Intrusion Detection System Based on Cost Based Support Vector Machine

Md. Rafiul Hassan

Abstract In this paper, a novel intrusion detection system (IDS) is developed using a cost based support vector machine (SVM). While developing an IDS, due to the imbalanced characteristics it is very difficult to differentiate the attack events from a non-attack (normal) event in any network environment. The cost based SVM facilitates to put much weight to one pattern over another ones to differentiate attack and non-attack cases with a high accuracy. The same can be applied on a multiclass attack problems by using cost factor to each ratio of different types of attacks. In this study, the cost based SVM has been applied to classify DARPA99 intrusion detection dataset. The experimental results show that the cost based SVM can outperform standard SVM while attempting to differentiate a case as either attack or non-attack (normal). Furthermore, we applied the cost based SVM with an RBF kernel to a multiclass attack problem. Experimental result achieved about 99 % detection accuracy when it was applied to detect the type of attacks as either of Normal, DOS, Probe and R2L from DARPA99 dataset.

Keywords IDS · Cost based SVM · Imbalanced data

1 Introduction

Intrusion Detection is very much essential these days to protect information systems security, especially in the view of worldwide increasing incidents of cyber attacks. Identification of unauthorized use, misuse and attacks on information system is defined as intrusion detection. It is needed because traditional firewalls can't provide full protection against security breaches. An Intrusion Detection System (IDS) doesn't prevent an intrusion, it only detects it and informs the operator.

Md.R. Hassan (✉)
Department of Information and Computer Science, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
e-mail: mrhassan@kfupm.edu.sa

P. Meesad et al. (eds.), *Recent Advances in Information and Communication Technology 2016*, Advances in Intelligent Systems and Computing 463,
DOI 10.1007/978-3-319-40415-8_11

It detects a hacker breaking into the system or a genuine user exploiting the system resources.

Primary measurement criteria for an IDS are as follows

- False Positives i.e. an event being incorrectly identified as an intrusion when none has occurred.
- False Negatives i.e. an event which IDS fails to identify as an intrusion when it really occurs.
- True positive i.e. an event being correctly classified as an intrusion when one has occurred.
- True Negative i.e. an event being not classified as an intrusion when none has occured.
- Accuracy i.e. how efficient the IDS is in detecting intrusions when it has really occurred.

It is essential to analyze the audit data (generated by the operating systems and networks) in order to estimate the extent of damage occurred, specially in attack trace and listing the attack pattern for future prevention. This makes an IDS a real time detection and prevention tool as well as forensic analysis tool [1].

Artificial intelligence techniques have been widely used by the IDS researchers worldwide due to their generalization capability that help in identifying known intrusions as well as unknown intrusions. Neural Networks have been used to identify both misuse and anomalous patterns [2–4]. Support vector machines (SVMs) have been emerged as a new and powerful technique for learning from data and in particular for solving classification and regression problems. It has been also proved that SVM yields very good result in different domains for the purpose of classification.

SVMs and their variants have been proposed and used by a number of studies for detection of intrusion. Xiao et al. [5] proposed a combined technique of Ad hoc technology and SVM for effective detection of intrusion. Yendrapalli et al. [6] also applied a biased SVM (BSVM) as an IDS tool on DARPA dataset. Recently, Aburomman and Reaz [7] used an ensemble of SVM with other optimization (Particle Swarm Optimization) and clustering method (k-nn) to detect intrusion. A brief discussion about application of SVM as IDS is provided in Sect. 2.

In this paper our aim is to use a cost-based SVM for intrusion detection. Even though the previous studies have evaluated performance of SVM in detecting intrusion, no study has yet explored the efficacy of cost-based SVM to detect intrusion. The available bench mark data in intrusion detection is highly imbalanced in the sense that number of intrusion samples highly outnumber the number of normal samples indicating imbalance in available data. This motivates the research in this paper to use different cost for different class justifying the use of cost-based SVM as an IDS.

The paper is organized as follows. Section 2, a brief literature review is provided. The learning theory of SVM and cost based SVM is written in Sect. 3. Experimental setup, results and analysis is discussed in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Literature Review

A literature review on application of SVM for intrusion detection has been provided by Kausar et al. [8]. Following the literature review it is found that, Xiao et al. [5] proposed a combined technique of Ad hoc technology and SVM for effective detection of intrusion. They identified an enhanced performance of IDS through selecting feature subset and then optimizing the SVM parameters. A Gaussian Kernel SVM could produce a better performance compared with other kernels. They evaluated their technique on DARPA 1998 dataset which has four different attacks namely DOS, R2L, U2R and probe.

