International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-4 Number-3 Issue-16 September-2014

Intrusion Detection System using Support Vector Machine (SVM) and Particle Swarm Optimization (PSO)

Vitthal Manekar[1], Kalyani Waghmare[2]

Abstract

Security and privacy of a system is vulnerable, when an intrusion happens. Intrusion Detection System (IDS) takes an important role in network security as it detects various types of attacks in the network. In this paper, the propose Intrusion Detection System using data mining technique: SVM (Support Vector Machine) and PSO (Particle Swarm Optimization). Here, first PSO performed parameter optimization using SVM to get the optimized value of C (cost) and g (gamma parameter). Then PSO performed feature optimization to get optimized feature. Then these parameters and features are given to SVM to get higher accuracy. The experiment is performed by using NSL-KDD dataset.

Keywords

Support Vector Machine (SVM), Particle Swarm Optimization (PSO), Intrusion Detection System (IDS).

1. Introduction

As network-based computer systems have important roles in modern society, they have become the targets of intruders. Therefore, we need to build the best possible rules to protect our systems. The security of a computer system is vulnerable when an intrusion takes place. An intrusion can be defined as any action done that harms the integrity, confidentiality or availability of the system. There are some intrusion prevention techniques which can be used to prevent computer systems as a first line of defence. A firewall is also one of it. But only intrusion prevention is not enough. As systems become more complex, there are always exploitable weaknesses in the systems due to design and programming errors, or various penetration techniques. Therefore Intrusion detection is required as another measure to protect our computer systems from such type of vulnerabilities.

Manuscript received August 15, 2014.

Vitthal Manekar, Department of Computer Engineering, PICT, Pune, India.
Kalyani Waghmare, Department of Computer Engineering, PICT, Pune, India.

Intrusion prevention techniques, such as user authentication and information protection via encryption have been used to protect computer systems as a first line of defense. Intrusion prevention alone is not sufficient because as systems become more complex, there are always exploitable weaknesses in the systems due to design and programming errors, or various “socially engineered” penetration techniques. Intrusion detection is therefore needed as another wall to protect computer systems. IDS system is only detect the intrusion with the help of different classification algorithm. The main functionality in intrusion system is performed by classification algorithm. There are several algorithm used with IDS such as PCA with SVM, genetic algorithm with SVM. The accuracy of IDS depends on these algorithms.so that why PSO is used along with SVM to improve IDS.

2. Literature Survey

There are lots of study to be done to prepare improve model for SVM to get maximum accuracy in IDS. Some of these techniques are studied below.

In 2008, Zhou, Jianguo, et al. Proposed system a Culture Particle Swarm Optimization algorithm (CPSO) used to optimize the parameters of SVM. By using the colony aptitude of particle swarm and the ability of conserving the evolving knowledge of the culture algorithm, this CPSO algorithm constructed the population space based on particle swarm and the knowledge space. The proposed CPSO-SVM model that can choose optimal values of SVM parameters was test on the prediction of financial distress of listed companies in China [5].

In 2011, Kolias, Constantinos, Georgios Kambourakis, and M. Maragoudakis et al. suggested that the RBF has certain parameter that affects the accuracy. PSO is used along with RBF artificial neural network it will improve the accuracy. If it is used in IDS it will improves the accuracy of classification [6].

In 2011, Horng, Shi-Jinn, et al. proposed an SVM-based intrusion detection system, which used a hierarchical clustering algorithm, leave one out, and the SVM technique. The hierarchical clustering algorithm provided the SVM with fewer, abstracted, and higher-qualified training instances that are derived from the KDD Cup 1999 training set. It was able to greatly minimize the training time, and improve the performance of SVM. The simple feature selection procedure (leave one out) was applied to eliminate unimportant features from the training set so the obtained SVM model could classify the network traffic data more accurately [1].

In 2012, Gaspar, Paulo, Jaime Carbonell, and José Luís Oliveira et al. gave the review on strategies that are used to improve the classification performance in term of accuracy of SVMs and perform some experimentation to study the influence of features and hyper-parameters in the optimization process, using kernels function. Huang et al provide a study on the joint optimization of C and g parameters (using the RBF kernel), and feature selection using Grid search and genetic algorithms [2].

