Neurocomputing 199 (2016) 90–102

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization

Seyed Mojtaba Hosseini Bamakan [a,b], Huadong Wang [a], Tian Yingjie [a], Yong Shi [a,b,c,*]

[a] Key Laboratory of Big Data Mining and Knowledge Management, University of Chinese Academy of Sciences, Beijing 10090, China
[b] School of Economics and Management, University of Chinese Academy of Sciences, Beijing 10090, China
[c] College of Information Science and Technology, University of Nebraska at Omaha, 68182 NE, USA

ARTICLE INFO

Article history:
Received 21 October 2015
Received in revised form
20 February 2016
Accepted 2 March 2016
Communicated by: K. Chan
Available online 20 April 2016

Keywords:
Intrusion detection
Support vector machine
Parameter setting
Feature selection

ABSTRACT

Many organizations recognize the necessities of utilizing sophisticated tools and systems to protect their computer networks and reduce the risk of compromising their information. Although many machine-learning-based data classification algorithm has been proposed in network intrusion detection problem, each of them has its own strengths and weaknesses. In this paper, we propose an effective intrusion detection framework by using a new adaptive, robust, precise optimization method, namely, time-varying chaos particle swarm optimization (TVCPSO) to simultaneously do parameter setting and feature selection for multiple criteria linear programming (MCLP) and support vector machine (SVM). In the proposed methods, a weighted objective function is provided, which takes into account trade-off between the maximizing the detection rate and minimizing the false alarm rate, along with considering the number of features. Furthermore, to make the particle swarm optimization algorithm faster in searching the optimum and avoid the search being trapped in local optimum, chaotic concept is adopted in PSO and time varying inertia weight and time varying acceleration coefficient is introduced. The performance of proposed methods has been evaluated by conducting experiments with the NSL-KDD dataset, which is derived and modified from well-known KDD cup 99 data sets. The empirical results show that the proposed method performs better in terms of having a high detection rate and a low false alarm rate when compared with the obtained results using all features.

1. Introduction

Nowadays, computer networks and the Internet has become an essential part of any organization to survive in the technological world. Not only become organizations more depended to computer networks to do their daily business such as online services, but also customers share and store their personal information in these networks [1,2]. Cyber-security is a comprehensive topic that has significant impacts on each entity consist of organizations, customers and also governments [3]. Hence, organizations need to be more concerned to protect their business from potential attacks or abnormal activities aimed at compromising their networks. Although a wide range of mechanisms such as firewall, user authentication, antivirus and data encryption, and intrusion detection systems has been designed to protect computer networks [1], because of increasing and sophisticated nature of new attacks, these traditional intrusion detection methods have failed to completely protect them [4].

As mentioned in Liao et al. [5], any attempts to compromise the three most crucial components of security, i.e., confidentiality, integrity and availability (CIA), or penetrate through the security mechanisms of computer networks considered as an intrusion [5]. According to the IDSs taxonomy [1,6,7], IDSs based on their detection methodologies are categorized as signature-based detection (misuse detections) and anomaly-based detection. In the signature-based detection systems, well-known intrusions will be detected by comparing the observed data with pre-defined attack's patterns. In fact, misuse detection methods work properly based on the known attack signatures. Although the detection rate of well-known intrusions is high with this method, however misuse detection methods face with high false positive rate because of the continuously changing nature of intrusions [6,7]. On the other hand, anomaly detection methods are mainly based on hypothesizes that abnormal behavior is diverse from normal behavior. Therefore, any deviation from the normal behavior

[*] Corresponding author at: Key Laboratory of Big Data Mining and Knowledge Management, University of Chinese Academy of Sciences, Beijing 100090, China. Tel.: +86 10 8268 0697.
E-mail addresses: yshi@ucas.ac.cn, yshi@unomaha.edu (Y. Shi).

http://dx.doi.org/10.1016/j.neucom.2016.03.031
0925-2312/© 2016 Elsevier B.V. All rights reserved.

considered as abnormal or intrusions. Since, anomaly-based IDSs need to build a model based on normal patterns, they have the capability of detecting unknown intrusions [5].

Although the results achieved by different researchers show promising improvements in the IDSs detection approaches, the intrusion detection problem is an ongoing research area since we have to deal with the huge volume of network traffic data, high dimensional training dataset, constantly changes in environments and need for real-time detection [6–8]. For instance, the high dimensionality of training set which some of its features are irrelevant, redundant or highly correlated will affect the performance of IDSs. Besides the importance of choosing the most suitable subset of features to design a detection model, setting the parameters of applied algorithms with the optimal value is another influential factor, which impresses the model [9,10].

In this paper, we proposed an effective intrusion detection framework based on time-varying chaos particle swarm optimization combined with multiple criteria linear programming (MCLP) and support vector machine (SVM) to provide an adaptive, robust, precise methodology to detect intrusions; it named TVCPSO–MCLP and TVCPSO–SVM. The performance of proposed methods has been evaluated and compared by conducting different experiments with the NSL-KDD dataset, which derived and modified from well-known KDD cup 99 data sets. As mentioned in Wu and Banzhaf [6], a robust IDS should has a high detection rate (DR) and a low false alarm rate (FAR). Even though most of intrusion detection methods have high DR, they suffer from higher FAR, thus in this paper, we proposed a weighted objective function to simultaneously take into account the detection rate along with the false alarm rate and also the number of features to maximize the effectiveness of proposed methods. Totally, the feasibility and efficiency of these two methods compared based on three metrics include accuracy, detection rate and false alarm rate. Furthermore, performance of proposed time-varying chaos particle swarm optimization in parameter setting and feature selection for MCLP and SVM is compared with the standard particle swarm optimization (PSO) and chaos particle swarm optimization (CPSO).

