Journal of Mathematical Sciences: Advances and Applications
Volume 39, 2016, Pages 1-20
Available at http://scientificadvances.co.in
DOI: http://dx.doi.org/10.18642/jmsaa_7100121648

INTRUSION DETECTION BASED ON IMPROVED LEAST SQUARES MULTI-CLASSIFICATION TWIN SUPPORT VECTOR MACHINE

SHENGKAI ZHONG

Department of Information Science and Technology
Jinan University
Guangzhou
Guangdong, 510632
P. R. China
e-mail: 352176822@qq.com

Abstract

For the detection to modern networks intrusion, an intrusion detection model is put forward based on improved twin support vector machine. It employs new binary particle swarm algorithm to select parameters and feature subset. For the multi-classification problems, the least squares multi-classification twin support vector machine takes empirical risk minimization principle. And its generalization is optimized by adding rule item. The experiments use classic dataset called KDD'99. The result shows that the proposed algorithm is effective to solve the two categories and three categories problems of networks intrusion detection, improving the accuracy of classifiers.

2010 Mathematics Subject Classification: 68T10.
Keywords and phrases: feature selection, twin support vector machine, multi-class intrusion detection, particle swarm algorithm.
Received March 22, 2016

1. Introduction

Intrusion detection system (IDS) is becoming a critical component of network security with the development of computer science. There are two main categories of intrusion detection: Anomaly intrusion detection system and misuse detection system. Anomaly detection is implemented with intelligent system and assumes that intrusions are highly correlated to abnormal behaviours [1, 2]. The job of feature selection is to eliminate inappropriate and redundant data which facilitate to improve the performance of learning algorithms. Feature selection (FS) is a spirited and productive field of research area in machine learning, data mining, pattern recognition, etc [3, 4].

The formulation of SVM is proposed by Vapnik et al. [5, 6] in 1990s which is based on statistical learning theory. On the basis of SVM and GEPSVM, Jayadeva et al. [7] proposed a novel binary classifier, twin support vector machine (TWSVM), which classifies the patterns of two classes by using two non-parallel hyper-planes. TWSVM solves a pair of smaller size QPPs instead of one complex QPP as in the conventional SVM. There are two different ways to improve the conventional TWSVM. In first method, improvement is achieved using some modifications in the formulation of original method in order to enhance its performance. In second method, conventional TWSVM is extended from binary classes to multi-classes.

Applications of TWSVM are used in various field and it is a very useful approach for the pattern classification [8-10]. The goal of TWSVM is to construct a model to predict the class label of given data samples. The dataset is partitioned into training and testing datasets. A classifier is constructed by using training data set and its classification ability is checked with the help of testing data set. TWSVM is used for intrusion detection which improves the detection speed and accuracy as well as decrease the time complexity [11, 12].

The evaluation data is from KDD'99, and there are only few public datasets like KDD'99 and the majority of the experiments in the intrusion detection domain performed on these datasets. Since our model is based on supervised learning methods, KDD'99 is the only available dataset which provides labels for both training and test sets [13].

In this paper, following the line of research in [10, 14, 15], a squares regularization version of the parameter of LST-KSVC (least squares twin K-class support vector classification) is proposed and is called improved LST-KSVC. The feature selection and parameters optimization of the proposed classifier are using the NBPSO.

This paper is organized as follows. Section 2 briefly dwells on the BPSO and the new improved PSO. Section 3 describe the LSSVM, TWSVM, and LST-KSVC. The improved LST-KSVC is formulated and described in Section 4. Section 5 provides some interesting experimental results on KDD'99 datasets to investigate our proposed algorithm and concluding remarks are given in Section 6.

2. New Binary Particle Swarm Optimization

Particle swarm optimization (PSO) algorithm is a kind of intelligent algorithm based on the swarm. The velocities of particles, largest number of iterations and the size of population are generated by the initial value. The position and the velocities of particles are updated on each generation by the formula, then the best position of the particles and swarm in history will be recorded.

PSO is originally used to solve the continuous space optimization problems, while the problems are always the discrete or combinatorial on actual engineering application. Binary particle swarm optimization (BPSO) is proposed and applied on the actual problems. The particles updating formula of the BPSO is following:

$$v_{ij}^{(t+1)} = wv_{ij}^{(t)} + c_1 r_1 \left(p_{ij} - x_{ij}^{(t)}\right) + c_2 r_2 \left(p_{gj} - x_{ij}^{(t)}\right), \tag{1}$$

$$x_{ij}^{(t+1)} = \begin{cases} 1, \text{ rand} < S(v_{ij}^{(t+1)}), \\ 0, \text{ rand} \geq S(v_{ij}^{(t+1)}), \end{cases} \tag{2}$$

where rand is a random number of submitting to the uniformly distribution, $S(\cdot)$ is the Sigmoid function:

$$S(v_{ij}^{(t+1)}) = 1 / (1 + \exp(-v_{ij}^{(t+1)})).$$

The parameters are positive number in the model, therefore plus and minus signs are needless at the highest position. Decimal fraction can be converted by using the decimal conversion between decimal and binary rules. The feature on this position is selected or not that depends on the position of features is 1 or 0, respectively.