Yendrapalli et al. [6] applied a biased SVM (BSVM) as an IDS tool on DARPA dataset. The experimental result using leave-one-out validation technique showed that the performance of IDS for differentiating between normal and u2r attacks is better achieved through using a standard SVM while that of between probe and R2L is better for BSVM. They also concluded that, the performance of SVMs as IDS depends on suitable choice of the SVM parameters.

Yuancheng et al. [9] used SVM followed by selection of features using Kernel Independent Component Analysis. They validated their approach on KDDCUP99 dataset and the experimental results showed a very promising performance of intrusion detection. The resultant SVM also could classify new types of attack which was not included in the training dataset. The ultimate aim of that study was to decrease false alarm with a penalty of overall classification performance.

Gao et al. [10] optimized the SVM parameters and applied the SVM as an IDS. The found SVM very time efficient in detecting intrusion types and also the generalization ability of it. The experimental result proved a better performance by SVM compared with that of RBFNN (Radial Basis Function Neural Network).

Rung-Ching et al. [11] used rough set theory (RST) to deselect less influential features from the dataset and then applied SVM to classify the type of attacks. They evaluated the approach on KDDCUP99 dataset and achieved a higher accuracy in terms of false positive rate and attack detection rate compared with that of an entropy based feature selections.

Yuan et al. [12] proposed the application of hypothesis test theory to SVM classifier (HTSVM) as an IDS. The hypothesis test theory was adopted to decrease the impact of penalty factor in SVM and thereby the overall performance of SVM was improved. The experimental results of using HTSVM on KDDCUP99 dataset showed a better intrusion detection performance compared with that of C-SVM. The results also showed a very good generalization ability of HTSVM classifier.

Guan et al. [13] used the concept of agent along with SVM as an IDS. Each agent involved one SVM to classify two different type of attacks. Finally, a majority voting approach was applied to decide about the type of attack. The experimental results on KDDCUP99 showed a better detection accuracy when compared with the performance of artificial neural networks.

Xiaomei et al. [14] proposed an adaptive genetic algorithm (AGA) to obtain an optimal penalty factor for SVM. Then the SVM was used to analyze audit and detect attack type. Their experimental results on KDDCUP99 dataset revealed the applicability of SVM as IDS.

3 Support Vector Machine

Support Vector Machines introduced by Vapnik [15] have been widely used for applications in many classification and regression problems. The underlying theory of SVM finds a hyperplane which is optimal in separating data in either of two classes. We refer to this hyperplane as optimal separating hyperplane (OSH). Further to this, the kernel trick of transforming data into a higher dimensional space makes SVM an efficient classification tool. Thus, an OSH generation becomes easier in this transformed feature space. The data vectors that lie closed to the OSH are called as support vectors (SV). Since these SVs determine the OSH, they are very useful in classifying data [16].

Let us, consider a training set $D = \{(\mathbf{x}_i , y_i)\}_{i=1}^{L}$, where x_i represents ith input $\mathbf{x}_i \in \Re^n$ and the associated class label is $y_i \in \{-1, +1\}$. In order to search for an OSH in SVM, every input pattern $\mathbf{x}$ is first mapped into a higher dimension feature space $\mathcal{F}$ by $\mathbf{z} = \phi(\mathbf{x})$; where, $\phi(x)$ is a non-linear mapping function as ϕ: $\Re^n \rightarrow \mathcal{F}$. In this case the assumption is that, data are not separable using a linear boundary in real feature space and data in the transformed feature space is linearly separable. Thus, there exists a vector $\mathbf{w} \in \mathcal{F}$ and a scalar value b such that the separating hyperplane is:

$$\begin{aligned} &\mathbf{w} \cdot \mathbf{z} + b = 0 \text{ and} \\ &y_i(\mathbf{w} \cdot \mathbf{z}_i + b) \geq 1 - \xi_i, \quad \forall i \end{aligned} \tag{1}$$

where ξ_i ($\geq$ 0) are referred as *slack variables*. The significance of ξ_i is that, it yields to the misclassified data patterns. Thereby, the term $\sum_{i=1}^{L} \xi_i$ is the measure of misclassification during OSH generation. The ultimate aim of OSH generation is to achieve a maximum classification accuracy and minimum training error. The condition of such optimal hyperplane generation considering data $\mathcal{F}$ is:

$$\begin{aligned} &\textit{minimize}: && \frac{1}{2}\mathbf{w}\cdot\mathbf{w} + C\sum_{i=1}^{L}\xi_i \\ &\textit{subject to}: && y_i(\mathbf{w}\cdot\mathbf{z}_i + b) \geq 1-\xi_i, \text{and} \quad \xi_i \geq 0, \forall i \end{aligned} \tag{2}$$

here C is a constant parameter known as *regularization parameter*. This parameter represents a trade off measurement between the maximum margin and minimum classification error.