The main contributions of this paper include the following:

(1) Modifications to the chaos particle swarm optimization have been proposed by adopting the time-varying inertia weight factor (TVIW) and time-varying acceleration coefficients (TVAC), namely TVCPSO, to make it faster in searching for the optimum and avoid the search being trapped into local optimum.
(2) A weighted objective function that simultaneously takes into account trade-off between the maximizing the detection rate and minimizing the false alarm rate, along with considering the number of features is proposed to eliminate the redundant and irrelevant features, as long as increase the attack's detection rate.
(3) An extended version of multiple criteria linear programing, namely PMCLP, has been adopted to increase the performance of this classifier in dealing with the unbalance intrusion detection dataset.
(4) The proposed TVCPSO has been adopted to provide an effective IDS framework by determining parameters and selecting a subset of features for multiple criteria linear programming and support vector machines.

The remainder of this paper is organized as follows. In Section 2, we present the related works. Section 3 gives a brief overview of SVM, MCLP, PSO, binary PSO and CPSO then in Section 4, explanation of proposed TVCPSO–SVM and TVCPSO–MCLP is presented in details. The experimental results and in depth comparisons of proposed methods are presented in Section 5. Finally, we conclude and provide future works in Section 6.

2. Related works

For the last decades, many researchers considered intrusion detection as a classification problem. They proposed different methodologies, by using data mining techniques including, Support Vector Machines (SVM) [11], Decision Trees [12], K-nearest Neighbor (K-NN) [13], Naïve Bayes networks [14,15] and Artificial Neural Networks [16], which we can find a short review on intrusion detection by machine learning in Tsai et al. [1]. Later, by achieving promising results by applying computational intelligence (CI) techniques in classification problems, many researchers focus on proposing new methodologies based on CI. A review of using these approaches was given by Wu and Banzhaf [6].

Much more recent researches on this domain have been focused on improving the performance of data mining techniques by combining them with swarm intelligence optimization techniques. For example, Chung and Wahid [4] proposed a hybrid feature selection and classification methodology by using dynamic swarm based rough set and simplified swarm optimization (SSO). Chung and Wahid [4] enhanced the performance of SSO to find a better solution from the neighborhood by using the weighted local search (WLS) strategy. Their results show this method achieved the 93.3% accuracy in classifying intrusions. Another hybrid methodology proposed by Kuang et al. [17] for intrusion detection by combining multi-layered SVM with kernel principal component analysis (KPCA) and genetic algorithm (GA) to increase the accuracy of the model. KPCA is applied to reduce the dimension of features set and to decrease the training time. As stated in Lin et al. [18], there are 41 features in the KDD 99 dataset which some of these features might have no effect or have high levels of noise. Hence, they proposed an intelligent algorithm based on support vector machine, decision tree and simulated annealing (SA) to find the suitable feature subset and increase the accuracy of anomaly intrusion detection. Among the aforementioned machine learning methods, SVM mentioned as an effective one in intrusion detection problem [19].

Support vector machine is a well-known classification technique because of its powerful generalization capability which is based on the structural risk minimization (SRM). In the recent years, some other new models were proposed based on the standard SVM, for example Bounded SVM [20], v-SVM [21], least squares SVM [22], Twin SVM [23], NPSVM [24] and nearly-isotonic SVM [25]. In the context of detecting intrusion by data mining techniques, SVMs were one of the most widely techniques. In Kim et al. [12] the authors proposed a hybrid misuse detection and anomaly detection system which C4.5 decision tree algorithm is utilized in the misuse detection part and the anomaly detection part is built based on multiple one-class SVM models. In order to deal with non-imbalanced class distributions [26], proposed an autonomous labeling approach to support vector machine algorithms. Their method is based on SNORT and the main idea of this work is that by excluding the well-known attacks from the dataset, it would improve the performance of SVM for novelty detection. Li et al. [19] takes into account the importance of feature selection on efficiency of the intrusion detection. Hence, they proposed a method based on gradually feature removal combined with SVM and ant colony algorithm. Training a model is a time consuming process in large scale intrusion detection datasets that makes SVMs inefficient in some cases. Gan et al. [27] considered this drawback of SVMs and proposed a methodology based on the Core Vector Machine (CVM) and Partial Least Square (PLS) feature extraction to increase the training speed and detection capability of SVM in large scale datasets. Generally speaking SVMs can be classified as L1-SVM and L2-SVM according to the loss function that core vector machine (CVM) is modified version of the L2-SVM algorithm which show outstanding results in pattern recognition problems with large-scale sample data and complex nonlinearity [27,28] considered the difficulty of real-time anomaly detection in wireless sensor networks by utilizing the advantages of ellipsoidal one-class SVM to make a distributed and online outlier detection techniques. Candidate Support Vectors (CSV) and Candidate Support Vector based Incremental SVM (CSV-ISVM) are the approaches proposed by Chitrakar and Huang [29] to select and retain non-support vectors of the current increment of classification throughout the whole learning process. As we reviewed the related works in intrusion detection methods based on support vector machines, there is issue that received less attention from researchers in this field and that is, the generalization power of support vector machine is strongly depended on well setting of its parameters [10,30]. Moreover, the dimension of the input space is a critical factor which will degrade the performance of the SVM, if it is a large scale and non-clean input [10]. It makes clear the importance of feature selection process which means choosing the most appropriate subset of features. It will decrease the classification computational complexity, and improve the accuracy of classifier by avoiding redundant or irrelevant features [31]. In intrusion detection context, the datasets are not only known for their huge volume and high dimensionality, but also some of the feature space inputs are not important as others. Hence, by reducing the feature set domain, besides increasing the training process speed and decreasing the storage demands, better understanding and interpretability of the results can be gained [32,33]. By considering the drawbacks of aforementioned methodologies, in this paper an effective intrusion detection framework with a multi-objective function which do the parameter setting and feature selection simultaneously has been proposed.