The position and velocities of particles are initialed randomly. $x_i = [x_{i1}, x_{i2}, \ldots, x_{iN}]$, $v_i = [v_{i1}, v_{i2}, \ldots, v_{iN}]$, where N is the length of the particle coding, the value of x_{ij} and v_{ij} is 0 or 1.

The velocities and position updating formulas as following:

$$v_{ij}^{(t+1)} = w \otimes v_{ij}^{(t)} \oplus C_1 \otimes \left(p_{ij} \oplus x_{ij}^{(t)}\right) \oplus C_2 \otimes \left(p_{gj} \oplus x_{ij}^{(t)}\right), \tag{3}$$

$$x_{ij}^{(t+1)} = x_{ij}^{(t)} \oplus v_{ij}^{(t+1)}, \tag{4}$$

where C_1 and C_2 are learning factors, w is the inertia weight, p_i is the best position of particles on each generation, and p_g is the best position of swam in history.

They are updated as following formula:

$$X = (X_1 + (X_2 - X_1) \times i_{th} / M), \tag{5}$$

where M is the largest number of iteration, i_{th} is the i generation X_1 and X_2 the initial value and final value of parameter, respectively.

We define an operation $a \oplus b$: determine the positive and negative of the symbol $sign^{(k)}$ at the front of b firstly. If $sign^{(k)} > 0$, adding the values of a and b at the corresponding position, one is carried forward in binary system. At the highest position of particle, we set a mechanism called *touch boundary back*: the value of highest position g exceeds one after adding, we set g to be one directly.

If $sign^{(k)} < 0$, subtracting the values of a and b at the corresponding position, one is borrowed to binary system, and then the value f of result's highest position from subtraction is determined whether is -1 or not. If is -1, changing the position of both a and b, subtracting and set $sign^{(k+1)} < 0$, otherwise just set $sign^{(k+1)} > 0$.

The basic flow chart is shown in Figure 1.

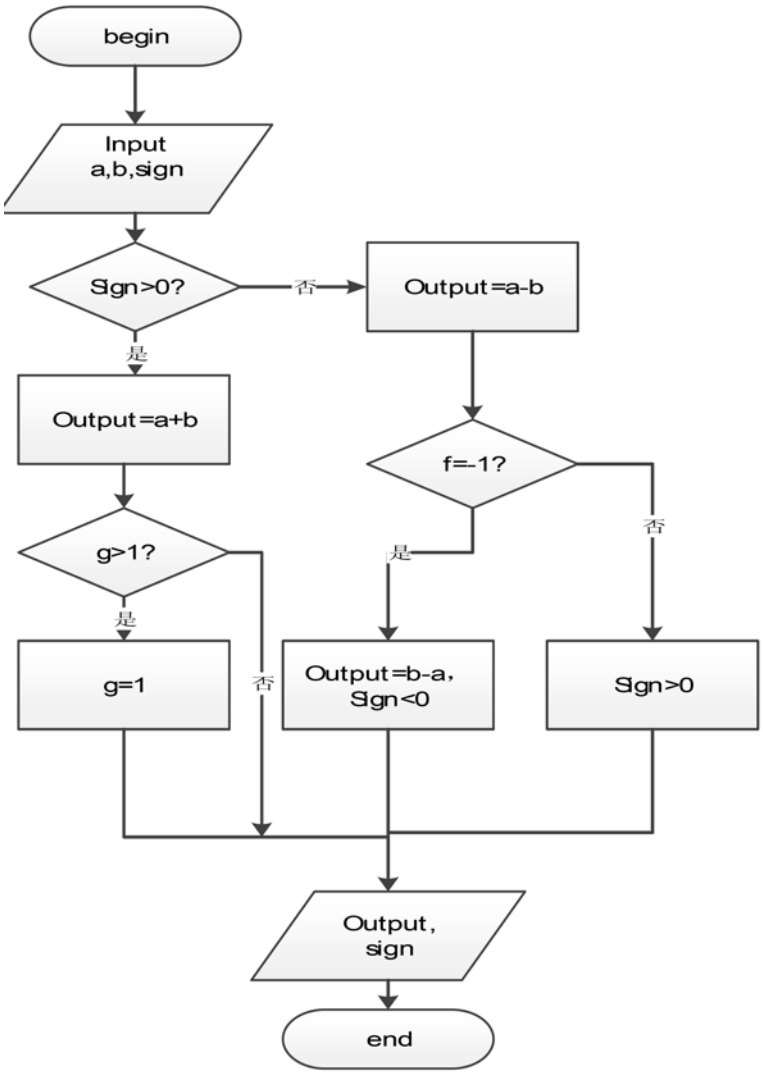

Figure 1. The flow chart of $a \oplus b$.

