Partial Least Square based Improved Intrusion Detection System

S. M. Sangve
Department of Computer Engineering
ZCOER, SPPU, Pune, India

U. V. Kulkarni
SGGS Institute of Engineering and Technology
SRTMU, Nanded, India

ABSTRACT

Various Artificial Intelligence (AI) based computing techniques for intrusion detection has been proposed using popular large-scale datasets like DARPA 98 and KDD Cup 99. However, AI based systems such as using representative instances are computationally inefficient. In this paper, the computationally efficient approach is proposed for anomaly detection by combining Partial Least Square (PLS) and technique of extracting representative instances. The PLS helps in feature selection and provides dimensionality reduction. Further, to decline the processing time the representative instances are properly chosen from the data set before classification. The classic instances are selected from the subsets of data which are obtained by Centroid-based partitioning technique. The system utilizes these paradigmatic instances as a training set. Finally, KNN classifier is trained using these paradigmatic instances. The results obtained using the proposed approach indicates a considerable fall in the processing time and space utilization.

General Terms

Network security, K-nearest neighbor classifier, Training dataset, Testing dataset, Anomaly based intrusion detection

Keywords

Intrusion detection, Artificial Intelligence, Feature selection, Preprocessing, Partial least square, Centroid-based classification

1. INTRODUCTION

The Internet is gaining more importance in different sectors like business and education centers that are utilizing Internet services (web and email for communication). The Internet users access these free services that make them susceptible to attacks which include data stealing [1]. For ensuring, the security policy of data, a modern computer network uses the intrusion detection system (IDS) which is an integral part of well-defined and organized network. The IDS can be a software program or hardware system which monitors the various actions occurring in the real network and analyzes the network for detection of security attacks [2].The intrusion detection system scans the network activity and finds out the attacks. Previously, many techniques have been developed for modeling normal and anomalous behaviour in the network. The most of the deployed techniques are misuse and supervised detections. But the problem generated from supervised detection is that they do not have enough labelled data. If a new type of intrusion comes in a network then the system is unable to capture it because no signature is available in labelled dataset. Thus, there is a need to update the dataset manually. This will consume more time and space. To get rid of these problems, unsupervised anomaly detection has been developed. It includes a set of unlabelled data. For detection of unknown attack, there is no need to keep previous knowledge of training dataset and new attacks.

There are many techniques developed for IDS using two broad categories as supervised (classification) and unsupervised (anomaly detection and clustering). To apply learning techniques for IDS, it is necessary to have the knowledge about the label information. To obtain the information of label can be very difficult because when we check the network traffic or audit logs it requires huge amount of time. Hence, in the real time applications the labelled set may not contain all possible types of attacks. If new attacks appear, the training dataset may not contain instances representing these fresh classes of attacks. Thus, it is important to have trade-off between supervised and unsupervised techniques for IDS [3]. Supervised algorithms are C4.5, k-nearest neighbor (KNN), and multilayer perceptron. The k-nearest neighbor finds the k-samples in training dataset that are closest to the test sample.

Most of the viable and open source IDS tools that are developed during last decades are signature-based. Such tools can detect only known attacks which are described previously by respective signatures. However, for new attacks, the signature databank should be stored and changed manually. To cope up with this issue, machine learning systems are used to learn new attacks those are not previously defined in training dataset [3]. The signature-based IDS are unable to detect zero-day attacks like worms and spyware. To solve this problem, anomaly intrusion detection methods have been developed. The support vector machine (SVM) is one of the known machine learning algorithms to classify abnormal samples[1]. There are two main approaches used for intrusion detection namely signature-based(SIDS) and anomaly-based(AIDS). The taxonomy of labelled and non-labelled attacks is briefly described by Garcia-Teodoro et al. [4].

The various AI techniques such as Naive Bayes, KNN, decision trees, artificial neural networks (ANNs), and SVM have been applied for detecting intrusion. The most commonly used techniques for intrusion detection are SVM and KNN. The multilayer perceptron is the example of neural network architecture which is widely used to solve the problem of intrusion detection. When the research in IDS started, many of the researchers suggested the fusion approach to increase the detection accuracy. The notion behind using a fusion classifier is to put together some learning techniques to attain improved detection performance than a particular classifier [5].