In order to address the parameters setting and providing the most appropriate subset of features simultaneously for SVM with particle swarm optimization method, a number of works have been done. For example, in Lin et al. and Huang and Dun [10,30], the authors developed a hybrid method, namely; PSO–SVM, to set the kernel parameters of SVM in training procedure, along with the feature selection to increase the classification accuracy. According to them, their approaches are an optimization mechanism, which simultaneously combined the binary PSO for the feature selection part with the continuous-valued PSO for the SVM kernel parameter setting. Recently, Chen et al. [9] made some improvements in PSO–SVM proposed by Lin et al. and Huang and Dun [10,30]. Although the main purpose of this research is to propose a hybrid PSO-SVM for proper parameter settings of support vector machine and feature selection, their modifications on the previous methods caused reducing the computational time and giving higher predictive accuracy. The modifications proposed in Chen et al. [9] consist of introducing a weighted fitness function, applying mutation operators and improving the binary PSO algorithm. The authors implemented the proposed method in parallel environment using Parallel Virtual Machine (PVM), then evaluated their method in some benchmark datasets. As reviewed in aforementioned works, although the combination of particle swarm optimization with SVM for parameter setting and feature set selection, shows an acceptable performance in classification problems, its performance in intrusion detection problem is not examined. In our previous work Bamakan et al. [34], we presented the initial idea of hybridizing MCLP and PSO for a binary class classification, but to the best of our knowledge, there not exists other study available for setting the parameters of multiple criteria linear programming which proposed and extended by Shi et al. [35], along with selecting the most appropriate set of features simultaneously in the intrusion detection problem.

The proposed methodology is a flexible framework which can apply to a variety of practical classification problems which need to classify some groups of objects and in some cases to find the most relevant subset of features such as Bioinformatics problems [36], Environmental Sciences [37], spam detection [38], customer churn analysis [39], credit scoring [40] and etc. In this paper the proposed method has been considered in the context of the intrusion detection problem. It should be noted that because of resources and technical limitations, using high-performance computing techniques, for example parallelization did not address in this research. However, because of the iterative nature of the meta-heuristic optimization techniques such as particle swarm optimization, genetic algorithm, ant colony optimization and some other similar methods, they need more computational time, which some researchers claimed that implementing these kind of algorithms on a parallelized environment will significantly increase the speed of them [9,30].

3. Background

3.1. Support vector machine

The support vector machine is a machine learning derived and linear classifier that in a two groups classification problem, a maximum hyper plane separates a class of positive samples from a class of negative samples based on structural risk minimization principle [41]. If the original data is not linearly separable, a nonlinear kernel function can be used to this problem [42,43]. In fact, the goal of SVM is to find an optimal separating hyper-plane by maximizing the margin between the separating hyper-plane and the closest data points of the training set [42,43].

Let's suppose a training set $T=\{(x_1,y_1),\ ....,\ (x_l,y_l)\}\in(R^n\times\{-1,1\})^l$ where $x_i\in([x_i]_1,\ldots,[x_i]_n)^T$ is the input attribute vectors, $y_i\in\{-1,+1\}$ is the corresponding output of x_i, n is the sample number, in the feature space, the classification function of SVM is $f(x)=w^T.x+b$, b is the bias and w is a weight vector of the same dimension as the feature space. By adjusting the b and w, we can determine the position of the separating hyper-plane. With respect to the maximizing the margin, the optimization problem can be defined as following:

$$\begin{array}{ll} min_{w,b,\xi} & \frac{1}{2}||w||^2+C\sum_{i=1}^{l}\xi_i \\ s.t. & y_i\left((w^T.x_i)+b\right)\geq 1-\xi_i,\quad i=1,2,\ldots,\ l \\ & \xi_i\geq 0,\ i=1,2,\ldots,\ l \end{array} \tag{1}$$

where the constant $C>0$ determines the trade-off between margin maximization and training error minimization and slack variable ξ_i defined as some noises that cause the overlap of the classes [35]. The above model is a classical convex optimization problem. By considering the Kuhn–Tucker condition, the above model converts to the following dual problem by introducing Lagrangian multipliers a_i:

$$\begin{array}{ll} min_a & \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}y_iy_ja_ia_j(x_i^T.x_j)-\sum_{j=1}^{l}a_j \\ s.t. & \sum_{i=1}^{l}y_ia_i=0 \\ & 0\leq a_i\leq C,\ i=1,\ldots,l \end{array} \tag{2}$$