The solution of Eq. (2) is a Quadratic Programming (QP) problem. In the process of solving Eq. (2) first a primal Lagrangian transformation is formed and then the primal Lagrangian is transformed into a dual.

From the primal and dual form of Lagrangian the following optimal hyperplane is obtained:

$$\begin{aligned} &\textit{maximize}: && W(\alpha) = \sum_{i=1}^{L}\alpha_i - \frac{1}{2}\sum_{j=1}^{L}\sum_{j=1}^{L}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ &\textit{subject to}: && \sum_{i=1}^{L} y_i\alpha_i = 0 \quad \text{and} \quad 0 \leq \alpha_i \leq C, \quad \forall i. \end{aligned} \tag{3}$$

where $\alpha_1, \alpha_2, \ldots, \alpha_L$ are the non-negative Lagrangian multipliers. The data points $\mathbf{x}_i$ corresponding to $\alpha_i > 0$ lie along the margins of decision boundary and are the SVs. The kernel function $K(.,.)$ is an inner product ($K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \bullet \phi(\mathbf{x}_j) = \mathbf{z}_i \bullet \mathbf{z}_j$) among pairwise input patterns. One condition of kernel function is that it must satisfy the Mercer's condition [13]. Through determination of the optimum Lagrangian multipliers, optimum solution for weight vector $\mathbf{w}$ is obtained as

$$\mathbf{w} = \sum_{i \in SVs} \alpha_i y_i \mathbf{z}_i \tag{4}$$

where SVs are the support vectors. For any unknown data vector $\mathbf{x} \in \Re^n$, the classification is done by

$$y = f(\mathbf{x}) = sign(\mathbf{w}\cdot\mathbf{z} + b) = sign\left(\sum_{i \in SVs}\alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right) \tag{5}$$

In the process of SVM classifier training, one must tune C and choose a suitable kernel function with its parameters. Since, no theory is available about how to choose the best C and kernels, the performance of SVMs for a problem may vary with this choice. Table 1 lists few different types of kernels used in SVMs.

When the number of samples in two classes (positive and negative classes) are grossly unequal, the classification data is regarded as imbalanced. A technique has been proposed by Morik et al. [17] to deal with imbalanced data for SVM learning using different costs (C_+ and C_- instead of single C in Eq. (2)) for each class.

Lacking data for designing a more refined cost model, the cost-factors are chosen so that the potential total cost of the false positives equals the potential total cost of

Table 1 Types of kernel functions in SVM

Kernel function	Mathematical formula
Linear	$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$
Polynomial	$K(\mathbf{x}_i, \mathbf{x}_j) = \left(\langle \mathbf{x}_i, \mathbf{x}_j \rangle + 1\right)^d$, d: degree of polynomial
Radial Basis Function (RBF)	$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2 / 2\sigma^2\right)$, σ: width of RBF function
Spline (ANOVA)	$K(\mathbf{x}_i, \mathbf{x}_j) = 1 + \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \frac{1}{2}\langle \mathbf{x}_i, \mathbf{x}_j \rangle \min\left(\langle \mathbf{x}_i, \mathbf{x}_j \rangle\right) - \frac{1}{6}\min\left(\langle \mathbf{x}_i, \mathbf{x}_j \rangle\right)^3$

the false negatives. This means that the parameters C_+ and C_- of the SVM are chosen to follow the ratio in Eq. (6).

$$\frac{C_+}{C_-} = \frac{number\ of\ positive\ training\ examples}{number\ of\ negative\ examples} \tag{6}$$