There are certain issues addressed while implementing the IDS. The IDS should be effective and efficient in terms of computational cost. The effectiveness of IDS is calculated in terms of detection accuracy (DA) and false alarm rate (FAR), while the response time is used to measure the efficiency during a network attack [6]. For improving the competency of AIDS, the various research groups have used feature selection to eliminate repetition of data and to decrease the computational complexity of preprocessing. The feature selection results in the dimensionality reduction. It decreases the feature space by eliminating the repetition or removal of inappropriate features [7]. The feature selection facilitates to increase the effectiveness of intelligent algorithms.

On another hand, feature selection will not remove all irrelevant or redundant data samples due to a number of instances in the dataset. Some of the AI algorithms like ANN and SVM having a high computational cost, while handling a huge number of instances in datasets [8]. As the size of training dataset increases, the computational complexity also increases. Thus, for improving the performance of the system, the core idea presented in this paper is to increase the quality of training dataset by selecting proper qualitative training data samples.

In the proposed method, to enhance the efficacy of identifying intrusion, PLS is used. Further, the representative instances are selected with respect to class (normal/anomaly) from given dataset. The dataset is divided using centroid-based strategy. The new representative instances are selected from each training dataset. The check is performed for effectiveness of representative instances for intrusion detection. The goal is to pick a small subset, which becomes good representative of the original dataset. Thus, the high-quality subset with fewer occurrences from the original dataset is used to train the classifier. The contribution of this paper is as: a) Time and space complexity analysis of existing system, b) Efficient signature-based IDS, c) Comparison of existing and proposed approach.

The paper is structured as follows. Related work is discussed in the section two. The proposed method and implementation description is given in the section three. The discussion on the obtained results using proposed method is given in the section four. The section five concludes the work.

2. RELATED WORK

In the field of intrusion detection system, there have been a long practice and activities. Tamer Ghanem et. al. [9], proposed network based intrusion detection approach using anomaly techniques which protect networks and systems against unsafe events. The approach proposed is a fusion by means of detectors generated based on multi-start metaheuristic and genetic algorithm. The approach uses certain theory from negative selection-based detector generation. The evaluation of this approach is performed using NSL-KDD dataset.

Kamran Shafi and Hussein A. Abbass [10], presented a biological-inspired computational strategy to learn the signatures dynamically for detection of network intrusion with the help of supervised learning classifier. It checks the population of classifier by using genetic algorithm in which classifier contains the rules, conditions, action taken, output class, number of parameters. The accuracy and fitness are main parameters to be considered. This framework used for automatic and adaptive searching for intrusion detection using supervised learning classifier system. They need to work towards real time IDS with large scale database.

Pedro Casas et al. [11] detected new attack (unknown) without knowledge of any signature, labelled data or training. For that purpose, unsupervised network IDS use outlier detection method which is based on subspace clustering and multiple evidence accumulation method for various network intrusion and attack. For example, DOS/DDOS, probing, unauthorized access to network resources, various traffic dataset. The system consists of: a) Detection of an irregular time slot where clustering analysis is performed. b) Input all the movements in the time slot marked as anomalous. A multi-clustering algorithm is used to identify outlying flows. c) At the top level, movements are anomalies calculated by thresholding detection approach.

The unsupervised IDS are able to detect unknown attacks in an automated manner i.e. zero-day attacks. However, the deployment of IDS is also important in a real time network environment. The several parameters are required for building a process. So, there may be a difficulty for a network manager to tune and optimize the required parameters based on changing behavior of network characteristics. Jungsuk Song et al. [12], presented a more practical unsupervised IDS and evaluated with real traffic data collected from Kytoto University honeypots.