Suppose $a^*=(a_1^*,\ldots,\ a_l^*)^T$ is a solution of the dual problem. According to the following equation the solution (w^*,b^*) of primal problem is given by [35,44]:

$$w^*=\sum_{i=1}^{l}a_i^*y_ix_i \tag{3}$$

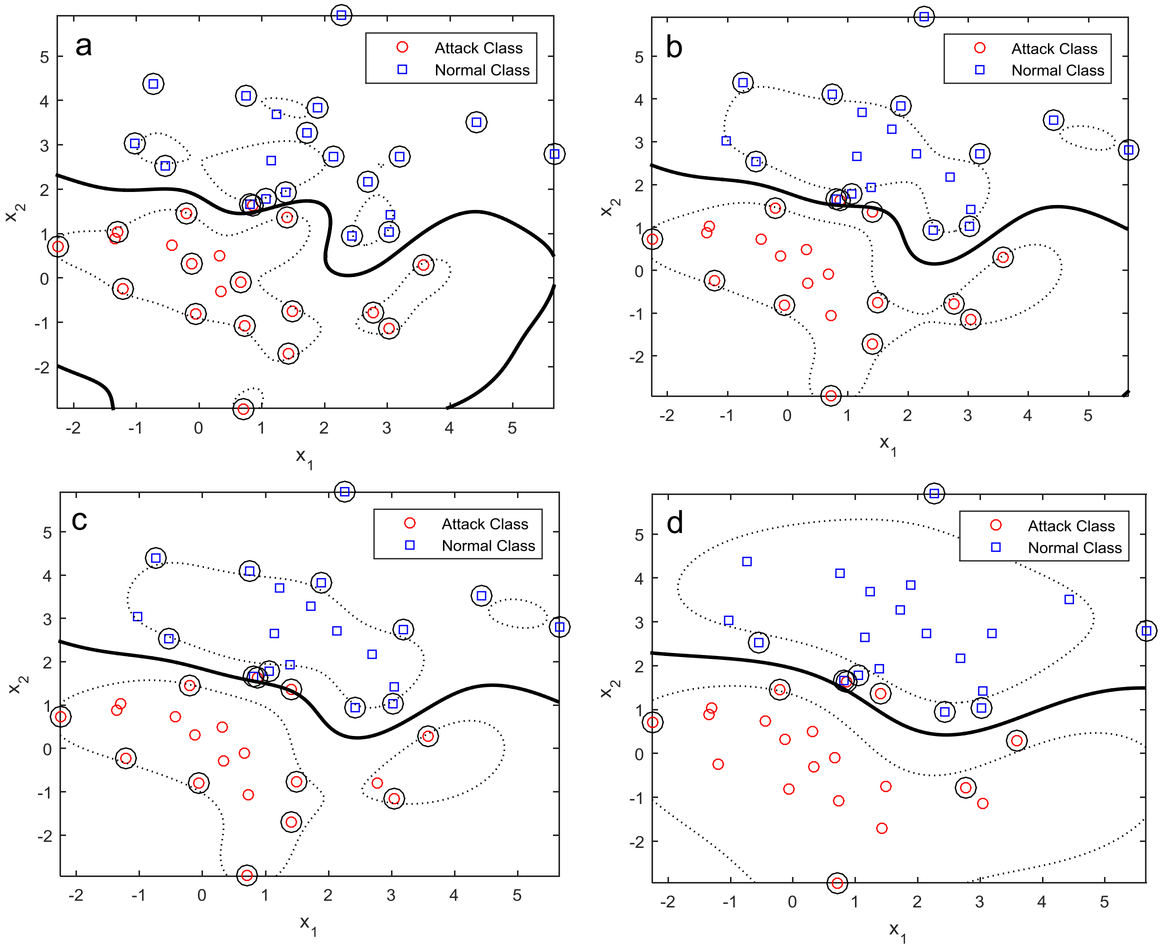


Fig. 1. Effect of different γ value on the decision boundary of the SVM with RBF kernel. Here $C=5$ and in a) $\gamma=0.65$ b) $\gamma=0.9$ c)$\gamma=1.0$ d) $\gamma=1.7$.

$$b^* = y_j - \sum_{i=1}^{l} y_i a_i^* (x_i^T . x_j) \quad a_i^* \in (0, C) \tag{4}$$

In order to, enhance the performance of SVMs in non-linear cases, some kernel functions $K(x_i . x_j)$ have been proposed to map $(x_i^T . x_j)$ in the original input space to $\varphi(x_i)^T \ \varphi(x_j)$ in some high-dimensional feature space. Polynomial kernel, $K(x_i . x_j) = (1+x_i^T . x_j)^p$ and Radial-Basis function (RBF) or Gaussian kernel, $K(x_i . x_j) = e^{-\gamma||x_i - x_j||^2}$, are the two most widely used kernels in SVM, where p is the polynomial order and γ is the predefined parameter controlling the width of the Gaussian kernel. In this study, we adopted Gaussian Kernel in that parameters C and γ should be perfectly optimized because of their significant effect on the decision boundary of SVM. As shown in Fig. 1, different values for γ will directly affect the flexibility of separating hyper-plane and it is caused shifting in the resulting decision boundary. For example, the decision boundary of the SVM model tends to be highly sensitive to the training data, if the value of γ be a small number, in contracts, in this case by choosing the bigger number, the decision boundary of the SVM tends to be smoother. Consequently, tuning the parameters of SVM with an optimal value has a great effect on the performance of this classifier.

By considering $f(x)$ as a generic test instance, the linear discriminant function can be formulated as follows:

$f(x) = sgn \sum_{i=1}^{l} (a_i y_i K(x_i . x_j) + b)$, if $f(x) = +1$ it will consider the test instance, as a normal traffic network and $f(x) = -1$ it will classify the test instance as an attack.