We has defined an operation $a \otimes b$ that dividing a into integer part called as int and decimal part called as fra and then converting int to a fraction if it is not zero. The computing method of P is following:

$$P = (N \times a1) / a2,$$

$$a1 / a2 = fra, \tag{6}$$

where N is the length of the b. B is generated by a random 0-1 code whose length is N and the number of '1' value is P. Make sure the operator. $\oplus$ times of b according to its integer part.

The basic flow chart is shown in Figure 2

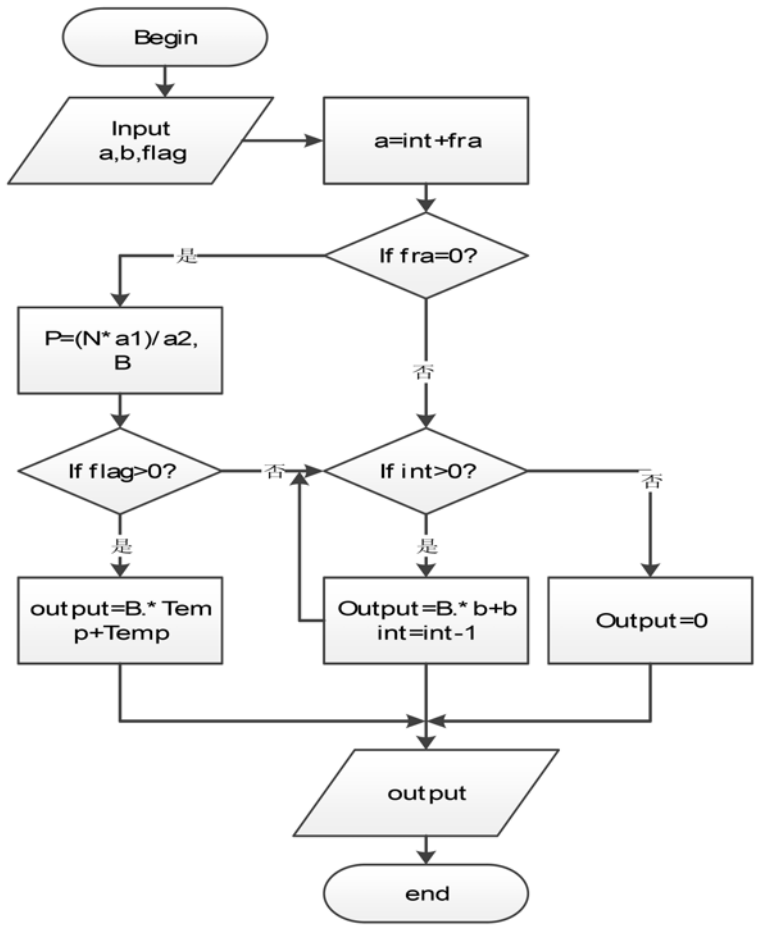

Figure 2. The flow chart of $a \otimes b$.

As comparing with the original algorithm, the ability of the adjusting illegal particle position into feasible region that can forbid behaviours of breaking boundary. Instead of using product between inertia weight and learning factor directly, the proposed method can random sample corresponding member of variable combining with the parameter fraction of PSO. With these new strategies, the random function of velocity updating formula will be briefer, and decrease the determinacy of product which can avoid immature convergence and improve the global search ability.

3. Preliminaries

3.1. Least squares support vector classification

Given a training set of N data points: $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$, where $\boldsymbol{x}_i \in R^p$ is the k-th input pattern and $y_i \in R$ is the k-th output pattern, the least squares support vector method classifier is introduced by formulating the classification problem as

$$\min_{w \in R^{n_h}, b \in R, \xi \in R^N} J(w, \boldsymbol{\xi}) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \boldsymbol{\xi}^T \boldsymbol{\xi}$$

$$s.t.\ y_i \left[w^T \varphi(\boldsymbol{x}_i) + b \right] = 1 - \xi_i, \quad i \in N_n. \tag{7}$$

One defines the Lagrangian function as

$$L(w, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) = \left(\frac{1}{2} w^T w + \gamma \frac{1}{2} \boldsymbol{\xi}^T \boldsymbol{\xi} \right)$$

$$- \sum_{i=1}^{N} \alpha_i \left\{ y_i \left[w^T \varphi(\boldsymbol{x}_i) + b \right] - 1 + \xi_i \right\}, \tag{8}$$

where α_i are the Lagrange multipliers which can be either positive or negative now due to the equality constraints as follows from the Kuhn-Tucker conditions and $\varphi : R^p \to R^{n_h}$ is the mapping function. $\boldsymbol{\xi} = (\xi_1, \xi_2, \cdots, \xi_N)$ is the vector composed of slack variable.