The unsupervised methods provide higher false alarm rate than supervised or semi-supervised approaches. Armin Daneshpazhouh and Ashkan Sami [13], have proposed the semi-supervised outlier detection method. The entropy-based solution has two phases: 1) reliable negative samples are mined from positive and unlabelled data 2) entropy-based outlier detection algorithm is used for detecting top N-outliers.

Shelly Xiaonan et al. [14] presented a paper on intrusion detection based upon computational intelligence (CI). The characteristics of CI systems in terms of noise information are suitable for building a good intrusion detection model.

The SVM method using supervised learning requires pre-defined learning information. This predefined learning process is divided into normal and anomaly labels. One class SVM using unsupervised learning for detecting anomalies has a limitation of a high false positive rate. Therefore, Shon T et al. [1] proposed enhanced SVM which combines unsupervised and supervised learning to reduce false alarms.

The SVM with hierarchical clustering is used for feature selection procedure. The feature selection procedure is applied to reduce unwanted features from training dataset [15]. Chen WH et al. [16] provided the applications of SVM and ANNs for intrusion detection. The ANN and SVM are used with two encoding methods i.e. simple frequency and term frequency–inverse document frequency (TFIDF) to detect intrusions. The SVM with TFIDF performs better compared to ANN with simple frequency based scheme.

The further research work in the field of IDS is feasible by considering the burden of computational cost and to reduce the time complexity.

3. PROPOSED METHOD

This section represents the comprehensive proposed method implemented for detection of intrusion based on signature IDS. The proposed method is an attempt to improve the time and space complexity of the representative method described in the work [8].

3.1 Workflow of Proposed Method

In many intrusion detection systems, a large amount of data requirement is responsible for the high computational cost. The main objective of this paper is to reduce training dataset which is directly used for the classifier. We have used data minimization technique to create small subset from the training dataset. Suppose $X = \{(x_1, D_1)....(x_M, D_M)\}$ be a labeled intrusion detection training data set with *M* training instances, where x_i represents an instance over the *d*-

dimensional feature space, such as $x_i = \{x_i^1, x_i^2, ...x_i^d\}$. Further, $\chi_i = \{x_i^1, x_i^2, ...x_i^M\}$ indicates the set of feature values, for the instance X_i , and $D_i \in \{D_1, ..., D_m\}$ is the corresponding class label for X_i , which belongs to one of the m possible classes. The given training dataset is partitioned using centroid based partitioning to select the high quality subsets from the available large set as a new training set to build a classification model for intrusion detection. Let $X_i^{'}$ denotes the selected i^{th} subset where, $i \in \{1,....,K\}$.

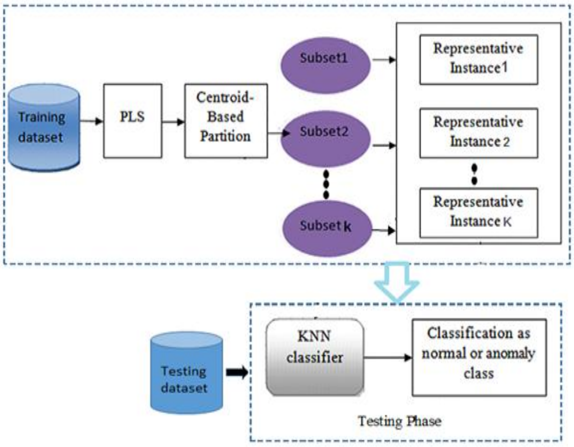

Fig 1: Improved Intrusion Detection System

The concrete steps involved in the creation of representative instances are given below.

1. Training dataset with labelled classes as normal and anomaly.
2. Pre-processing on the training dataset.
3. Apply PLS method to reduce attributes from pre-processed dataset.
4. Creation of subsets based on Centroid-based partition technique.
5. Selection of Top-N representative instances from identified each subset.
6. Selected representative instances act as a training dataset for KNN classifier.
7. Classification of packet (normal or anomaly) based on KNN classifier.

Figure 1 consists of four major stages as: Training Dataset, Data Pre-processing, Classifier Training and Attack recognition. Each stage is precisely described as follows.