3.2. Multiple criteria linear programming

Linear programming classification method (LP) was first proposed by Fred Glover in the 1980's [45]. In 1990's Shi et al. [46] developed multiple criteria linear programming (MCLP) and multiple criteria quadratic programming (MCQP) classification models, which have been successfully used in classification problems. The objectives of the initial form of the linear programming classification method are to maximize the minimum distance (MMD) of observations from the critical value and minimizing the sum of the distance (MSD) of the observations from the critical value [47], consider the Fig. 2, the overlapping of data α should be minimized while the distance β has to be maximized. However, it is difficult for traditional linear programming to optimize MMD and MSD simultaneously [48–50].

The first multiple criteria linear programming (MCLP) model is formulated as follows:

$$\begin{array}{lll} min & & \sum_{i=1}^{n} \alpha_i \\ max & & \sum_{i=1}^{n} \beta_i \\ s.t. & (w^T . x_i) = b + y_i \ (\alpha_i - \beta_i), \ i = 1, \ldots, n \\ & & \alpha, \ \beta \geq 0 \end{array} \tag{5}$$

here α_i is the overlapping and β_i is the distance from the training sample x_i to the discriminator $(w^T . x_i) = b$. If $y_i \in 1, -1$ denotes the label of x_i and n is the number of samples, a training set can be interpreted as a pair $\{x_i, \ y_i\}$, here x_i are the vector values of the variables and $y_i \ \in \{1, -1\}$ is the label in the classification case. The weight vector w and the bias b are the unknown variables to be optimized for the two objectives.

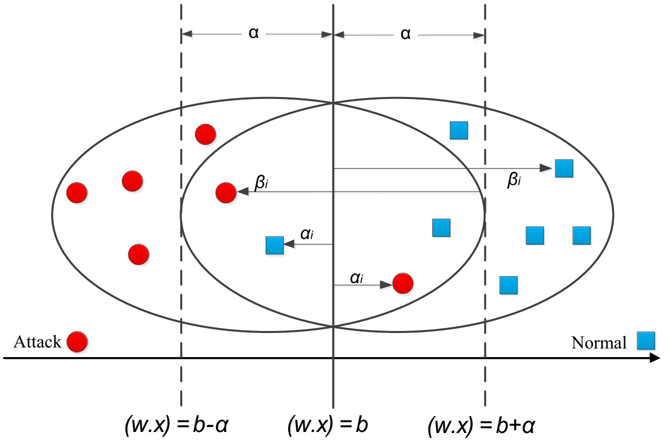


Fig. 2. Illustration of MCLP for two-class problem.

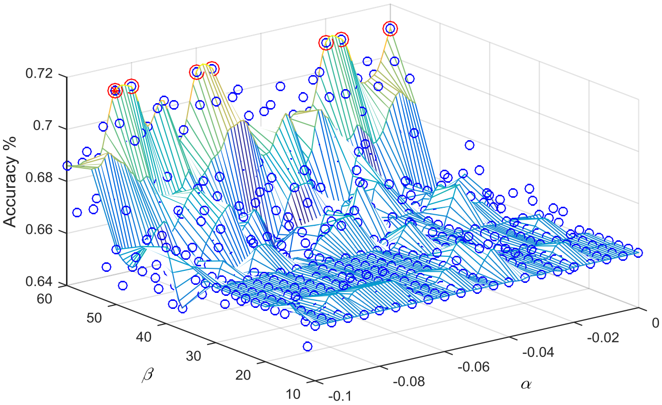


Fig. 3. Illustration of impact of different values for α^* and β^* on accuracy of MCLP model, the red points are the possible maximum accuracies.

In compromise solution approach [35], “the best trade-off between $-\sum_{i=1}^{n}\alpha_{ii}$ and $\Sigma_i\beta_i$ is identified for an “optimal” solution. To explain this, assume the “ideal value” of $-\Sigma_i\alpha_i$ be $\alpha^*>0$ and the “ideal value” of $\Sigma_i\beta_i$ be $\beta^*>0$. Then, if $-\Sigma_i\alpha_i\ >\alpha^*$, the regret measure is defined as $-d_\alpha{}^+=\sum_{i=1}^{n}\alpha_i+\alpha^*$. Otherwise, it is defined as 0. If $-\sum_{i=1}^{n}\alpha_{ii}<\alpha^*$, the regret measure is defined as $d_\alpha^- = \alpha^*+\Sigma\alpha_i$; otherwise, it is 0. Thus, the relationship of these measures are (i) $\alpha^*+\sum_{i=1}^{n}\alpha_{ii}= d_\alpha^- - d_\alpha^+$, (ii) $|\alpha^*+\sum_{i=1}^{n}\alpha_i|= d_\alpha^- + d_\alpha^+$, and (iii) $d_\alpha^-, d_\alpha^+ \geq 0$. Similarly, we derive $\beta^*-\Sigma_i\beta_i{=}d_{\beta^-}-d_{\beta^+}$, $|\beta^*-\Sigma_i\beta_i|{=}d_{\beta^-}+d_\beta{}^+$, and $d_\beta{}^-$, $d_{\beta^+}\geq 0$”. An MCLP model for two-class separation is formulated as following [51]:

$$\begin{array}{ll} min & d_\alpha^- + d_\alpha^+ + d_\beta^- + d_\beta^+ \\ s.t. & \alpha^* + \sum_{i=1}^{n}\alpha_i = d_\alpha^- - d_\alpha^+, \\ & \beta^* - \sum_{i=1}^{n}\beta_i = d_\beta^- - d_\beta^+, \\ & (w^T.\, x_i) = b + y_i\ (\alpha_i - \beta_i),\ i=1,\ldots,n \\ & \alpha,\beta,d_\alpha^-, d_\alpha^+,\ d_\beta^-,\ d_\beta^+ \geq 0, \end{array} \tag{6}$$

here α^* and β^* are the two parameters which need to be tuned with the optimal value till the MCLP model perform in an efficient way, w and b are unrestricted variables [34]. As shown in Fig. 3 different values for α^* and β^* will cause different accuracies, as an output of the model.