In 2014, Ahmad, Iftikhar, et al. proposed a genetic algorithm to search the genetic principal components that offers a subset of features with optimal sensitivity and the highest discriminatory power. The support vector machine (SVM) is used for classification. The results show that proposed method enhances SVM performance in intrusion detection [3].

3. Optimization of SVM using PSO

There are four steps used in this system. These are as below:

A. Data Pre-Processing:

The training dataset of NSL-KDD consist of approximately 4,900,000 single connection vectors [13]. Each connection contains 42 features including attacks or normal. From these labeled connection records, we need to map the labels to numeric values so as to make it suitable to be the input of our machine learning algorithm: SVM. Also assign target class to the connections according to class label feature, which is the last feature in the connection record. By considering this, we have assigned a target class ‘zero’ for ‘normal connection’ and a ‘one’ for any deviation from that (i.e. if that is an attack).

In this step, some useless data will be filtered and modified. For example, some text items need to be converted into numeric values. Every process (i.e. single connection vector) in the database has 41 attributes.

B. Conversion of datasets to LibSVM format:

Pre-processed datasets are converted to LibSVM format. In this process, first categorical features from both training and testing datasets are converted to a numeric value and then we have to determine target classes for classification phase. Here, conversion and scaling function determined two target classes: class ‘zero’ for normal instance and class ‘one’ for attack or intrusion [7]. Then it saved target class and feature values of each instance in LibSVM format.

LibSVM format is:
[Label] [Index 1]:[value 1] [index 2]:[value 2]…
Where,
‘Label’ is target ‘classes’ of classification. Usually put integers here. [0, 1] target class
‘Index’ is the ordered index. Usually continuous integer.
‘Value’ is the input data for training. Usually lots of real (floating point) numbers. Input dataset to the problem we are trying to solve involves lots of 'features' or 'attributes', so the input will be a set (or vector/array).
After this step, we have to perform linear scaling of LibSVM format datasets and store these scaled datasets for further use. Linear scaling of datasets is done to improve the performance of SVM classification.

C. Optimization using SVM and PSO:

The NSL-KDD dataset in LibSVM format is scaled in [0, 1]. The scaling is the method used to reduce the impact of bigger value on small value. It improves the performance of SVM. Here we are using LibSVM. Jar for implementation SVM algorithm. The SVM is statistical machine learning algorithm takes the data input in the form of numeric value and prepare the model or build the model for classification. There are four type of kernel function using SVM for classification. Kernel function is used to map the dataset into higher dimension. These are linear, RBF, Polynomial. There are several parameters used in SVM. In RBF Kernel type of SVM, there are two parameter C (Cost) and g (gamma). The accuracy of the SVM for RBF type depends on these two parameters. Optimized value of this parameter and features increases the accuracy of SVM. PSO uses to optimized features and parameters. PSO is dynamic clustering algorithm based social interaction [8]. It has fast convergence ability [11]. It works better in integration of SVM [9].

D. Classification Using SVM :

The SVM uses a portion of the data to train the system, finding several support vectors that represent the training data. These support vectors will form a SVM model. According to this model, the SVM will work with PSO for C and g (mention in Eqn(1) and Eqn(2)) optimization and feature subset selection. And it improves the SVM model. After that SVM is used to classify a given unknown dataset. A basic input data format and output data domains are listed as follows

$$(X_i, Y_i)\ldots\ldots\ldots\ldots\ldots(X_n, Y_n)$$

Where

$$X \in R^m \text{ and } Y \in \{0, 1\}$$

Where (X_i, Y_i)……………. (X_n, Y_n) is training data records, n is the numbers of samples m is the inputs vector, and y belongs to category of class '0' or class '1' respectively. On the problem of linear, a hyper plane can be divided into the two categories as shown in Figure. The hyper plan formula is:

$$(w \cdot x) + b = 0$$

The category formula is:

$$(w. x) + b \geq 0 \text{ if } Y_i = 1$$

$$(w. x) + b \leq 0 \text{ if } Y_i = 0$$

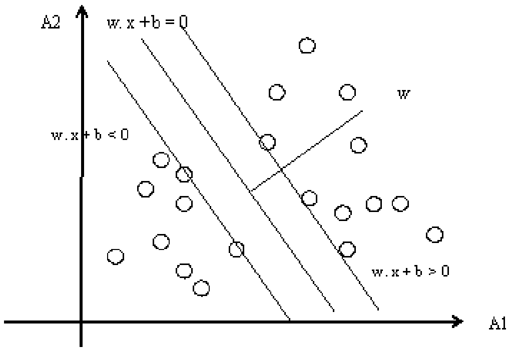

Figure 1: Hyper-Plane of SVM

A classification task usually involves with training and testing data which consist of some data instances. Each instance in the training set contains one "target value" (class labels: Normal or Attack) and several "attributes" (features).The goal of SVM is to produce a model which predicts target value of data instance in the testing set which is given only attributes. To attain this goal there are four different kernel functions.in this experiment RBF kernel function is used

The Formula for RBF Kernel Optimization function

$$\text{Exp } (-g * | X_i - X_j | \wedge 2) \qquad \text{Eqn (1) [12]}$$

Finding vectors from training data is formulated as

$$\underset{w,\, b,\, \xi}{\text{Minimize}} \ \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{l} \xi i \qquad \text{Eqn (2)}$$

4. Experiment Result

In this Experiment, PSO optimizes the parameters of SVM (RBF) using SVM and also reduces the features of the training set. It reduces he noisy feature from the training set. Training set contains 25149 records and testing set contains 11850 records. The algorithms used in experiment are given below.

During this Experiment, comparison of Different kernel unction of SVM with feature selection with accuracy. The measure of accuracy is given confusion matrix [10]. The kernel function used here are Linear, Gaussian, RBF, Polynomial. The results show that RBF Kernel function with optimized features gives highest accuracy.

Table 1: Comparison of accuracy and time of Different kernel function of SVM

Algorithm	Accuracy	Time(in second)
SVM(RBF with parameter and feature selection	81.8	13.719
SVM (RBF without Feature Selection)	47.9916	11.11
SVM(Linear With feature selection)	49.4093	17.157
SVM(Gaussian with feature selection)	52.6835	41.986