The conditions for optimality

$$\begin{cases} \dfrac{\partial L}{\partial w} = 0 \to w = \sum_{i=1}^{N} \alpha_i y_i \varphi(\boldsymbol{x}_i), \\ \dfrac{\partial L}{\partial b} = 0 \to \sum_{i=1}^{N} \alpha_i y_i = 0, \\ \dfrac{\partial L}{\partial \xi_i} = 0 \to \alpha_i = \gamma \xi_i,\ i = 1, \cdots, N, \\ \dfrac{\partial L}{\partial \alpha_i} = 0 \to y_i \left[w^T \varphi(\boldsymbol{x}_i) + b \right] = 1 - \xi_i. \end{cases} \tag{9}$$

Can be written immediately as the solution to the following set of linear equations:

$$\begin{bmatrix} 0 & \boldsymbol{Y}^T \\ \boldsymbol{Y} & \Omega + \gamma^{-1} \boldsymbol{I} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{1} \end{bmatrix}, \tag{10}$$

where $\Omega_{ij} = y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) = y_i y_j \varphi(\boldsymbol{x}_i)^T \varphi(\boldsymbol{x}_j)$, $\boldsymbol{Y} = [y_1; y_2; \ldots; y_N]$, $\mathbf{1} = [1; 1; \ldots; 1]$, $\boldsymbol{\alpha} = [\alpha_1; \alpha_2; \ldots; \alpha_N]$.

Given $\boldsymbol{H} = \boldsymbol{\Omega} + \gamma^{-1} \boldsymbol{I}$, $\boldsymbol{A} = \boldsymbol{Y}$, $\zeta_1 = b$, $\zeta_2 = \boldsymbol{\alpha}$, $d_1 = 0$, $\boldsymbol{d}_2 = \mathbf{1}$, $\boldsymbol{S} = \boldsymbol{A}^T \boldsymbol{H}^{-1} \boldsymbol{A}$. The solution is also given by

$$\begin{bmatrix} \boldsymbol{S} & 0 \\ 0 & \boldsymbol{H} \end{bmatrix} \begin{bmatrix} \zeta_1 \\ \boldsymbol{H}^{-1} \boldsymbol{A} \zeta_1 + \zeta_2 \end{bmatrix} = \begin{bmatrix} -d_1 + \boldsymbol{A}^T \boldsymbol{H}^{-1} \boldsymbol{d}_2 \\ \boldsymbol{d}_2 \end{bmatrix}. \tag{11}$$

The HS conjugate gradient method is used to solve the equations as $\boldsymbol{H}\eta = \boldsymbol{A}$, $\boldsymbol{H}\boldsymbol{\upsilon} = \boldsymbol{d}_2$ and solution is obtained by

$$b = \boldsymbol{S}^{-1} \eta *^T \boldsymbol{d}_2, \tag{12}$$

$$\boldsymbol{\alpha} = \boldsymbol{\upsilon} * - \eta * b. \tag{13}$$

Hence, the classifier function is found

$$y(\boldsymbol{x}) = sign\left[\sum_{i=1}^{N} \alpha_i y_i k(\boldsymbol{x}, \boldsymbol{x}_i) + b\right]. \tag{14}$$

3.2. Twin support vector machine

TWSVM is one of the new emerging machine learning approaches suitable for both classification and regression problems. The goal of TWSVM is to construct two non-parallel planes for each class by optimizing a pair of QPPs in such a manner that each hyper-plane is nearer to the data samples of one class while distant from the data samples of the other class. The linear TWSVM search for two non-parallel hyper-planes in as follows:

$$x^T w_{(1)} + b_{(1)} = 0, \qquad x^T w_{(2)} + b_{(2)} = 0, \tag{15}$$

where $A \in R^{d^+ \times m}$ represent the training data belong $+1$, $B \in R^{d^- \times m}$ represents the training data belong to the class -1.

Such that each hyper-plane is the closest to the one class training data of and the farthest from the training data of another class. A new data sample is assigned to class $+1$ or -1 depends on which of the two planes is closest to it. The linear TWSVM solves two QPPs with objective function corresponding to one class and constraints corresponding to the other class.

$$\min_{w_{(1)}, b_{(1)}} \frac{1}{2} \left\| A w_{(1)} + e_1 b_{(1)} \right\|^2 + c_1 e_2^T \lambda_2$$

$$s.t. \; -(B w_{(1)} + e_2 b_{(1)}) + \lambda_2 \geq e_2, \quad \lambda_2 \geq 0, \tag{16}$$

$$\min_{w_{(1)}, b_{(1)}} \frac{1}{2} \left\| B w_{(2)} + e_2 b_{(2)} \right\|^2 + c_2 e_1^T \lambda_1$$

$$s.t. \; (A w_{(2)} + e_2 b_{(2)}) + \lambda_1 \geq e_1, \quad \lambda_1 \geq 0, \tag{17}$$