3.2.1. Penalized multiple criteria linear programming

In intrusion detection problems we mostly face with strictly unbalanced classes, for example, in the KDD 99 dataset and other driven dataset from it the distribution of R2L and U2R attacks are highly unbalance. Hence, we introduce penalized parameter to the MCLP model to make it more effective in unbalanced datasets and named the enhanced model, penalized MCLP which is formulated as following:

$$\begin{array}{ll} min & d_\alpha^- + d_\alpha^+ + d_\beta^- + d_\beta^+ \\ s.t. & \alpha^* + p\frac{n_2}{n_1}\sum_{i\in B}\alpha_i + \sum_{i\in G}\alpha_i = d_\alpha^- - d_\alpha^+, \\ & \beta^* - p\frac{n_2}{n_1}\sum_{i\in B}\beta_i - \sum_{i\in G}\beta_i = d_\beta^- - d_\beta^+, \\ & (w^T.\, x_i) = b + y_i\ (\alpha_i - \beta_i), \quad i=1,\ldots,n \\ & \alpha,\beta,d_\alpha^-, d_\alpha^+,\ d_\beta^-,\ d_\beta^+ \geq 0, \end{array} \tag{7}$$

where n_1 and n_2 are the numbers of records corresponding to the two groups of classes, and $p\geq 1$ is the penalized parameter. Although, the distribution of attacks are unbalanced in KDD99 dataset, by using parameters n_2/n_1, we can get a balanced distance on the two sides of separating hyper-plane b. The value of p has been used as a penalized parameter to regularize the balance of two unbalanced classes. In the rest of this paper wherever we use MCLP term, we mean the last proposed model.

Intrusion detection can be defined as a multi-class classification problem in which two main strategies are proposed to deal with it by [52]: one-against-all and one-against-one. Although both strategies relatively can give similar results in terms of classification accuracy, the second one is adopted in this study for both MCLP and SVM classifier. Let suppose training set $T = \{(x_1,y_1),\ldots.,(x_l,y_l)\}\in(R^n\times\{1,\ldots,k\})^l$ where $x_i\in\left([x_i]_1,\ldots,[x_i]_n\right)^T$ is the input attributes vectors, $y_i\in\{1,\ldots,k\}$ is the corresponding output of x_i, n is the sample number. In one-against-all, k two class classifier will be trained and a new test instance is classified in response to the largest classifier decision function. For one-against-one strategy, $k(k-1)/2$ binary classifier will be trained and a new test instance will be classified by the result of a majority vote. In fact, each new instance is classified in one of the two classes made by binary classifier, then class of this instance determines by counting the frequency of assigned classes.

3.3. Particle swarm optimization (PSO)

In the resent years, biology inspired approaches has been used to solve complex problems in a variety of domains such as computer science, medicine, finance and engineering [7]. Swarm intelligence considered as an artificial intelligence techniques which inspired from a flock of birds, a school of fish swims or a colony of ants and their unique capability to solve complex problems [7]. Briefly, swarm intelligence (SI) considered as some methodologies, techniques and algorithms inspired by study of collective behaviors in decentralized systems [6]. Particle swarm optimization is one of these techniques, which introduced by Eberhart and Kennedy in 1995 [53]. Particle swarm optimization is a population based meta-heuristic optimization technique that simulates the social behavior of individuals, namely, particles. This technique, compare with the other algorithms in this group has several advantages such as simple to implement, scalability, robustness, quick in finding approximately optimal solutions and flexibility [6,53,54].

In particle swarm optimization, each individual of a population that considered as a representative of the potential solution move through an n-dimensional search space. After the initialization of the population, at each iteration particle seeks the optimal solution by changing its direction which consists of its velocity and position according to two factors, its own best previous experience (*pbest)* and the best experience of all particles (*gbest)*. Eqs. (8) and (9), respectively represents updating the velocity and position of each percale at iteration $[t+1]$. At the end of each iteration the performance of all particles will be evaluated by predefined fitness functions.

$$v^{id}[t+1] = w.v^{id}\ [t] + c_1\ r_1\left(p^{id,best}[t] - x^{id}[t]\right) + c_2\ r_2\left(p^{gd,best}[t] - x^{id}[t]\right) \quad d = 1,2,\dots,D \tag{8}$$

$$x^{id}[t+1] = p^{id}[t] + v^{id}[t+1] \quad d = 1,2,\dots,D \tag{9}$$

where $i = 1,2,\dots,N$, N is the number of swarm population. In D-dimensional search space, $x^{i}[t] = x^{i1}[t], x^{i2}[t], \dots., x^{iD}[t]$ represent the current position of the i^{th} particle at iteration $[t]$. Likewise, the velocity vector of each particle at iteration $[t]$ represented by$v^{i}[t] = v^{i1}[t], v^{i2}[t], \dots, v^{iD}[t]$. $p^{i,best}[t] = p^{i1}[t], p^{i2}[t], \dots, p^{iD}[t]$ represent the best position that particle i has obtained until iteration t, and $p^{g,best}[t] = p^{g1}[t], p^{g2}[t], \dots, p^{gD}[t]$ represent the previous best position of whole particle until iteration t.

