A Real-time Intrusion Detection System Based on PSO-SVM

Jun Wang, Xu Hong, Rong-rong Ren, Tai-hang Li
Northeastern University at Qinhuangdao, Qinhuangdao, PRC
Email: wjhit@sina.com
Email: renrr@mail.neuq.edu.cn

Abstract—The success of any Intrusion Detection System (IDS) is a complicated problem due to its nonlinearity and the quantitative or qualitative network traffic data stream with irrelevant and redundant features. How to choose the effective and key features to IDS is very important topic in information security. Support vector machine (SVM) has been employed to provide potential solutions for the IDS problem. However, the practicability of SVM is affected due to the difficulty of selecting appropriate SVM parameters. Particle swarm optimization (PSO) is an optimization method, which is not only has strong global search capability, but also is very easy to implement. Thus, the proposed PSO-SVM model is applied to an intrusion detection problem, the KDD Cup 99 data set. The standard PSO is used to determine free parameters of support vector machine and the binary PSO is to obtain the optimum feature subset at building intrusion detection system. The experimental results indicate that the PSO-SVM method can achieve higher detection rate than regular SVM algorithms in the same time.

Index Terms—Intrusion detection system, Support vector machines (SVM), PSO, Feature selection

I. INTRODUCTION

Traditional security policies or firewalls have difficulty in preventing such attacks because of the hidden vulnerabilities contained in software applications. Therefore, intrusion detection system (IDS) is required as an additional wall for protecting systems despite the prevention techniques. Support vector machine (SVM) is the method that is receiving increasing attention with remarkable results for the design of IDS recently. Unfortunately, the determination of parameters values becomes an optimization problem in the practicability of SVM. IDS is always to deal with huge amount of data causing slow training and testing process and low detection rate. So feature selection is one of the key topics in IDS[1]. Thus a new technology: A Global optimal search performance of particle swarm optimization (PSO) is used to optimize the SVM model parameters and feature selection for IDS from the KDD Cup 99 data sets[2]. The method is very easy to implement and there are few parameters to adjust[3].

This paper is organized as follows: Section II introduces the regression arithmetic of SVM. Parameters selection of SVM and the feature selection for IDS based on PSO is introduced in Section III. Section IV testifies the performance of the proposed model with the data sets from KDD Cup (1999) intrusion detection dataset. Finally, the conclusion is presented in Section V.

II. SUPPORT VECTOR MACHINES (SVM)

Support vector machine is put forward by Vapnik (1995), it becomes a popular method in machine learning area. The basic concept of SVM regression is to map nonlinearly the original data x into a high-dimensional feature space, and to solve a linear regression problem in this feature space. Let \( \{(x_i, y_i)\}_{i=1}^m \in X \times \{\pm 1\} \). Where \( x \) denotes the input vector, \( y \) denotes the corresponding output value and \( m \) denotes the total number of data patterns, the SVM regression function is:

\[
\hat{y}(x) = w \cdot x + b
\]

Where \( w \) denotes the weight vector and \( b \) denotes the bias term. The coefficients \( w \) and \( b \) are estimated by minimizing the following regularized risk function:

\[
R(C) = \frac{1}{2} ||w||^2 + C \frac{1}{m} \sum_{i=1}^{m} L(y_i, f(x_i))
\]

Where \( C \) denotes a cost function measuring the empirical risk. \( \frac{1}{2} ||w||^2 \) is the regularization term.

\( L(y_i, f(x_i)) \) is called the e-insensitve loss function, which is defined as:

\[
L(y_i, f(x_i)) = \begin{cases} 
|y_i - f(x_i)| - e & |y_i - f(x_i)| \geq e \\
0 & |y_i - f(x_i)| < e 
\end{cases}
\]

In (2), the loss equals zero if the error of forecasting value is less than \( e \), otherwise the loss equals value beyond \( e \). Two positive slack variables \( \xi \) and \( \xi^* \) are introduced to represent the distance from actual values to the corresponding boundary values of the \( e \)-tube. Then, \( R(C) \) is transformed into the following constrained form:

\[
\min_{\phi(w, \xi, \xi^*)} \frac{1}{2} ||w||^2 + C \sum_{i=1}^{m} (\xi + \xi^*)
\]

\[
\begin{align*}
&\sum_{i=1}^{m} \left\{ y_i - w \cdot x_i - b \leq e + \xi & \xi \geq 0 \\
&w \cdot x_i + b - y_i \leq e + \xi^* & \xi^* \geq 0 
\end{align*}
\]

This constrained optimization problem is solved using the following Lagrangian form:
\[
\max H(\hat{e}, \hat{e}^*) = \sum_{j=1}^{m} y_j (\hat{e}_j - \hat{e}^*_j) - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{m} (\hat{e}_i - \hat{e}^*_i)(\hat{e}_j - \hat{e}^*_j) K(x_i, x_j) \\
- \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{m} (\hat{e}_i - \hat{e}^*_i)(\hat{e}_j - \hat{e}^*_j) K(x_i, x_j) \tag{4}
\]