The quantity $\frac{C_+}{C_-}$ is expressed by a quantity j. Considering the above stated equation, Eq. (2) is reformulated as

$$\begin{aligned} &minimize: && \frac{1}{2}\boldsymbol{w}\cdot\boldsymbol{w} + C_+ \sum_{i=positive\ class} \xi_i + C_- \sum_{\{j=negative\ class\}} \xi_j \\ &subject\ to: && y_i(\boldsymbol{w}\cdot\boldsymbol{z_i} + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0, \forall i \\ & && y_i\left(\boldsymbol{w}\cdot\boldsymbol{z_j} + b\right) \geq 1 - \xi_j \text{ and } \xi_j \geq 0, \forall j \end{aligned} \tag{7}$$

4 Experimental Setup and Results

4.1 Dataset

In this paper we used DARPA99 dataset to evaluate the cost based SVM as an IDS. To generate the dataset an environment was set up to acquire raw TCP/IP dump data for a network by simulating a typical US Air Force LAN. The LAN was operated like a real environment, but being blasted with multiple attacks. A connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows to and from a source IP address to a target IP address under some well defined protocol. Each connection is labeled as either normal, or as an attack, with exactly one specific attack type. Each connection record consists of about 100 bytes. For each TCP/IP connection, 41 various quantitative and qualitative features were extracted. Of this database a subset of 494,021 data were used, of which 20 % represent normal patterns and the rest 80 % are attack data.

4.2 Data Preparation

In this experiment, 15,000 random samples have been chosen from the subset of 494,021 samples. The experiment aims to identify intrusion differentiating it from the normal pattern while the second part of the experiment aims to identify different types of intrusion along with the normal pattern. So the first part is essentially a binary classification problem while the second part is a multiclass problem. For the first part we have defined intrusion as +1 (positive symbolizes attack or intrusion) and normal pattern as −1. For the second part we have defined normal pattern as 1, DoS as 2, Probe as 3, R2L as 4, and U2Su as 5. So it symbolizes a multiclass problem.

The following 10 features (urgent, root_shell, su_attempted, num_root, num_file_creations, num_shells, num_access_files, num_outbound_cmds, is_host _login, is_guest_login) have been deleted from the 15,000 random samples because the value of each feature is zero and hence these features doesn't contribute anything to the variation factor. The rest 31 features were finally used for the intrusion detection in the experiment. Out of 15,000 random samples—3000 were normal pattern, 11,850 were DoS, 120 were probe, 29 were R2L and 1 was U2Su. This was done in accordance to their proportion in the original subset of 494,021 samples. In the last phase of data preparation, the data set was divided into 5 equal sets each consisting of 3000 samples for a fivefold cross validation scheme.

4.3 Performance Measures

The following three performance measures (accuracy, sensitivity and specificity) were used to assess the performance of the SVM as an IDS.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \times 100\,\% \tag{8}$$

$$Sensitivity = \frac{TP}{TP + FN} \times 100\,\% \tag{9}$$

$$Specificity = \frac{TN}{TN + FP} \times 100\,\% \tag{10}$$

where TP is the number of true positives, i.e. the classifier identifies an intrusion that was labeled as intrusion; TN is the number of true negatives, i.e. classifier identifies a normal pattern that was labeled as normal; FP is the number of false intrusion identification; and FN is the number of false normal identification. Accuracy indicates overall detection accuracy for both normal and intrusion patterns, sensitivity is defined as the ability of the classifier to accurately recognize a

intrusion pattern whereas specificity would indicate the classifier's ability not to generate a false detection.

In addition to the above measures, classifier's performance was also evaluated in terms of receiver operating characteristic (ROC). ROC curve plots sensitivity against (1-specificity) as the threshold level of the classifier is varied and depicts the performance of a classifier without regard to class distribution. The area under the ROC curve (AUC) summarizes the quality of classification and is used as a single measure of accuracy.

4.4 Experimental Results and Analysis

4.4.1 Intrusion Detection Between Two Classes: Normal Versus Attack

This section provides the results of two-class (intrusion and normal) detection problem. Two kernel functions—Linear and RBF were used.

For Linear kernel, different values of j (j is the cost factor as defined in Eq. (6)) were used to get different results. For each value of j, fivefold cross validation was performed and then the final value was given by averaging the results from different folds. Table 2 shows the IDS performance for linear kernel cost based SVM with varying values of j.

Table 2 clearly depicts that both accuracy and ROC increase with the increase in value of j till j = 2.0. After that they start to drop, within the region studied. When j = 1.0 i.e. equal emphasis is given both to negative class and positive class (may be referred as standard SVM) the accuracy and ROC are 97.94 % and 0.995 respectively. When the value of j is increased to 2.0 i.e. number of negative training examples (i.e. intrusion pattern) is given double the emphasis as compared to number of positive training examples (i.e. not intrusion), both accuracy and ROC improve.

