than multi-level SVM and multi-level ELM models. The modified K-means is designated to pre-process the training dataset and provide new high-quality training datasets that can improve the training time and overall performance of SVM and ELM.

According to the experiments on the KDD Cup 1999 dataset, the performance of the proposed model in terms of accuracy can achieve up to 95.75% with a false alarm rate of 1.87%. This system

Table 12
Comparison of proposed model with other methods by detection rate, accuracy, and false alarm rate.

Method	Normal	DoS	Probe	U2R	R2L	Accuracy	DR	FAR
Propose method	98.13	**99.54**	87.22	**21.93**	**31.39**	**95.75**	95.17	1.87
NFC (He, 2014)	98.2	99.5	84.1	14.1	31.5	N/A	**95.2**	1.9
Genetic algorithm (Hoque, Mukit, & Bikas, 2012)	69.5	99.4	71.1	18.9	5.4	90	94.95	30.46
SVM+BIRCH clustering (Horng et al., 2011)	99.3	99.5	**97.5**	19.7	28.8	95.7	N/A	0.7
MOGFIDS (Tsang, Kwong, & Wang, 2007)	98.36	97.2	88.6	15.79	11.01	93.2	91.96	1.6
Association rules (Xuren, Famei, & Rongsheng, 2006)	99.47	96.6	74.8	3.8	1.21	92.4	N/A	0.53
Multiclass SVM (Ambwani, 2003)	**99.6**	96.8	75	5.3	4.2	92.46	90.74	**0.43**
Winning the KDD99 (Elkan, 2000)	99.5	97.1	83.3	13.2	8.4	93.3	91.81	0.55

is implemented with the entire training and testing KDD Cup 1999 dataset. The main contribution of this study is presenting a model with superior performance compared with those presented in the most similar studies by providing a balanced performance among all categories. The system can also improve the detection rate of new attacks that did not appear in the training dataset, reaching 40.02%. In our future research, we will strive to construct a more effective model based on efficient classifiers to competently classify new attacks with high performance. We will also exploit the characteristics of multi-agent system to speed up data analysis and facilitate model retraining on new attacks to increase the efficiency of the system.

Acknowledgement

This work is supported in part by the National University of Malaysia (UKM), Grant no. AP2013-007.

References

Al-daoud, M. B. (2007). A new algorithm for cluster initialization. *International Journal of Computer, Information, Mechatronics, Systems Science and Engineering, 1*(4), 1026–1028.

Ambwani, T. (2003). Multi class support vector machine implementation to intrusion detection. In *Proceedings of the international joint conference on neural networks, 2003: Vol. 3* (pp. 2300–2305).

Arthur, D., Arthur, D., Vassilvitskii, S., & Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms* (pp. 1027–1035).

Balcázar, J., Dai, Y., & Watanabe, O. (2001). A random sampling technique for training support vector machines. In *Algorithmic learning theory* (pp. 119–134). Berlin Heidelberg: Springer.

Cheng, C., Tay, W.-P., & Huang, G.-B. (2012). Extreme learning machines for intrusion detection. In *WCCI 2012 IEEE world congress on computational intelligence* (pp. 1–8).

Creech, G., & Jiang, F. (2012). The application of extreme learning machines to the network intrusion detection problem. In *Numerical analysis and applied mathematics ICNAAM: Vol. 1479* (pp. 1506–1511).

Elkan, C. (2000). Results of the KDD'99 classifier learning. *ACM SIGKDD Explorations Newsletter, 1*(2), 63–64.

Erisoglu, M., Calis, N., & Sakallioglu, S. (2011). A new algorithm for initial cluster centers in k-means algorithm. *Pattern Recognition Letters, 32*(14), 1701–1705.

Feng, W., Zhang, Q., Hu, G., & Huang, J. X. (2014). Mining network data for intrusion detection through combining SVMs with ant colony networks. *Future Generation Computer Systems, 37*, 127–140.

Gogoi, P., Bhattacharyya, D. K., Borah, B., & Kalita, J. K. (2014). MLH-IDS: A multi--level hybrid intrusion detection method. *Computer Journal, 57*(4), 602–623.

Golmah, V. (2014). An efficient hybrid intrusion detection system based on C5. 0 and SVM. *International Journal of Database Theory & Application, 7*(2), 59–70.

Hasan, A. M., Nasser, M., Pal, B., & Ahmad, S. (2013). Intrusion detection using combination of various kernels based support vector machine. *International Journal of Scientific & Engineering Research, 4*(9), 1454–1463.

He, L. (2014). An improved intrusion detection based on neural network and fuzzy algorithm. *Journal of Networks, 9*(5), 1274–1280.

Hoque, M. S., Mukit, M. A., & Bikas, M. A. N. (2012). An implementation of intrusion detection system using genetic algorithm. *International Journal of Network Security & Its Applications, 4*(2), 109–120.