Where \( \hat{e}_i \) and \( \hat{e}^*_i \) are the so-called Lagrangian multipliers. By the Lagrange multipliers calculation, an optimal desired weight vector is obtained, that is:

\[
w^* = \sum_{i=1}^{n} (\hat{e}_i - \hat{e}^*_i) K(x_i, x) \tag{5}
\]

Hence, the regression function is:

\[
f(x) = \sum_{i=1}^{n} (\hat{e}_i - \hat{e}^*_i) K(x_i, x) + b \tag{6}
\]

Based on the Karush–Kuhn–Tucker’s (KKT) conditions of solving quadratic programming problem, the corresponding data points of \( \hat{e}_i - \hat{e}^*_i \neq 0 \) are support vectors, which are employed in determining the decision function. SVM constructed by radial basis function (RBF) has excellent nonlinear forecasting performance and fewer free parameters need determination. Thus, in this work, RBF is adopted in the SVM. In (6), \( K(x_i, x) = \exp(-\|x_i - x\|^2 / \sigma^2) \). Here, \( C, \sigma \) and \( \epsilon \) are user-determined parameters, the election of the parameters plays an important role in the performance of SVM[4].

III. HYBRID PSO-SVM FOR PARAMETERS AND FEATURES SELECTION

A. Standard PSO (SPSO)

SPSO is a populated search method which is used to select \( C, \sigma \) and \( \epsilon \) parameters in SVM. Similar to genetic algorithms, PSO performs searches using a population (called swarm) of individuals (called particles) that are updated from iteration to iteration. The particle of SPSO is composed of three parts, \( C, \sigma \) and \( \epsilon \). To discover the optimal solution, each particle moves in the direction of its previously best position (pbest) and its best global position (gbest). For each particle \( i \) and dimension \( j \) search space, we assume: \( x_i = (x_{i1}, x_{i2}, \cdots x_{ij})^T \) denotes the current position of the \( i \)th particle, \( v_i = (v_{i1}, v_{i2}, \cdots v_{ij})^T \) denotes the fly velocity of the \( i \)th particle, the velocity and position of particles can be updated by the following equations:

\[
v_{ij}^{t+1} = w \cdot v_{ij}^{t} + c_1 \cdot rand_i (pbest_{ij} - x_{ij}) + c_2 \cdot rand_i (gbest_{ij} - x_{ij})
\]

\[
x_{ij}^{t+1} = x_{ij}^{t} + v_{ij}^{t+1} \tag{7}
\]

In the above formula, \( i \) is the evolutionary generation. \( v_{ij} \) is the velocity of particle \( i \) on dimension \( j \), whose value is limited to the range \([-v_{max}, v_{max}] \), \( x_{ij} \) is the position of particle \( i \) on dimension \( j \), whose value is limited to the range \([-x_{max}, x_{max}] \). The inertia weight \( w \) is used to balance the global exploration and local exploitation. The \( rand_i \) and \( rand_i \) are random function in the range \([0,1]\). Positive constant \( c_1 \) and \( c_2 \) are personal and social learning factors.

B. Binary PSO (BPSO)

The previous PSO algorithm is a standard PSO algorithm, i.e., each dimension of a particle can only be set as real values. So it is hard to be used in discrete optimization problems such as feature selection of network connection records in DARPA data. Datasets with unimportant, noisy or highly correlated features will significantly decrease the classification accuracy rate. By removing these features, the efficiency and classification accuracy rate can be obtained. For the BPSO algorithm, a binary encoding is adopted, where \( x_i \), \( p_t \) and \( g_t \) for each dimension are limited to 0 or 1, but this limitation is not used for particle velocity. The sigmoid function of velocity is a logical choice to do this as follows:

\[
s(v_{ij}^{t}) = \frac{1}{1 + \exp(-v_{ij}^{t})} \tag{8}
\]

The position is updated as follows:

\[
\text{if } rand() < s(v_{ij}^{t}) \text{ then } x_{ij}^{t+1} = 1 \\
\text{else } x_{ij}^{t+1} = 0 \tag{9}
\]

Where \( rand() \) sets the value of \( v_{ij}^{t} \) to a range of \([0,0.1,0]\) . The maximal velocity \( v_{max} \) can be used to limit the probability to 0 or 1. Selection feature of subset in this case is binary, that is, attack and normal features are distinguished[5].