where $c_1, c_2 > 0$ are penalty parameters, e_1 and e_2 are vectors of ones of appropriate dimensions, λ_1 and λ_2 are vectors of slack variables, respectively. Let $P = [B\ e_2]$ and $Q = [A\ e_1]$. The Wolf dual problems have been shown to be

$$\max_{\alpha} e_2^T \alpha - \frac{1}{2} \alpha^T P (Q^T Q)^{-1} P^T \alpha \tag{18}$$

$$s.t.\ 0 \le \alpha \le c_1 e_2,$$

$$\max_{\alpha} e_1^T \beta - \frac{1}{2} \beta^T Q (P^T P)^{-1} Q^T \beta \tag{19}$$

$$s.t.\ 0 \le \beta \le c_2 e_1,$$

where Lagrangian multipliers are $\alpha \in R^{m_2}$ and $\beta \in R^{m_1}$. In order to avoid the possible ill-conditioning of $P^T P$ and $Q^T Q$, TWSVM introduces a term $\varepsilon I (\varepsilon > 0)$, where I is an identity matrix of appropriate dimensions. The non-parallel hyper-planes can be obtained from the solutions α and β by

$$z_1 = -(Q^T Q + \varepsilon I)^{-1} P^T \alpha, \qquad z_2 = -(P^T P + \varepsilon I)^{-1} Q^T \beta, \tag{20}$$

where $z_{(i)} = [w_{(i)}^T\ b_{(i)}]^T$, $i = 1, 2$.

A new point $x \in R^m$ is assigned to class depending on which of the two hyper-planes in (1) is closer to, i.e.,

$$Class(i) = \arg\min_{i=1,2} \frac{\left| x^T w_{(i)} + b_{(i)} \right|}{\left\| w_{(i)} \right\|}. \tag{21}$$

3.3. Least squared twin *K*-class support vector classification

A least squares version of twin- KSVC called least squared twin *K*-class support vector classification (LST-KSVC) has recently been proposed. The matrix $A \in R^{l_1 \times m}$ represent the training data belong to $+1$, $B \in R^{l_2 \times m}$ represents the training data belong to the class -1, and $C \in R^{l_3 \times m}$ indicates the rest training data which are belong to 0.

The nonlinear LST-KSVC is considering the kernel generated surfaces as

$$K(x^T, D^T)u_{(1)} + \gamma_{(1)} = 0, \quad K(x^T, D^T)u_{(2)} + \gamma_{(2)} = 0, \tag{22}$$

where $D = [A; B; C]$ and K is an arbitrary kernel. The primal QPPs of the nonlinear LST-KSVC can be modified with 2-norm of slack variables and equality constraints corresponding to surfaces are given by

$$\min_{u_{(1)}, \gamma_{(1)}} \frac{1}{2}\left\|K(A, D^T)u_{(1)} + e_1\gamma_{(1)}\right\|^2 + \frac{c_1}{2} y^T y + \frac{c_2}{2} z^T z$$

$$s.t. -(K(B, D^T)u_{(1)} + e_2\gamma_{(1)}) + y = e_2$$

$$-(K(C, D^T)u_{(1)} + e_3\gamma_{(1)}) + z = e_3(1 - \varepsilon), \tag{23}$$

$$\min_{u_{(2)}, \gamma_{(2)}} \frac{1}{2}\left\|K(B, D^T)u_{(2)} + e_2\gamma_{(2)}\right\|^2 + \frac{c_3}{2} y^T y + \frac{c_4}{2} z^T z$$

$$s.t. (K(A, D^T)u_{(2)} + e_1\gamma_{(2)}) + y = e_1$$

$$(K(C, D^T)u_{(2)} + e_3\gamma_{(2)}) + z = e_3(1 - \varepsilon). \tag{24}$$

By substituting the constraints into the objective function, these QPPs become

$$\min_{u_{(1)}, \gamma_{(1)}} \frac{1}{2}\left\|K(A, D^T)u_{(1)} + e_1\gamma_{(1)}\right\|^2 + \frac{c_1}{2}\left\|K(B, D^T)u_{(1)} + e_2\gamma_{(1)} + e_2\right\|^2$$

$$+ \frac{c_2}{2}\left\|K(C, D^T)u_{(1)} + e_3\gamma_{(1)} + e_3(1 - \varepsilon)\right\|^2, \tag{25}$$

$$\min_{u_{(2)}, \gamma_{(2)}} \frac{1}{2}\left\|K(B, D^T)u_{(2)} + e_2\gamma_{(2)}\right\|^2 + \frac{c_3}{2}\left\|- K(A, D^T)u_{(2)} - e_1\gamma_{(2)} + e_1\right\|^2$$

$$+ \frac{c_4}{2}\left\|- K(C, D^T)u_{(2)} - e_3\gamma_{(2)} + e_3(1 - \varepsilon)\right\|^2. \tag{26}$$

The solution of QPPs above can be derived as

$$\begin{bmatrix} u_{(1)} \\ \gamma_{(1)} \end{bmatrix} = (c_1 N^T N + M^T M + c_2 O^T O)^{-1} (c_1 N^T e_5 + c_2 O^T e_6 (1 - \varepsilon)), \quad (27)$$

$$\begin{bmatrix} u_{(2)} \\ \gamma_{(2)} \end{bmatrix} = (c_3 M^T M + N^T N + c_4 O^T O)^{-1} (c_3 M^T e_4 + c_4 O^T e_6 (1 - \varepsilon)), \quad (28)$$

where $M = [K(A, D^T)e_1]$, $N = [K(B, D^T)e_2]$, and $O = [K(C, D^T)e_3]$.