To control the pressure of local and global search, the concept of an inertia weight w was introduced in the PSO algorithm by Shi and Eberhart [55]. r_1 and r_2 are two D-dimensional vectors with random number between 0 and 1. c_1 and c_2 are positive acceleration coefficients which respectively called cognitive parameter and social parameter. In fact, these two parameter control the importance of particle's self-learning versus learning from all the swarm's population.

In this research, in order to balance the global exploration and local exploitation, time-varying acceleration coefficients (TVAC) [9,56] and time-varying inertia weight (TVIW) [9,30,55] is adopted to justify the acceleration coefficients and inertia weight, respectively. Both of these concepts help PSO algorithm to have better performance to find the region of global optimum and do not trap in local minima [9,55,56]

In TVAC, the acceleration coefficients adjusted by decreasing the value of c_1 from initial value of c_{1i} to c_{1f}, while the value of c_2 is increasing from its initial value of c_{2i} to c_{2f} as shown in Eqs. (10) and (11). Moreover, in TVIW, the inertia weight w is updated according to the Eq. (12), which means a large inertia weight makes PSO has more global search ability at the beginning of the run and by a linearly decreasing the inertia weight makes PSO has better local search.

$$c_1 = c_{1i} + \frac{t}{t_{max}}(c_{1f} - c_{1i}) \tag{10}$$

$$c_2 = c_{2i} + \frac{t}{t_{max}}(c_{2f} - c_{2i}) \tag{11}$$

$$w = w_{max} - \frac{t}{t_{max}}(w_{max} - w_{min}) \tag{12}$$

here t represents the current iteration and t_{max} means the maximum number of iterations, c_{1i}, $c_{1f,}c_{2i}, c_{2f}$ are the constant values and w_{max}, w_{min} are the predefined maximum and minimum inertia weight.

3.4. Discrete binary PSO

Although the original PSO was proposed to act in continuing space, Kennedy and Eberhart [57] proposed the discrete binary version of PSO. In this model particle moves in a state space restricted to zero and one on each dimension, in terms of the changes in probabilities that a bit will be in one state or the other. The formula proposed in Eq. (8) remains unchanged except that $x^{id}[t], p^{gd,\text{best}}[t]$ and $p^{id,\text{best}}[t] \in 0, 1$ and v^{id} restricted to the [0.0, 1.0] [9,30]. By introducing the sigmoid function the velocity mapped from a continuous space to probability space as following:

$$sig\left(v^{id}\right) = \frac{1}{1 + e^{(-v^{id})}} \quad d = 1,2,\dots,D \tag{13}$$

The new particle position calculated by using the following rule:

$$x^{id}[t+1] = \begin{cases} 1, & if\ \ rnd() < sig(v^{id}) \\ 0 & if\ \ rnd() \geq sig(v^{id}) \end{cases}, d = 1,2,\dots,D \tag{14}$$

where $sig(v^{id})$ is a sigmoid function and *rnd( )* is a random number in range [0.0, 1.0].

3.5. Chaotic particle swarm optimization

Although traditional PSO gains considerable results in different fields, however, the performance of the PSO depends on the preset parameters and it often suffers the problem of being trapped in local optima. In order to further enhance the search ability of swarm in PSO and avoids the search being trapped in local optimum, chaotic concept has been introduced by Cai et al., Angeline, and Liu et al. [58–60]. Here, chaos is characterized as ergodicity, randomicity and regularity.

In this paper, Logistic equation which is a typical chaotic system adopted to make the chaotic local search as represented in the following:

$$z_{j+1} = \mu z_j(1 - z_j) j = 1,2,\dots m \tag{15}$$

here by considering n-dimensional vector $z_j = (z_{j1}, z_{j2}, \dots, z_{jn})$, each component of this system is a random value in the range $[0, 1]$, μ is the control parameter and the system of Eq. (15) has been proved to be completely chaotic when $0 \leq z_0 \leq 1$ and $\mu = 4$. Chaos queues $z_1, z_2, z_3, \dots., z_m$ are generated by iteration of Logistic equation.

In fact the basic ideas of chaotic is adopted in this paper are described as follows:

*Chaos initialization***:** In spite of standard PSO, which particle's position in the search space initialized randomly, here chaos initialization is adopted to better initialize the position of each particle and to increase the diversity of the population.

*Chaotic local search (CLS)***:** By using the chaos queues, it helps PSO to does not trapped in a local optimum besides it can cause to search the optimum quickly. It will happen by generating the chaos queues based on the optimal position ($p^{g,\text{best}}$), and then replace the position of one particle of the population with the best position of the chaos queues.

4. Proposed methodology

4.1. Fitness function

Although different performance metrics has been proposed to evaluate the effectiveness of IDSs, the most two popular of these metrics are detection rate (DR) and false alarm rate (FAR). By comparing the actual nature of a given record which here “Positive” means an “attack classes” and “Negative” means a “normal record” to the prediction ones, it’s possible to consider four outcomes for this situation as shown in Table 1, which known as the confusion matrix.

here true positive and true negative means correctly labeled the records as an attack and normal, respectively, that is, IDSs predict the labels perfectly. False positive (FP), refer to normal record is considered as an attack and False negative (FN) means those attack records falsely considered as a normal one.