C. Hybrid PSO-SVM Approach

Firstly, SPSO is used to elect the \( C, \sigma \) and \( \epsilon \) in SVM. Secondly, we selected the best feature subsets by using BPSO algorithm. Feature subset selection and parameter values determination: Each particle represents a solution, which denotes the selected subset of features and parameter values. The selected features, parameter values, and training dataset are used for building SVM classifier models. The basic process of the PSO algorithm is given by:

Step 1: (Initialization) Randomly generate initial particles. For the BPSO algorithm, the complete set of features is represented by a binary string of length \( N \), where a bit in the string is set to ‘1’ if it is to be kept, and set to ‘0’ if it is to be discarded, and \( N \) is the original number of features.

Step 2: (Fitness) Measure the fitness of each particle in the population. The selection of this fitness function is a crucial point in using the PSO algorithm, which determines what a PSO should optimize. Here, the task of the PSO algorithm is to find the global minimum value according to the definition of the fitness function. The definition of the fitness function for the basic method is simply the accuracy of detection.

Step 3: (Update) Compute the velocity of each particle.
Step 4: (Construction) For each particle, move to the next position.
Step 5: (Termination) Stop the algorithm if the termination criterion is satisfied; return to Step 2 otherwise.

IV. EXPERIMENT SETUP AND PERFORMANCE EVALUATION

A. Data Set and Processing
The data used here originated from MIT’s Lincoln Labs and is considered a standard benchmark for intrusion detection evaluations. In the experiments, we firstly utilized our feature selection algorithm to select important features, and then built intrusion detection systems using these selected features. The training data set is then separated into attack data sets and normal data sets, which are then subsequently fed into the hybrid PSO-SVM algorithms. Through the training process, hybrid PSO-SVM models can be built. We then feed the test data set into the PSO-SVM models.

B. Parameter of Hybrid PSO-SVM
The proposed model has been implemented in Matlab7.1 programming language. The experiments are made on a 1.80 GHz Core(TM) 2 CPU personal computer (PC) with 2.0G memory under Microsoft Windows xp professional. For the PSO algorithm, 30 particles were generated randomly. The learning factors $c_1$ and $c_2$ were set to 2.3 and 1.8, the inertia weight $w$ was reduced linearly from 0.9 to 0.5, the maximum number of iterations was set to 200, the dimensions of the solution space was set to 41 (each type of attacks will be included in one of the all 41 features), the maximum fly velocity of each particle was set at 200. According to a previous study [2], the searching range of parameter $C$ of SVM was between 0.0001 and 32 (Lin & Lin, 2003). The searching range of parameter $C$ of SVM was between 0.01 and 35,000, while the searching range of parameter $\delta$ of SVM was between 0.0001 and 32 (Lin & Lin, 2003). The fitness function selected for the PSO can directly reflect the classification performance as follows: $f = \text{Detection rate}$. Where the Detection rate denotes overall classification accuracy for each individual of a population obtained using ten multiple cross-validation with SVM classifiers, which will be used to guide the optimization of a particle.

C. Anomaly Detection Results
The research intends to compare the efficiency of SVM and PSO-SVM under different circumstances. Detection and identification of attack and non-attack behaviors can be generalized as the following: True positive (TP): the amount of attack detected when it is actually attack. True negative (TN): the amount of normal detected when it is actually normal.
Thus detection rate is:

$$\text{Detection rate} = \frac{TP}{TP + TN} \times 100\%$$

(10)

After selected five best feature subsets by BPSO algorithm, we then built several intrusion detection models on the sampled training datasets using the above feature subsets and all 41 features. We considered all attacks as a whole, and built two types of intrusion detection system, one type was built using all 41 features, and the other was built using selected features. We selected all attack type features including service, src_bytes, count, dst_host_count by BPSO and SVM parameters $C = 2245.2050$, $\sigma = 1.0632$ and $\varepsilon = 0.01327$ by SPSO.

We can calculate that intrusion detection systems with selected features have higher detection rate (99.8438%) than those(82.6387%) with standard features in terms of detecting known attacks.

V. CONCLUSION
In this paper, we proposed a hybrid PSO-based feature selection algorithm to build novel IDS. In SVM parameters $C$, $\sigma$ and $\varepsilon$ are elected by SPSO; IDS Data feature selection algorithm consists of search strategy - BPSO and evaluation criterion- SVM. We developed a series of experiments on KDD Cup (1999) intrusion detection dataset to examine the effectiveness of our feature selection and its free parameters in building effective IDS. The experiment results show that our approach is not only able to achieve the process of selecting important features but also to yield high detection rates for IDS. In our future work, we will further improve our feature selection algorithm on search strategy and evaluation criterion to help build efficient and practical intrusion detection.

ACKNOWLEDGMENT
The authors gratefully thank the anonymous referees and collaborators for their substantive suggestions. We also acknowledge research support from the Science Foundation of Northeastern University at Qinhuangdao of PRC.

REFERENCES