Network Security Situation Assessment Model Based on GSA-SVM
Yu-xin CHEN, Xiao-chuan YIN* and Ao SUN
Information and Navigation College, Air Force Engineering University, Xi’an, China
*Corresponding author

Keywords: Network security situation assessment (NSSA), Network security, Gravitational search algorithm (GSA), SVM.

Abstract. In order to solve the problem of insufficient accuracy caused by the selection of support vector machine (SVM) parameters in situation assessment, a network security situation assessment model (GSA-SVM) based on GSA optimization of SVM parameters is proposed, which combines the characteristics of less parameters needed to be set by gravity search algorithm (GSA) and strong global optimization capability. Firstly, the model receives the situation assessment data evaluated by the expert system, and then searches for the optimal parameters in SVM through GSA, minimize the error between the generated data and the actual network security situation assessment data. The effect of the model is verified by using Eggcrate function and actual situation assessment data. The results show that this method has good learning ability and is better than Particle Swarm Optimization (PSO) in SVM optimization.

Introduction
Nowadays, the development of computer networks makes network security a top priority. At the same time, the number of threats to network security is huge and often does not occur alone. Thus, Network Security Situation Assessment (NSSA) is very important.

Tim Bass[1] put forward the concept of network security situation awareness (NSSA). Such situation awareness means the current states and trends of the whole network, which was constructed by the factor of operation status of various network equipment, network and user behavior. Keramati et al. [2] proposed an evaluation method to calculate attack reach ability using the ratio of the average score and path length of the general vulnerability scoring system. This method can quantitatively analyze network security and calculate losses on the network. Szwed et al. [3] proposed an evaluation method based on fuzzy cognitive map, which uses fuzzy cognitive map to obtain the dependency relationship of important assets in the network and evaluate the degree of harm. Graf[4] and others use the expert system to solve the network security situation assessment problem. The expert system is transformed into an assessment system through setting rules and inputting knowledge base. However, the preceding method is subjective, the data source is relatively single, and the evaluation index is also single.

Machine learning technology can find regulars from data. Thus, machine learning technology can be used in NSSA to extract regulars from previous data and generate new assessment results. As a machine learning technology, support vector machine (SVM) [5] model has been introduced to effectively solve pattern recognition problems due to its superior performance.

However, there is often no corresponding theory for parameter selection of SVM, which requires a lot of experience and constant attempts, and costs a lot of time. In order to optimize the parameters, a model such as Particle Swarm Optimization (PSO) [6], combined with SVM appears. Rashedi et al. proposed Gravity Search Algorithm (GSA) [7] in 2009. Inspired by the law of movement of objects under the influence of gravity, rashedi designed a new optimization search algorithm. the literature [6] verifies that its global optimization ability is obviously better than PSO algorithm.

Therefore, in order to solve the problem of evaluation accuracy in situation evaluation, a network security situation evaluation model using GSA to optimize SVM parameters is proposed. this model combines the characteristics of less parameters that GSA needs to set and strong global optimization
ability to optimize SVM parameters. The evaluation result of the established evaluation model is better than that of PSO-SVM.

Basic Principle

GSA

The basic principle of gravity search algorithm (GSA) is the law of universal gravitation, and it is assumed that all particles in space have its mass, and the particles in space are not affected by any resistance, and the particles are continuously approaching each other under the action of gravity.

The magnitude of the gravitational force is inversely proportional to the square of the distance between particles, and the product of inertial mass is directly proportional.

The gravitational interaction between particles is expressed as follows: if you want to know the specific steps and principles of the gravitational search algorithm, we can see from article[6]

Suppose there are \( n \) particles, the position of particle \( i \) is

\[
X_i = (x_{i1}, x_{i2}, ..., x_{id}, ..., x_{in}) \quad i=1,2,...,N
\]

The position of particle \( i \) in \( d \)-dimensional space is \( x_{id} \)

The \( d \)-th dimension of particle \( i \) at time \( t \) is acted upon by particle \( j \) as follows:

\[
F_{ji}^d(t) = G(t)\frac{M_i(t)M_j(t)}{R_{ij}(t)^{\alpha}}(x_{d}^j(t) - x_{d}^i(t)) + \epsilon
\]

Where \( M_i \) and \( M_j \) are the inertial masses of particles \( i \) and \( j \), respectively, which are the distances between particles \( i \) and \( j \), \( G(t) \) are gravitational constants, and \( \epsilon \) are very small constants.

Gravity constant as follows:

\[
G(t) = G_0 e^{-\frac{at}{T}}
\]

\( G_0 \) is initialization gravitational constant, \( a \in [20,30] \), \( T \) is the maximum number of iterations.