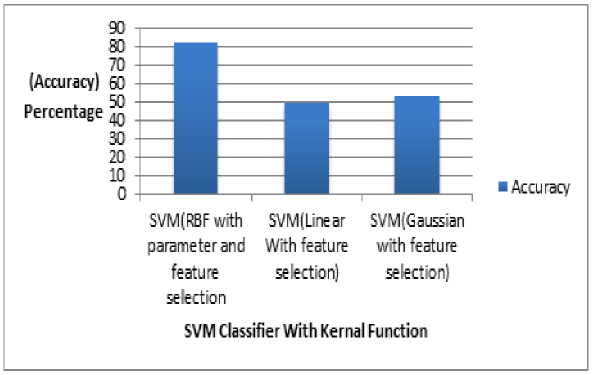

Figure 2: Comparison of Accuracy SVM with Different Kernel Function

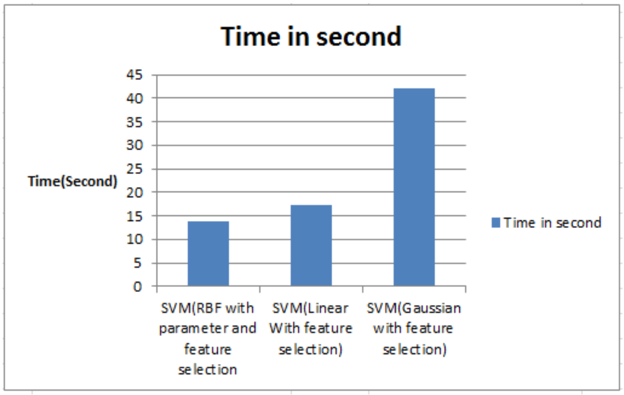

Figure 3: Comparison of Time SVM with Different Kernel Function

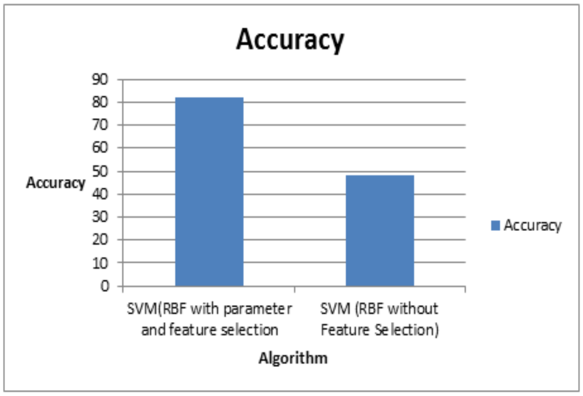

Figure 4: Comparison of Accuracy of SVM with RBF

Parameter Optimization

```
Input:   $S_{tr}$ , $S_{test}$ , $C_l$ , $C_p$ , $g_l$ , $g_p$
Output:  C, g
/* $S_{tr}$ , $S_{test}$ are the scaled training and testing dataset.
$C_l$, $C_p$ is the lower and upper limit of parameter C. $g_l$ ,
$g_p$ is the lower and upper limit of parameter g. */
Step 1: particle = {pos, fitness, velocity, bestpos, bestfitness}
Step 2: initialize population of parameter [max_size of swarm=10]
Particle [] swarm=new particle [max_size], bestGlobalpos, bestGlobalfit
 For each swarm i from 1 to 10
    Initialize pos in range [$C_l$, $C_p$] and [$g_l$, $g_p$ ]
   // particle Consist of two   dimension C and g
          $P_{Fitness}$= SVM ($S_{tr}$, $S_{test}$ , pos);
 // calculation of fitness value based on mean square error (MSE) using SVM
    Initialize velocity in range [$C_l$, $C_p$] and [$g_l$, $g_p$ ]
    swarm[ i ]← {pos, $P_{Fitness}$, velocity, pos, $P_{Fitness}$ }
     if ( swarm[i].fitness< bestGlobalfit) Then
           bestGlobalfit = swarm[i].fitness;
           bestGlobalpos=Swarm[i].pos;
      End if
     End For
Step 3: choose particle with best fitness value
       While ( i < max_iteration)
             Do for j from 1 to 10
             Particle currP = swarm[i];
 Newvelocity = w * velocity[j] + (c1 * r1 * (currP.bestpos - currP.pos])) + (c2 * r2 * (bestGlobalpos - currP.pos));
 // w is inertia c1, c2  cognitive local and global weight
             Newpos = pos + Newvelocity;
             Newfit =SVM (Newpos);
             if (Newfit < currP.bestfit) Then
               currP.pos= Newpos;
                currP. bestfitness = Newfit;
             End if
             if (Newfit < bestGlobalfit) Then
               bestGlobalpos = Newpos;
               bestGlobalfit = Newfit;
             End if
             End for loop
       End While
```

Feature Optimization

```
Step 1:  take l as the binary string of size 40
 // as l = 0101010101010101001010101….
Step 2:  particle = {pos, fitness, bestpos, bestfitness}
Step 3:  Particle [ ] swarm=new Particle [max_size], bestGlobalpos, bestGlobalfit
Step 4:  do for each particle in swarm i from 1 to 10
              pos= random_string ( l );
              writeRandomFeatures (pos, $S_{tr}$);
 //In this function, feature.txt generated from binary string, $S_{tr}$ is scaled training Dataset
           Fitness=SVMF (feature.txt, $S_{test}$, C, g);
// In this function, $S_{test}$ is scaled test dataset , C and g is parameter obtain from parameter  optimization
      swarm[ i ]←particle{pos, Fitness, pos, Fitness}
      if (swarm[i]. Fitness < bestGlobalfit) Then
            bestGlobalfit = swarm[i]. Fitness;
            bestGlobalpos = swarm[i].pos;
          End if
       End for
Step 5: do while i from 1 to max_iteration
             Newpos ; Newfit ;
            Do for j from 1 to 10
            Particle P=swarm[ i ];
            Newpos = random_string ( pos );
             writeRandomFeatures (Newpos, $S_{tr}$);
             Newfit= SVMF (feature.txt, $S_{test}$, C, g);
           if (Newfit < P. bestfitness)
                    P. bestpos = Newpos;
             P. bestfitness = Newfit;
           End if
          if (Newfit < bestGlobalpos)
                    bestGlobalpos = Newpos;
           bestGlobalfit = Newfit;
          End if
      End For
    End while
```