In the case of nonlinear LST-KSVC, the corresponding decision function is designed as

$$f(x_i) = \begin{cases} +1, \text{ if } K(x_i, D^T)u_{(1)} + e\gamma_{(1)} > -1 + \varepsilon, \\ -1, \text{ if } K(x_i, D^T)u_{(2)} + e\gamma_{(2)} < 1 - \varepsilon, \\ 0, \quad \text{otherwise.} \end{cases} \quad (29)$$

In the "1-versus-1-versus-rest" structure, the proposed method constructs $K(K-1)/2$LST-KSVC classifiers for K-class classification. For a new testing point xi, a vote is given to the class (i) or class (j) based on which condition is satisfied.

4. Improved Least Squared Twin *K*-class Support Vector Classification

LST-KSVC is constructed by using empirical risk minimization principle due to which it suffers from the over-fitting problem. An improved LST-KSVC and performed experiment on KDD'99 dataset is proposed.

4.1. Linear improved LST-KSVC

For linearly separable data samples, the Improved LST-KSVC is given below:

$$\min_{u_{(1)},\gamma_{(1)}} \frac{1}{2}\left\|Au_{(1)} + e_1\gamma_{(1)}\right\|^2 + \frac{c_1}{2} y^T y + \frac{c_2}{2} z^T z$$

$$+ c_3(\left\|u_{(1)}\right\|^2 + \gamma_{(1)}{}^2)$$

$$s.t. -(Bu_{(1)} + e_2\gamma_{(1)}) + y = e_2$$

$$-(Cu_{(1)} + e_3\gamma_{(1)}) + z = e_3(1-\varepsilon), \tag{30}$$

$$\min_{u_{(2)},\gamma_{(2)}} \frac{1}{2}\left\|Bu_{(2)} + e_2\gamma_{(2)}\right\|^2 + \frac{c_4}{2} y^T y + \frac{c_5}{2} z^T z$$

$$+ \frac{c_6}{2}(\left\|u_{(2)}\right\|^2 + \gamma_{(2)}{}^2)$$

$$s.t. (Au_{(2)} + e_1\gamma_{(2)}) + y = e_1$$

$$(Cu_{(2)} + e_3\gamma_{(2)}) + z = e_3(1-\varepsilon). \tag{31}$$

The additional term in above the equations is for measuring the separation of two hyper-planes. Larger separation indicates good generalization capability.

4.2. Non-linear improved LST-KSVC

Improved LST-KSVC is also extended for non-linearly separable data samples by using kernel function. Improved LST-KSVC for non-linear data samples is formulated as

$$\min_{u_{(1)},\gamma_{(1)}} \frac{1}{2}\left\|K(A, D^T)u_{(1)} + e_1\gamma_{(1)}\right\|^2 + \frac{c_1}{2} y^T y + \frac{c_2}{2} z^T z$$

$$+ c_3(\left\|u_{(1)}\right\|^2 + \gamma_{(1)}{}^2)$$

$$s.t. -(K(B, D^T)u_{(1)} + e_2\gamma_{(1)}) + y = e_2$$

$$-(K(C, D^T)u_{(1)} + e_3\gamma_{(1)}) + z = e_3(1-\varepsilon), \tag{32}$$

$$\min_{u_{(2)},\gamma_{(2)}} \frac{1}{2}\left\|K(B, D^T)u_{(2)} + e_2\gamma_{(2)}\right\|^2 + \frac{c_4}{2} y^T y + \frac{c_5}{2} z^T z$$

$$+ \frac{c_6}{2}(\|u_{(2)}\|^2 + \gamma_{(2)}{}^2)$$

$$s.t.\ (K(A, D^T)u_{(2)} + e_1\gamma_{(2)}) + y = e_1$$

$$(K(C, D^T)u_{(2)} + e_3\gamma_{(2)}) + z = e_3(1-\varepsilon). \tag{33}$$

The Sherman-Morrison-Woodbury (SMW) formula is used to reduce the computation cost as:

$$\begin{bmatrix} u_{(1)} \\ \gamma_{(1)} \end{bmatrix} = -(Z - ZN(\frac{I}{c_1} + NZN^T)^{-1}NZ)$$