The attraction of particle \( i \) to other particles at time \( t \) is:

\[
F_{i}^d(t) = \sum_{j=1,j\neq i}^{N} rand \cdot F_{ji}^d(t)
\]

The acceleration corresponding to the \( d \)-th dimension particle \( i \) is

\[
a_{i}^d(t) = \frac{F_{i}^d(t)}{M_i(t)}
\]

Inertia mass \( M_i(t) \) of particle \( i \)

\[
m_{i}(t) = \frac{f_{i}(t) - f_{worst}(t)}{f_{best}(t) - f_{worst}(t)}
\]

\[
M_i(t) = m_{i}(t)\sum_{i=1}^{N} m_{i}(t)
\]

\( f_{i}(t) \) indicates the fitness function value of particle \( i \) at time \( t \), and \( f_{worst}(t) \) and \( f_{best}(t) \) are respectively the worst fitness function value and the best fitness function value of the entire particle group at time \( t \).
The velocities $v^d_i(t)$ and positions $x^d_i(t)$ of the particles $i$ in the $d$-th dimension are:

$$v^d_i(t) = \text{rand} \cdot v^d_i(t-1) + a^d_i(t) \tag{7}$$

$$x^d_i(t) = x^d_i(t-1) + v^d_i(t) \tag{8}$$

After calculating the new position and speed during each iteration, proceed to the next iteration until the maximum number of iterations is met or the specified accuracy is met.

**SVM**

SVM is very effective in solving nonlinear, small sample and other problems. SVM nonlinear regression prediction combines structural risk minimization theory and VC dimension theory. After mapping the data $x$ from low-dimensional input space to the high-dimensional feature space and carries out linear regression in the high-dimensional feature space to construct the optimal decision function. Linear regression function:

$$f(x) = w^T \phi(x) + b \tag{9}$$

Using $\varepsilon$ insensitive loss function to optimize, through the minimum value of the function to find the optimal regression function:

$$\min \frac{1}{2} w^T w + C \sum_{i=1}^{n} (\xi_i + \xi_i^*) \tag{10}$$

Constraint condition

$$y_i - [w, \phi(x_i)] - b \leq \xi_i + \varepsilon \tag{11}$$

$$[w, \phi(x_i)] + b - y_i \leq \xi_i^* + \varepsilon \tag{12}$$

Where $C$ is the penalty coefficient, $b$ is offset, $\varepsilon$ as a function of insensitive loss, $\xi_i, \xi_i^*$ is a relaxation variable. The nonlinear regression function obtained by quadratic programming:

$$f(x) = \sum_{i=1}^{n} (\beta_i - \beta_i^*) K(x, x_i) + b \tag{13}$$

Kernel function of SVM is $K(x, x_i)$

$\beta_i, \beta_i^*$ is lagrange multiplier. The nonlinear kernel function selects RBF as the kernel function.

$$K(x, x_i) = \exp\left\{-\frac{|x-x_i|^2}{\sigma^2}\right\} \tag{14}$$

$C$ in Eq.10 is the penalty coefficient, i.e. the tolerance for errors. The higher the $C$ is, the worse the tolerance of error will be, and at the same time, the fitting phenomenon will appear. The smaller the value of $C$ is, the opposite will result in an under-fit. If $C$ is too large or too small, its generalization ability will be affected.

Gamma is a parameter in RBF function. The selection of its value affects the distribution after the data is mapped to the new feature space. The number of support vectors potentially determines the speed of training and prediction. The gamma in RBF is:
\[ k(x, z) = \exp(-\frac{d(x, z)^2}{2*\sigma^2}) = \exp(-\text{gamma} \cdot d(x, z)^2) \Rightarrow \text{gamma} \] (15)

So, parameters need to be optimized in SVM are: error penalty coefficient C and kernel parameter gamma (abbreviated as g), where penalty coefficient C controls the ratio between classification interval and allowable sample error. Kernel parameter g mainly affects the distribution of sample data in high dimensional space.

**Network Situation Assessment based on GSA-SVM**

![Network Situation Assessment based on GSA-SVM](image)

What the network security situation assessment needs is the knowledge of human experts and the rules formulated by human experts. GSA-SVM learns the GSA-SVM situation assessment model from the situation assessment data composed of the situation attributes and situation assessment values generated after the expert assessment, and then uses it in the network security situation assessment. There are m attributes in the expert system, \((x_1, x_2, ..., x_m)\) defined by experts to describe the situation. Each expert has its different experiences. when evaluating network security, he often focuses on some aspects, which is subjective and makes the evaluation inaccurate. In order to reflect the overall state of network security as much as possible, it is necessary to weigh the opinions of n different experts to generate comprehensive situation NSSA, attributes and comprehensive situation as NSSA data source for GSA-SVM learning.