5. Conclusion

Here we used two method of optimization. First, parameter optimization and other is Feature optimization. Parameter optimization gives an optimized value of parameters (C and g) and feature optimization gives optimized features and used these features and parameters with different kernel function of SVM. Here we have used linear, RBF, Gaussian and Polynomial kernel function with SVM. The RBF kernel function gives highest accuracy. Hence here we conclude that SVM with RBF kernel function give high accuracy with optimized features and also takes less time for classification.

In future, we can include these methods in mahout (machine learning library) to improve the accuracy of SVM.

References

[1] Horng, Shi-Jinn, et al. "A novel intrusion detection system based on hierarchical clustering and support vector machines." Expert systems with Applications 38.1 (2011): 306-313.

[2] Gaspar, Paulo, Jaime Carbonell, and José Luís Oliveira. "On the parameter optimization of Support Vector Machines for binary classification." J Integr Bioinform 9.3 (2012): 201.

[3] Ahmad, Iftikhar, et al. "Enhancing SVM performance in intrusion detection using optimal feature subset selection based on genetic principal components." Neural Computing and Applications 24.7-8 (2014): 1671-1682.

[4] Hashem, Soukaena Hassan. "Efficiency of Svm and Pca to Enhance Intrusion Detection System." Journal of Asian Scientific Research 3.4 (2013): 381-395.

[5] Zhou, Jianguo, et al. "The study of SVM optimized by Culture Particle Swarm Optimization on predicting financial distress." Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on. IEEE, 2008.

[6] Kolias, Constantinos, Georgios Kambourakis, and M. Maragoudakis. "Swarm intelligence in intrusion detection: A survey." computers & security 30.8 (2011): 625-642.

[7] Bhavsar, Yogita B., and Kalyani C. Waghmare. "Intrusion Detection System Using Data Mining Technique: Support Vector Machine." International Journal of Emerging Technology and Advanced Engineering 3.3 (2013).

[8] L. Zhen, L. Wang, X. Wang Z. Haung "A Novel PSO-inspired Probability-based Binary Optimization Algorithm" 2008 International Symposium on Information Science and Engineering.

[9] Alba, Enrique, et al. "Gene selection in cancer classification using PSO/SVM and GA/SVM hybrid algorithms." Evolutionary Computation, 2007. CEC 2007. IEEE Congress on. IEEE, 2007.

[10] Han, Jiawei, and Micheline Kamber. Data Mining, Southeast Asia Edition: Concepts and Techniques. Morgan kaufmann, 2006.

[11] Garšva, Gintautas, and Paulius Danenas. "Particle swarm optimization for linear support vector machines based classifier selection." Nonlinear Analysis 19.1 (2014): 26-42.

[12] Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[13] KDD Cup 1999. Available on http://kdd.ics.uci.edu/ Databases/kddcup99/kddcup99.html, October 2007.

Vitthal Manekar is pursuing his Master Degree in Computer Engineering from PICT Pune. He received his B.E in Computer Science and Engineering from Nagpur University in 2011. His research interest include: Data Mining, Distributed System.

Prof. Kalyani Waghmare is Assistant Professor in Computer department at PICT College, Pune. She has completed her Master Degree in IT. She has completed her B.E in Computer Science and Engineering. Her research interest include: Data Mining, Distributed