3.1.1 Training dataset

In our method, we have used standard KDD Cup 99 training dataset [18]. It contains a huge description of normal and anomaly packets along with 14 additional test sets. A vector contains attributes with information about source bytes, destination bytes, a start flag, protocol, class etc. We found that KDD Cup 99 is more relevant to our research.

3.1.2 Data preprocessing

Considering the selection of relevant attributes, the PLS method help to removeirrelevant or redundant features. The use of PLS is given in section 3.2.1.

3.1.3 Subset creation and classifier training

Once the dataset is pre-processed, the subsets are created using centroid based partitioning method. We also, find the useful and relevant representative instances from each subset. Further, KNN classifier is applied that helps to determine the desired class of the incoming packet. So, the combination of PLS and paradigmatic instances proposed in this method help to discover the class of packet without more computational load. Hence, this will improve attack detection phase. The use of KNN is given in section 3.2.2.

3.1.4 Attack Recognition

This is final stage of the workflow of proposed method. Based on the results of KNN classifier, the packet is labelled as either normal or anomaly.

3.2 Algorithms

The algorithms involved in the proposed system are PLS, centroid calculations and KNN classifier. Each one is described below.

3.2.1 Partial Least Square (PLS)

Let, assume that independent variables $a_1, a_2, a_3, ...a_l$ and dependent variables $b_1, b_2, b_3, ...b_m$ .To check the relation between independent and dependent variables, $A = [a_1, a_2, a_3, ..a_l]_{n \times l}$ and $B = [b_1, b_2, ...b_m]_{n \times m}$. The PLS extracts element k_1- linear combination of $a_1, a_2, a_3, ...a_l$ and r_1 –linear combination of $b_1, b_2, b_3, ...b_m$ .Here, we use standard matrices (C_0 and D_0) to represent independent (A) and dependant variables (B). When first element is extracted, regression model C_0, D_o against k_1 is reformed. After recursive reformation of elements the accuracy is obtained. Therefore, the process terminates once we achieve preferred accuracy. Here, PLS is used for feature extraction technique. The pseudo code of PLS algorithm [17] is as follows.

```
1. For f = 1 to g
2.     $V_f = C_0^T D_0 / \| C_0^T D_0 \|$
3.     $k_f = C_0 V_f$
4.     $h_f = D_0^T k_f / \| k_f \|^2$
5.     $l_f = C_0^T k_f / \| k_f \|^2$
6.     $C_0 = C_0 - k_f l_f^T$
7.     $D_0 = D_0 - k_f h_f^T$
8. End.
```

By using above algorithm, we can fetch g elements, Where, $K = [k_1, k_2, k_3, ..k_g]$ with $V = [v_1, v_2, v_3, ..., v_g]$, $L = [l_1, l_2, l_3, ..., l_g]$.

For selecting the representative instances: Suppose, R_i gives the set of r most similar instances of p_i in the same class Q_i

The term (p_i, Q_i) denotes the training instances in training dataset p where Q_i ,is its class label. Now, specify the training dataset such that, $P = \{(p_1, Q_1), ..., (P_M, Q_M)\}$ of M-instances and $Q_i \in \{Q_1, ..., Q_M\}$. For every training instance (p_i, Q_i), the representativeness of p_i of its class Q_i is defined as:

$$G(p_i, Q_i) = \left(\sum_{j=1, p_j \in R_i}^{r} S(p_j, p_i) \right)^{\gamma} * \left\{ \sum_{P_t = P_1}^{P_m} \sum_{p_h \in P_t} S(p_h, p_i) E_{[Q_h \neq Q_i]} \right\}^{-(1-\gamma)} \quad (1)$$

Where, $\left(\sum_{j=1, p_j \in R_i}^{r} S(p_j, p_i) \right)^{\gamma}$ is a first term in Equation (1) which gives the degree of similarity between instance p_i and its r most similar instances within the same class Q_i