**Experiment Analysis**

Firstly, Eggcrate function is used to verify the convergence speed of GSA and PSO. The function has several extreme points, of which the minimum value is 0 and the position is at the origin.

Eggcrate function:

\[ f(x) = x^2 + y^2 + 25*(\sin^2(x) + \sin^2(y)) \quad -2\pi \leq x, y \leq 2\pi \] (16)

from Fig.3 we can get some solution that under the same number of iterations, GSA algorithm converges faster than PSO.
This paper uses MATLAB R2012b software and uses libsvm designed by Lin chih-jen. In this experiment, one of user1 and user2 may be an attacker. user1 and user2 access one of FTP and IIS web server in the network. Attackers exploit vulnerability CVE-2011-0762 to attack FTP and CVE-2004-2650 to attack IIS web server. According to the vulnerability scoring system cvss, the network is rated as a NSSA value, and the network security situation is divided into five levels: good (G), better (B), normal (N), risk (R), and high risk (HR) and corresponding to quantitatively described in five intervals of [0,1), [1,2.5), [2.5,6), [6,8), [8,10). Collect the historical data of NSSA manually evaluated by experts in Fig. 3 select the number of alarms (x1) and the rate of change of subnet traffic (x2); The historical frequency of security events (x3), the number of security devices (x4) in the subnet, the total number of open ports (x5), the critical coordination mean survival time (x6) in the subnet as the input of the data set, the expert evaluation result as expected output, 400 samples randomly selected as the training set, and 100 samples as the test set, the generalization ability of the test model to normalize the data input, Normalized between 0 and 1, the data have shown in Table 1.

<table>
<thead>
<tr>
<th>sample number</th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
<th>x6</th>
<th>SA level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.655</td>
<td>0.136</td>
<td>0.832</td>
<td>0.271</td>
<td>0.144</td>
<td>0.361</td>
<td>B</td>
</tr>
<tr>
<td>2</td>
<td>0.069</td>
<td>0.729</td>
<td>0.068</td>
<td>0.311</td>
<td>0.270</td>
<td>0.229</td>
<td>N</td>
</tr>
<tr>
<td>3</td>
<td>0.106</td>
<td>1.186</td>
<td>1.031</td>
<td>0.863</td>
<td>0.787</td>
<td>0.416</td>
<td>R</td>
</tr>
<tr>
<td>4</td>
<td>0.621</td>
<td>0.161</td>
<td>0.370</td>
<td>0.544</td>
<td>0.163</td>
<td>0.402</td>
<td>VR</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>400</td>
<td>0.220</td>
<td>0.589</td>
<td>0.186</td>
<td>0.295</td>
<td>0.540</td>
<td>0.245</td>
<td>VR</td>
</tr>
</tbody>
</table>

The GSA-SVM is used to train the data in Table1. GSA selects the optimal parameters C and g. after training, the mse values between the data generated in the two algorithms and the actual NSSA
are calculated respectively, then the mse of PSO-SVM is 0.0168 and the GSA-SVM is 0.0015, which shows that the two algorithms have strong fitting ability to the training data, while GSA has less error than PSO.

![Figure 4. Comparison of SA value between GSA-SVM and PSO-SVM.](image)

Afterwards, using the remaining 100 samples as the test set, the situation assessment values with sample numbers between 80 and 100 are selected. the training results obtained are shown in figure 4: the mse of GSA-SVM, PSO-SVM compared to the actual NSSA are 19.4896 and 23.8463 respectively after calculation. Obviously, the evaluation effect of GSA-SVM is better than PSO-SVM.

**Summary**

Machine learning technology is an effective method and therefore can be used for NSSA. In order to solve the problem of insufficient accuracy caused by the selection of support vector machine (SVM) parameters in situation assessment, a network security situation assessment model (GSA-SVM) based on GSA optimization of SVM parameters is proposed, which combines the characteristics of less parameters needed to be set by gravity search algorithm (GSA) and strong global optimization capability. A set of experiments on Eggcrate function and practical problems verify the effectiveness of the method. The results show that the method has good learning ability from the previous situation data and can generate perception of the current situation. It can be further applied to NSSA systems.

**Acknowledgement**

This research was supported in part by the National Natural Science Foundation No. 71503260 and Industrial science and technology research project in Shan Xi province No. 2016GY-087. And thanks my advisor YIN gave me a lot of help, i have learned a lot about how to find ideas, this is important for research. I also thanks to my classmate SUN, she help me find some materials, so i can find something new quickly.

**References**