$$\times (c_1 N^T e_5 + c_2 O^T e_6(1-\varepsilon)), \tag{34}$$

$$\begin{bmatrix} u_{(2)} \\ \gamma_{(2)} \end{bmatrix} = -(F - FM^T(\frac{I}{c_4} + MFM^T)^{-1}MF)$$

$$\times (c_4 M^T e_4 + c_5 O^T e_6(1-\varepsilon)), \tag{35}$$

where $Z = (M^T M + c_2 O^T O + c_3 I)^{-1}$ and $F = (N^T N + c_5 O^T O + c_6 I)^{-1}$ using SMW formula as

$$Z = \frac{1}{c_2}(Y_1 - Y_1 M^T(c_2 I + MY_1 M^T)^{-1}MY_1, \tag{36}$$

$$F = \frac{1}{c_5}(Y_2 - Y_2 N^T(c_5 I + NY_2 N^T)^{-1}NY_2. \tag{37}$$

Here $Y_1 = (O^T O + \frac{c_3}{c_2} I)^{-1}$, $Y_2 = (O^T O + \frac{c_6}{c_5} I)^{-1}$. The regularization term $\frac{c_3}{c_2} I$ and $\frac{c_6}{c_5} I$ is used to take care of problems due to the possible ill-conditioning of Y_1 and Y_2.

$$Y_1 = \frac{c_2}{c_3}(I - O^T(\frac{c_3}{c_2}I + OO^T)^{-1}O), \tag{38}$$

$$Y_2 = \frac{c_5}{c_6}(I - O^T(\frac{c_6}{c_5}I + OO^T)^{-1}O). \tag{39}$$

5. Numerical Experiments

5.1. The KDD'99 datasets

KDD'99 datasets are used to show the ability of Improved LST-KSVC. All experiments except those shown in Table 1 and Table 2, have been implemented in Matlab R2012a on a PC with system configuration Intel(R)core(TM) i3-2130 CPU at 2.53GHz with 4GB of RAM, and Windows 7 operating system.

The KDD'99 contest dataset extracted from the 1998 DARPA Intrusion Detection Evaluation Program. It is contains 494021 connection records for training and 311029 connection records for testing. Each connection record contains 7 discrete and 34 continuous features for a total of 41 features. The KDD'99 contest builds effective classifier to distinguish the four kinds of attack connections and normal connections. It contains four major categories of attacks: Probing (PROBE) attacks, Denial-of-Service (DOS) attacks, User-to-Root (U2R) attacks, and Remote-to-Local (R2L) attacks [16].

Discretization is typically applied to attributes that are used in classification or association analysis. The modified entropy minimization discretization (EMD) is used on the KDD'99 dataset, since entropy-based approaches are one of the most promising approaches to discretization.

The normal, DOS, R2L and probe class data are selecting 700, 950, 200, and 150 data samples of each class by randomly. Because of the U2R subset is too small so all of them should be used. The selected dataset are randomly divided into fifty-fifty. One of them is used for training, while another one is used to test. And then get the model parameters and the accuracy of the proposed detection model.

5.2. Result comparisons and discussion

The binary particle swarm optimization (PSO) algorithm and the proposed new particle swarm algorithm (NBPSO) are used to search parameters and feature subset. And its effect is compared with eight models composed of the following four categories: LS-SVC, LSTK-SVC, ILST-SVC, and ILSTK-SVC [17].

Different classification model employ the same least square (LS) loss function and classification solution is learned by two linear equations. As it is shown in Table 1 and Table 2, NBPSO is able to reduce the features of classifiers while the accuracy of the classifiers still performance better result. To further show the advantage of proposed improved LST-KSVC in the precision of classification model, the accuracy of improved LST-KSVC have been compared with the LST-KSVC on the KDD'99 datasets. Table 2 shows the different prediction accuracy of each classifier. Although the method of selecting the parameters and features have been fixed in advance, but results confirm that using the improved LST-KSVC makes better performance of classifiers.

Table 1. The classification of 1-versus-1

Classifier	PSO LS-SVC Acc(%) Feature Cnt	PSO ILST-SVC Acc(%) Feature Cnt	NBPSO LS-SVC Acc(%) Feature Cnt	NBPSO ILST-SVC Acc(%) Feature Cnt
Normal	97.56	97.47	97.66	98.54
	41	41	13	14
PROBE	97.95	98.34	98.34	98.15
	41	41	18	15
DOS	99.51	99.71	99.90	99.61
	41	41	13	21
U2R	99.12	99.32	99.03	99.22
	41	41	17	14
R2L	97.66	98.05	98.54	98.15
	41	41	17	17