$\left\{ \sum_{P_t = P_1}^{P_m} \sum_{p_h \in P_t} S(p_h, p_i) E_{[Q_h \neq Q_i]} \right\}^{-(1-\gamma)}$ is a second term in Equation (1) which gives the degree of similarity between p_i and the instances in all of the classes except for class Q_i. E[.], is an indicator which is set to 1 if condition is satisfied otherwise 0. γ is a balancing factor in Equation (1) to determine the significant value of first and second term. Accordingly, it set the value in the range of 0 to 1 depending on the specific criteria. The Euclidean distance metric is used to calculate the similarity between two instances $S(p_j, p_i)$ from two vectors [8]. The calculated similarity is inverse to the distance between two instances.

Let, $p_1 = \{p_1^1, p_1^2, p_1^3, ..., p_1^d\}$

$p_2 = \{p_2^1, p_2^2, p_2^3, ..., p_2^d\}$

Thus, the Euclidean distance metric is calculated as follows:

$$dist(p_1, p_2) = \sqrt{(p_1^1 - p_2^2)^2 + ... + (p_1^d - p_2^d)^2} \quad (2)$$

Now, by substituting the above Equation (2) in first term of Equation (1), we can write as:

$$\left(\sum_{j=1, p_j \in R_i}^{r} S(p_j, p_i) \right)^{\gamma} = \left(\sum_{j=1, p_j \in R_i}^{r} dist(p_j, p_i) \right)^{-\gamma} \quad (3)$$

Where, r parameter gives the number of nearest neighbors. The similarity between an instance of cluster and instances in remaining cluster is calculated to write the second term in Equation (1). Since, some classes in a given dataset consists of huge number of instances like denial of service attack or normal classes in KDD CUP 99 datasets and possibility is that it may take more time for calculation.

To solve this problem, utilization of centroid–based classification gives similarity quickly. Thus, to find centroid for a set p_i of instances, its centroid T_i can be defined as its average vector as:

$$T_i = \frac{1}{N_i} \sum_{p \in P_i} p \quad (4)$$

In above Equation (4), N_i gives the number of instances within class Q_i .

3.2.2 K-Nearest Neighbor (KNN)

Consider k as the preferred number of nearby neighbors and $S := p_1, p_2, ...p_n$ be the set of training samples in the form of $p_1 = (X_i, C_i)$,where X_i is the dimensional feature vector of the point p_i and c_i is the class that p_i belongs for each. The working of KNN in the context of problem mentioned in this paper is as below:

- Compute the distance between p and all p_i belonging to S.
- Categorize all points p_i according to the key.
- Choose the first k points from the arranged list; those are the k nearby training samples to p

Assign a class to p based on a popular poll.

4. RESULTS AND DISCUSSION

This section discusses on results obtained from the implementation of proposed method. This includes a description of used dataset, scenarios of embedding attack, its types, and comparison of results obtained by earlier research and our proposed method.

For experimental setup, we have used Windows 7 OS, Intel i5processor, 2 GB RAM, 500GB Hard disk, Net Beans IDE 8 + JDK tool. To calculate the results, KDD Cup 99 dataset is used. In training dataset, there are 23 types of attackand in testing phase additional 14 attacks are included. Using this dataset, we look fordetection of time and memory space with PLS and without PLS.

4.1 Dataset

KDD Cup 99 dataset consists of comparatively 4,900,000 single association vectors wherever every single connection vectors consists of 41 options and is marked as either traditional or associate attack, with specifically one explicit attack type [18].

The dataset is segregated into following types of attacks: Denial of Service (DoS), Probing, user to root (U2R) and remote to local (R2L).

4.1.1 DoS Attack

This attack causes a network or machine resources not available to its knowing user which means it makes an interrupt to service or suspends some services to the host connected to the internet. The examples of DoS are attacks on web servers, i.e. banks, credit card payment gateways.

4.1.2 Probing

It includes a device which is inserted in a key juncture for monitoring or collecting data about network activity and gain access tocomputer.