Table 2 IDS performance using linear kernel cost based SVM

Linear kernel function				
j	Accuracy (%)	Sensitivity (%)	Specificity (%)	ROC
5.0	95.41	99.95	79.41	0.992
3.0	96.72	99.95	87.52	0.993
2.0	**98.19**	**98.93**	**94.91**	**0.996**
1.0	97.94	97.47	98.85	0.995
0.5	97.85	97.32	98.97	0.995
0.25	79.75	77.40	94.36	0.848

The cell with the bold highlights the case where the accuracy as well as the ROC is the highest

We also used an RBF kernel to the cost based SVM as an IDS. In this case, for each value of j, fivefold cross validation was performed for a varying values of σ and then the final value was given by averaging the results from different folds. Table 3 summarizes the performance IDS while the classifier used is an RBF kernel with cost based SVM. the table clearly depict that higher value of j and lower value of σ (i.e. when the width of the kernel decreases and it becomes more non-linear) give a better measure of accuracy, ROC and sensitivity.

It is evident from Table 3 that accuracy, sensitivity and ROC increase with the increase in value of j within the region studied. When j = 1.0 (standard SVM) i.e. equal emphasis is given both to negative class and positive class the accuracy and ROC are 95.80 % and 0.975 respectively. When the value of j is increased to 5.0 i.e. number of negative training examples (i.e. intrusion pattern) is given five times emphasis as compared to number of positive training examples (i.e. not intrusion),

Table 3 IDS performance summary for RBF kernel cost based SVM

RBF kernel function					
J	σ	Accuracy (%)	Sensitivity (%)	Specificity (%)	ROC
5.0	1.0	95.54	94.02	100	0.967
	0.5	95.75	94.28	100	0.973
	0.25	95.98	94.54	100	0.979
	0.1	96.14	94.75	100	0.988
	0.001	**98.21**	**97.40**	**100**	**0.996**
3.0	1.0	95.53	94.00	100	0.967
	0.5	95.75	94.28	100	0.973
	0.25	95.97	94.53	100	0.979
	0.1	96.08	94.71	100	0.987
	0.001	98.17	97.29	100	0.996
2.0	1.0	95.46	93.9	100	0.963
	0.5	95.66	94.15	100	0.97
	0.25	95.79	94.30	100	0.975
	0.1	95.97	94.56	100	0.979
	0.001	98.08	97.14	100	0.995
1.0	1.0	95.33	93.72	100	0.959
	0.5	95.45	93.88	100	0.963
	0.25	95.65	94.14	100	0.97
	0.1	95.80	94.31	100	0.975
	0.001	97.97	97.01	100	0.994
0.5	1.0	95.22	93.54	100	0.951
	0.5	95.27	93.6	100	0.955
	0.25	95.33	93.68	100	0.959
	0.1	95.64	94.11	100	0.97
	0.001	97.91	96.8	100	0.994

The cell with the bold highlights the case where the accuracy as well as the ROC is the highest

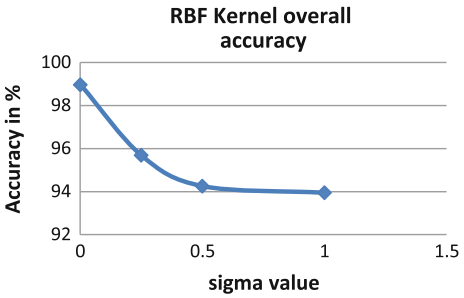

Fig. 1 Overall accuracy RBF kernel with varying σ

both accuracy and ROC improve by a good extent. The improvement in accuracy is almost by 2.5 %, as compared to when equal emphasis is given to both positive and negative classes.

As compared to Linear kernel, there is an improvement in both accuracy and ROC by using RBF kernel. However specificity i.e. the ability of the classifier to identify a normal pattern as normal only increases significantly to 100 % as compared to Linear kernel.

4.4.2 Intrusion Detection Among Multiple Classes

We applied the RBF kernel cost based SVM (as mentioned above RBF kernel provided better performance compared to a linear kernel) to classify the dataset into four attack types: Normal, DOS, Probe, R2L. Figure 1 summarizes the intrusion detection accuracy for $j = 2$ and varying values of σ for the considered data samples.