Table 2. The classification of 1-versus-1-versus-rest

Classifier	PSO LSTK-SVC Acc(%) Feature Cnt	PSO ILSTK-SVC Acc(%) Feature Cnt	NBPSO LSTK-SVC Acc(%) Feature Cnt	NBPSO ILSTK-SVC Acc(%) Feature Cnt
Normal + PROBE	96.00	96.00	97.33	97.33
	41	41	13	15
Normal + DOS	99.27	99.88	99.76	99.88
	41	41	21	19
Normal + U2R	98.35	98.59	96.24	97.18
	41	41	18	20
Normal + R2L	93.88	94.68	94.15	96.28
	41	41	18	21
PROBE + DOS	99.65	99.48	99.83	99.83
	41	41	18	22
PROBE + U2R	96.57	96.57	96.57	99.43
	41	41	21	13
PROBE + R2L	84.13	82.54	94.44	97.62
	41	41	16	18
DOS + U2R	98.91	99.27	99.82	100.00
	41	41	16	19
OS + R2L	96.81	97.60	99.80	100.00
	41	41	14	18
U2R + R2L	84.16	81.19	87.13	91.09
	41	41	14	13

6. Conclusion

NBPSO is a rapid and effective intelligent algorithm based on ecological phenomenon. In this paper, a new binary version of PSO called as NBPSO is formulated for optimizing the multi-class classification. This formulation leads to selecting the parameters and features of classification model extremely simple and effective algorithm. New binary algorithm not only has good accuracy in the classification of 1-versus-1,

but also has good precision in the classification of 1-versus-1-versus-rest. Improved LST-KSVC, similar to the LST-KSVC, evaluates the KDD'99 dataset into a novel "1-versus-1-versus-rest" structure that generates ternary output $\{-1, 0, +1\}$. Computational results on datasets demonstrate that improved LST-KSVC obtains classification accuracy comparable to the others by reduced features. The number of the parameters in the model increase is a practical problem and should be addressed in the future.

References

[1] J. Saranya and Dr. G. Padmavathi, A brief study on different intrusions and machine learning-based anomaly detection methods in wireless sensor networks, Advanced Networking and Applications 6(4) (2015), 2414-2421.

[2] Levent Koc, Thomas A. Mazzuchi and Shahram Sarkani, A network intrusion detection system based on a Hidden Naive Bayes multiclass classifier, Expert Systems with Application 39 (2012), 13492-13500.

[3] H. Liu, H. Motoda, R. Setiono and Z. Zhao, Feature selection: An ever evolving frontier in data mining, In Proc. The Fourth Workshop on Feature Selection in Data Mining 4 (2010), 4-13.

[4] Lan Bai, Zhen Wang, Yuan-Hai Shao and Nai-Yang Deng, A novel feature selection method for twin support vector machine, Knowledge-Based Systems, Elsevier 59 (2014), 1-8.

[5] C. Cortes and V. Vapnik, Support vector network, Mach. Learn. 20(3) (1995), 273-297.

[6] V. Vapnik, The Nature of Statistical Learning, 2nd Edition, New York: Springer, 1998.

[7] Jayadeva, R. Khemchandani and S. Chandra, Twin support vector machine for pattern classification, IEEE Trans. Pattern Anal. Mach. Intell. 29(5) (2007), 905-910.

[8] M. A. Kumar and M. Gopal, Least squares twin support vector machines for pattern classification, Expert Syst. Appl. 36 (2009), 7535-7543.

[9] D. Tomar and S. Agarwal, Feature selection based least square twin support vector machine for diagnosis of heart disease, Int. J. Bio.-Sci. Bio-Technol. 6(2) (2014).

[10] J. A. Nasiri, N. M. Charkari and S. Jalili, Least squares twin multi-class classification support vector machine, Pattern Recogn. 46 (2015), 984-992.

[11] J. He and S. H. Zheng, Intrusion detection model with twin support vector machines, J. Shanghai Jiaotong Univ. (Sci.) 19 (2014), 448-454.

[12] X. Ding, G. Zhang, Y. Ke, B. Ma and Z. Li, High efficient intrusion detection methodology with twin support vector machines, In: International Symposium on Information Science and Engineering, 2008, ISISE'08, Vol. 1, IEEE (2008), 560-564.

[13] C. Tsai, Y. Hsu, C. Lin and W. Lin, Intrusion detection by machine learning: A review, Expert Systems with Applications 36(10) (2009), 11994-12000.

[14] Y. Xu, R. Guo and L. Wang, A twin multi-class classification support vector machine, Cogn. Comput. 5(4) (2013), 580-588.

[15] Y. Xu, W. Xi, X. Lv and R. Guo, An improved least squares twin support vector machine, J. Inform. Comput. Sci. 9 (2012), 1063-1071.

[16] V. Bolon-Canedo, N. Sanchez-Marono and A. Alonso-Betanzos, Feature selection and classification in multiple class datasets: An application to KDD Cup 99 dataset, Expert Systems with Applications 38(5) (2011), 5947-5957.

[17] F. Kuang, W. Xu and S. Zhang, A novel hybrid KPCA and SVM with GA model for intrusion detection, Appl. Soft Comput. 18 (2014), 178-184.

■