4.1.3 User to Root (U2R)

The attacker tries to access administrator privilege levels by making use of some vulnerability in the victim. The example is buffer overflow attacks.

4.1.4 Remote to Local (R2L)

This attack may cause unauthorized access to the remote machine and gives local access to the victim machine for example password guessing.

4.2 Time and Memory Space

Before applying KNN, we reduced training dataset by using PLS. The time taken with PLS in training is much less than the time taken without PLS.

The Table 1 indicates the comparison for detection time by using representativeness and PLS. It is observed that for dataset size from 1000 to 15000, the detection time required is less by using PLS algorithm. Reduction in the size of training dataset using PLS is key parameter to reduce the time of computation.

Table 1. Detection Time using Representativeness and PLS

Dataset Size	Detection Time With Representativeness (ms)	Detection Time with PLS (ms)
1000	110	10
3000	106	8
5000	114	7
8000	116	7
10000	117	6
15000	118	6

Figure 2 shows the detection time where the representativeness take more time than PLS. The PLS method selects the number of attributes that are required indetection of intrusion. The important features are extracted and applied for detection of intrusions. Because of this, the time required for detection is reduced using PLS. The representative method selects the representative from original dataset for each of class and then these instances are used for intrusion detection.

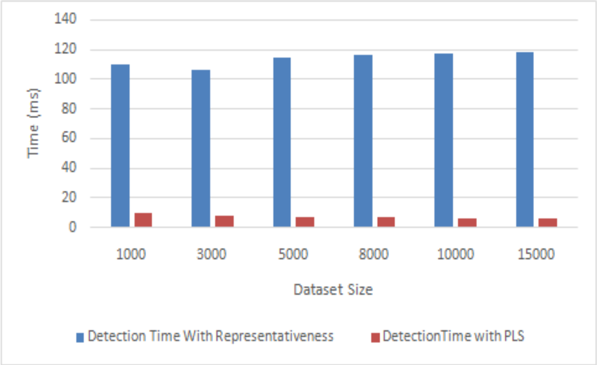

Fig 2: Detection time using representativeness and PLS

Similarly, the comparison of memory space for Representativeness and PLS method is shown in Table 2.

Table 2. Memory space with Representativeness and PLS

Dataset Size	Memory Space with Representativeness (Mb)	Memory Space with PLS (Mb)
1000	13.572	13.073
3000	24.158	20.612
5000	32.528	25.166
8000	42.492	30.017
10000	44.15	32.746
15000	46.251	33.878

Figure 3 shows comparison of memory space for Representativeness and PLS.

We have considered dataset size ranges from 1000 to 15000. The observation shows that, the memory space required is less by using PLS algorithm than representativeness.

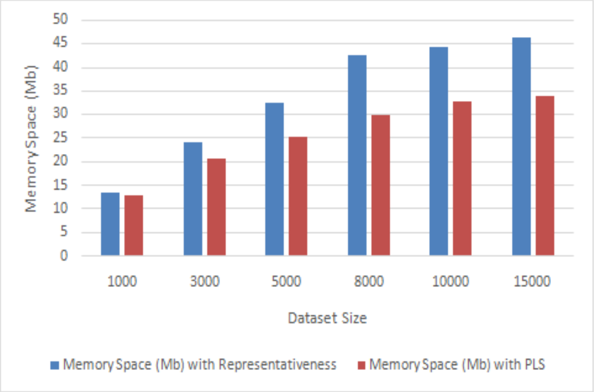

Fig 3: Memory space using representativeness and PLS

5. CONCLUSION

The IDS framework that uses extensive size of dataset with maximum number of attributes results additional computational overhead. The numbers of attributes are reduced efficiently using PLS that focus on relevant and desired attributes. Hence, the proposed method and its results show the reduction in the time and space complexity. The proposed work presented efficient method to improve the performance of representativeness which is based on the subsets. An important feature of the proposed method in this paper is an effective creation of subset based on centroid allotment. The PLS method reduces the number of attributes that are irrelevant in detection of intrusion. Hence, the complexity is found to be reduced because of utilization of PLS before classification. The arrangement of paradigmatic occurrences used for preparing dataset which is utilized as a part of KNN classifier. The important features are extracted and applied for detection of intrusions. Because of this, the time required for detection is reduced using PLS. The observation is that, time required for PLS is less than representativeness. This method is based on non-real time dataset. This method can be improved in future for real time datasets using other techniques.