Horng, S. J., Su, M. Y., Chen, Y. H., Kao, T. W., Chen, R. J., Lai, J. L., et al. (2011). A novel intrusion detection system based on hierarchical clustering and support vector machines. *Expert Systems with Applications, 38*(1), 306–313.

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2003). *A practical guide to support vector classification*.

Huang, G., Zhu, Q.-Y., & Siew, C.-K. (2004). Extreme learning machine : A new learning scheme of feedforward neural networks. In *IEEE international joint conference on neural networks: Vol. 2* (pp. 985–990).

Huang, G.-B., Wang, D. H., & Lan, Y. (2011). Extreme learning machines: A survey. *International Journal of Machine Learning and Cybernetics, 2*(2), 107–122.

Huang, G.-B., Zhou, H., Ding, X., & Zhang, R. (2012). Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics, 42*(2), 513–529.

Ibrahim, H. E., Badr, S. M., & Shaheen, M. A. (2012). Adaptive layered approach using machine learning techniques with gain ratio for intrusion detection systems. *International Journal of Computer Applications, 56*(7), 10–16.

Katsavounidis, I., Kuo, C. C. J., & Zhang, Z. (1994). New initialization technique for generalized Lloyd iteration. *IEEE Signal Processing Letters, 1*(10), 144–146.

Khan, L., Awad, M., & Thuraisingham, B. (2007). A new intrusion detection system using support vector machines and hierarchical clustering. *The VLDB Journal, 16*, 507–521.

Kuang, F., Xu, W., & Zhang, S. (2014). A novel hybrid KPCA and SVM with GA model for intrusion detection. *Applied Soft Computing Journal, 18*, 178–184.

Lee, W., Stolfo, S. J., & Mok, K. W. (1999). A data mining framework for building intrusion detection models. In *Proceedings of the 1999 IEEE symposium on security and privacy* (pp. 1–13).

Lu, H., & Xu, J. (2009). Three-level hybrid intrusion detection system. In *Proceedings - 2009 international conference on information engineering and computer science, ICIECS 2009* (pp. 1–4).

Panda, M., Abraham, A., & Patra, M. R. (2012). A hybrid intelligent approach for network intrusion detection. *Procedia Engineering, 30*, 1–9.

Rajeswari, L. P., & Kannan, A. (2008). An intrusion detection system based on multiple level hybrid classifier using enhanced C4.5. In *IEEE international conference on signal processing, communications and networking* (pp. 75–79).

Roesch, M. (1999). Snort: lightweight intrusion detection for networks. In *LISA '99: 13th systems administration conference* (pp. 229–238).

Sabhnani, M., & Serpen, G. (2003). Application of machine learning algorithms to KDD intrusion detection dataset within misuse detection context. In *Proceedings of international conference on machine learning: models, technologies, and applications (MLMTA)* (pp. 209–215).

Selim, S., Hashem, M., & Nazmy, T. M. (2011). Hybrid multi-level intrusion detection system. *International Journal of Computer Science and Information Security, 9*(5), 23–29.

Sharma, N., & Mukherjee, S. (2012). A novel multi-classifier layered approach to improve minority attack detection in IDS. *Procedia Technology, 6*, 913–921.

Shih, L., Rennie, J. D. M., & Karger, D. R. (2003). Text bundling : Statistics-based data reduction. In *Proceedings of the twentieth international conference on machine learning (ICML-2003)* (pp. 1–8).

Singh, R., Kumar, H., & Singla, R. K. (2015). An intrusion detection system using network traffic profiling and online sequential extreme learning machine. *Expert Systems with Applications, 42*(22), 8609–8624.

Staelin, C. (2003). *Parameter selection for support vector machines*. Hewlett-Packard Company *HPL-2002-354 (R.1)*.

Tsang, C.-H., Kwong, S., & Wang, H. (2007). Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection. *Pattern Recognition, 40*, 2373–2391.

Xiang, C., Chong, M. Y., & Zhu, H. L. (2004). Design of multiple-level tree classifiers for intrusion detection system. In *Proceeding of the 2004 EEE conference on cybernetics and intelligent systems* (pp. 873–878).

Xiang, C., Yong, P. C., & Meng, L. S. (2008). Design of multiple-level hybrid classifier for intrusion detection system using Bayesian clustering and decision trees. *Pattern Recognition Letters, 29*(7), 918–924.

Xuren, W., Famei, H., & Rongsheng, X. (2006). Modeling intrusion detection system by discovering association rule in rough set theory framework. In *International conference on computational inteligence for modelling control and automation and international conference on intelligent agents, web technolgies and internet commerce (CIMCA-IAWTIC'06)* (pp. 1–6).

Yu, H., Yang, J., & Han, J. (2003). Classifying large data sets using SVMs with hierarchical clusters. In *ACM SIGKDD international conference on Knowledge discovery and data mining'03* (pp. 306–314).