As shown in the graph, we notice that for a small value of σ and $j = 2$, the performance of RBF kernel with cost based SVM reaches up to 99 %. However, for an increased value of σ, the performance of intrusion detection is not as good as 99 %. This result evidently suggest the importance of choosing SVM parameters for a successful application of SVM as an IDS. Nonetheless, the high detection accuracy and ROC area also encourages the application of cost based SVM for detecting attacks in network environment.

5 Conclusion

In this paper, we proposed the application of a cost based SVM to detect intrusion in a network environment. The experimental results also revealed that, a cost based SVM (i.e. cost factor $j \geq 2$) can outperform the standard SVM (i.e. $j = 1$) while attempting to differentiate whether an event is attack or non-attack (normal). Having proven the efficacy of cost based SVM with RBF kernel, the same method was

applied to detect the type of attacks (i.e. Normal, DOS, Probe, R2L) and the experimental results showed an overall accuracy is about 99 %. These high accuracies in attack detection certainly establishes the applicability of cost based SVM as an IDS.

References

1. Mukkamala, S., Sung, A.H., Abraham, A.: Intrusion detection using an ensemble of intelligent paradigms. J. Netw. Comput. Appl. 168–179 (2004)
2. Debar, H., Dorizzi, B.: An application of a recurrent network to an Intrusion detection system. In: Proceedings of the International Joint Conference on Neural Networks, pp. 78–83 (1992)
3. Ryan, J., Lin, M.-J., Miikkulainen, R.: Intrusion detection with neural networks. Advances in Neural Information Processing Systems, pp. 78–83. MIT Press (1997)
4. Mukkamala, S., Janoski, G., Sung, A.H.: Intrusion detection using neural networks and support vector machines. In: Proceedings of IJCNN, pp. 1702–1707 (2002)
5. Xiao, H., Peng, F., Wang, L., Li, H.: Ad hoc-based feature selection and support vector machine classifier for intrusion detection. In: IEEE International Conference on Grey Systems and Intelligent Services (GSIS 2007), pp. 1117–1121 (2007)
6. Yendrapalli, K., Mukkamala, S., Sung, A.H., Ribeiro, B.: Biased support vector machines and kernel methods for intrusion detection. In: Proceedings of the World Congress on Engineering (WCE) 2007, London, U.K (2007)
7. Aboromman, A.A., Reaz, M.B.I.: A novel SVM-kNN-PSO ensemble method for intrusion detection system. Appl. Soft Comput. **38**, 360–372 (2016)
8. Kausar, N., Samir, B.B., Abdullah, A., Ahmad, I., Hussain, M.: A review of classification approaches using support vector machine in intrusion detection. Commun. Comput. Inf. Sci. 1–11 (2016)
9. Yuancheng, L., Zhongqiang, W., Yinglong, M.: An intrusion detection method based on KICA and SVM. In: 7th World Congress on Intelligent Control and Automation (WCICA 2008), pp. 2141–2144 (2008)
10. Gao, M., Tian, J., Xia, M.: Intrusion detection method based on classify support vector machine. In: Proceedings of the 2009 Second International Conference on Intelligent Computation Technology and Automation, pp. 391–394 (2009)
11. Rung-Ching, C., Kai-Fan, C., Ying-Hao, C., Chia-Fen, H.: Using rough set and support vector machine for network intrusion detection system. In: First Asian Conference on Intelligent Information and Database Systems (ACIIDS 2009), pp. 465–470 (2009)
12. Yuan, J., Li, H., Ding, S., Cao, L.: Intrusion detection model based on improved support vector machine. In: Proceedings of the 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, pp. 465–469 (2010)
13. Guan, X., Guo, H., Chen, L.: Network intrusion detection method based on Agent and SVM. In: The 2nd IEEE International Conference on Information Management and Engineering (ICIME), pp. 399–402 (2010)
14. Xiaomei, Y., Peng, W.: Security audit system using adaptive genetic algorithm and support vector machine. In: 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), pp. 265–268 (2010)
15. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, NY (1995)
16. Haykin, S.: Neural Networks—A Comprehensive Foundation. Upper Saddle River (2004)
17. Morik, K., Brockhausen, P., Joachims, T.: Combining statistical learning with a knowledge-based approach-a case study in intensive care monitoring. In: Proceedings of ICML, pp. 268–277 (1999)