6. ACKNOWLEDGMENTS

Our sincere thank to the college authorities for providing the required resources.

7. REFERENCES

[1] Shon T, Moon J, "A hybrid machine learning approach to network anomaly detection," Information Sciences 177(2007):3799-3821.

[2] Bouzida Y, Cuppens F, Cuppens-Boulahia N, Gombault S, "Efficient intrusion detection using principal component analysis,"Proceedings of the 3eme Conference surla Scurit et Architectures Rseaux (SAR) 2004.

[3] Pavel Laskov, Patrick Dussel, Christin Schafer and Konrad Rieck, "Learning intrusion detection: supervised or unsupervised?,"iciap, 2005.

[4] Garcia-Teodoro P, Diaz-Verdejo J, Macia -Fernandez G, Vazquez E., "Anomaly-based network intrusion detection: techniques, systems and challenges," Computers & Security 28(2009):18-28.

[5] Tsai CF, Hsu YF, Lin CY, Lin WY, "Intrusion detection by machine learning: a review," Expert Systems with Applications 36(2009):11994-12000.

[6] Su MY, "Using clustering to improve the KNN-based classifiers for online anomaly network traffic identification," Journal of Network and Computer Application 34 (2011):722-730.

[7] Davis JJ, Clark AJ, " Datapreprocessing for anomaly based network intrusion detection: a review," Computers & Security 30(2011):353-375.

[8] Chun Guo, Ya-Jian Zhou, Yuan Ping, Shou-Shan Luo, Yu-Ping Lai, Zhong-Kun Zhang, "Efficient intrusion detection using representative instances," Computers & Security 39 (2013): 255 -267.

[9] Tamer F. Ghanem, Wail S. Elkilani, Hatem M. Abdul-kader, "A hybrid approach for efficient anomaly detection using metaheuristic methods," Journal of Advanced Research (2015) 6, 609-619.

[10] Kamran Shafi, Hussein A. Abbass, "An adaptive genetic-based signature learning system for intrusion detection," Expert Systems with Applications 36 (2009) 12036–12043.

[11] Pedro Casas, Johan Mazel, Philippe Owezarski, "Unsupervised Network Intrusion Detection Systems: Detecting the Unknown without Knowledge," Computer Communications 35 (2012) 772–783.

[12] Jungsuk Song, Hiroki Takakura, Yasuo Okabe, Koji Nakao, "Toward a more practical unsupervised anomaly detection system," Information Sciences 231 (2013) 4–14.

[13] Armin Daneshpazhouh, Ashkan Sami, "Entropy-based outlier detection using semi-supervised approach with few positive examples," Pattern Recognition Letters 49 (2014) 77–84.

[14] Shelly Xiaonan Wu, Wolfgang Banzhaf, "The use of computational intelligence in intrusion detection systems: A review," Applied Soft Computing 10 (2010) 1–35.

[15] Shi-Jinn Horng, Ming-Yang Su, Yuan-Hsin Chen, Tzong-Wann Kao, Rong-Jian Chen, Jui-Lin Lai, Citra Dwi Perkasa, " A novel intrusion detection system based on hierarchical clustering and support vector machines," Expert Systems with Applications 38 (2011) 306–313.

[16] Chen WH, Hsu SH, Shen HP, "Application of SVM and ANN for intrusion detection," Computer Operation Research 2005; 32(10):2617-2634.

[17] GanXu-sheng, Duanmu JS, Wang JF, Cong Wei, "Anomaly intrusion detection based on PLS feature extraction and core vector machine," Knowledge-based Systems 40 (2013)1-6.

[18] The KDD Cup 99 dataset is available at http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html