A well performed IDS should has a high detection rate (DR) as well as low false positive rate. In intrusion detection domain false positive rate typically named false alarm rate (FAR). Thus, the

Table 1
Confusion matrix.

	Test result positive (Predicted as an attack)	Test result negative (Predicted as a normal record)
Actual positive class (Attack record)	True positive (TP)	False negative (FN)
Actual negative class (Normal record)	False positive (FP)	True negative (TN)

particles with higher detection rate, lower false positive rate and the small number of selected features can produce a high objective function value. Hence, in this research a weighted objective function that simultaneously takes into account trade-off between the maximizing the detection rate and minimizing the false alarm rate, along with considering the number of features is proposed according to the following equation:

$$\text{Objective function}(\mathrm{F}_{fit}) = w_{DR}.\left[\frac{TP}{(TP+FN)}\right] + w_{FAR}.\left[1-\frac{FP}{(FP+TN)}\right] + w_{F}.\left[1-\frac{\sum_{i=1}^{nF}\mathrm{f}_i}{n_F}\right] \tag{16}$$

Since any of these three elements of objective function have different effect on the performance of IDS, we convert this multiple criteria problem to a single weighted fitness function that combines the three goals linearly into one. Where w_{DR}, w_{FAR} and w_F represents the importance of detection rate, false alarm rate and number of selected features in the objective function. Detection rate or sensitivity in biomedical informatics terms, known as a true positive rate (TPR), which means the ratio of true positive recognition to the total actual positive class; $\frac{TP}{(TP+FN)}$. False alarm rate (FAR) or false positive rate (FPR) defined as: $\frac{FP}{(FP+TP)}$. f_i represents the value of feature mask ("1" represents that feature i is selected and "0" represents that feature i is not selected), and n_F indicates the number of features.

4.2. The developed TVCPSO–MCLP/SVM framework

The specific steps of TVCPSO–MCLP and TVCPSO–SVM are described as follows:

Step1 Chaotic initialization for $n+2$ particle, for the MCLP algorithm, the first two parameters are α^* and β* and for SVM algorithm the first two parameters are c and γ. The rest of n particle is binary features mask of feature sets which here is 41 features of NSL-KDD cup 99 datasets. Here in binary features mask, 1 and 0 adopted to present as selected features and discarded features, respectively.

a) Initialize a vector $z_0 = (z_{01}, z_{02}, \ldots, z_{0n})$, each component of it is set as a random value in the range [0,1], and by iteration of Logistic equation a chaos queue $z_1, z_2, \ldots, z_n$ is obtained.

b) In order to transfer the chaos queue z_j into the parameter's range the following equation is used:

$$\hat{Z}_{jk} = a_k + (b_k - a_k).\, z_{jk} \quad (k = 1, 2, \ldots, n) \tag{17}$$

where the value range of each particle defined by $[a_k, b_k]$.

Step2: Compute the fitness value of the initial vector $\hat{Z}_j(j = 1, 2, \ldots, m)$ and then choose the best M solutions as the initial positions of M particles.
Step 3: Randomly initialize the velocity of M particles, here, $v_j = (v_{j1}, v_{j2}, \ldots, v_{jn}) j = (1, 2, \ldots, M)$
Step 4: Update the velocity and position of each classifier's parameters (α^*, β^* in MCLP and c ,γ in SVM) according to Eqs. (7) and (8), and in order to update the velocity and position of the features in each particle Eqs. (7) and (14) have been used, respectively.
Step 5: Evaluate the fitness of each particle according to Eq. (16) and then compare the evaluated fitness value of each particle (personal optimal fitness (pfit)) to its personal best position ($p^{i,\mathrm{best}}$):

a) If the pfit is better than $p^{i,\mathrm{best}}$ then update the $p^{i,\mathrm{best}}$ as the current position, otherwise keep the previous ones in memory.

b) If the pfit is better than $p^{g,\mathrm{best}}$ then update the $p^{g,\mathrm{best}}$ as the current position, otherwise keep the previous$p^{g,\mathrm{best}}$.

Step 6: Optimize $p^{g,\mathrm{best}}$ by chaos local search according to the following steps:

a) Consider $T = 0$, scale the $p^{gk,\mathrm{best}}$ into the range of [0,1] by $z_k^T = \frac{p^{gk,\mathrm{best}} - a_k}{b_k - a_k}(k = 1, 2, \ldots, n)$.

b) Generate the chaos queues $Z_j^T(T = 1, 2, \ldots, m)$ by iteration of Logistic equation.

c) Obtain the solution set $p = (p^1, p^2, \ldots, p^m)$ by scale the chaotic variables Z_j^T into the decision variable according to the $p_k^T = a_k + (b_k - a_k).z_k^T$.

d) Evaluate the fitness value of each feasible solution $p = (p^1, p^2, \ldots, p^m)$, and get the best solution $\hat{p}^{g,best}$.

Step 7: If the stopping criteria are satisfied, then stop the algorithms and get the global optimum that are the optimal value of (α^*, β^* in MCLP and c, γ in SVM) and the most appropriate subset of features. Otherwise, go to step 5.

5. Experimental setting

5.1. Data description

In this paper, a modified version of the KDD Cup 99 namely, NSL-KDD data set [61] has been used to evaluate the effectiveness of proposed methods. The KDD Cup 99 dataset known as the first used dataset in the intrusion detection domain was derived from the DARPA 98 dataset in 1999. The aim of the KDD 99 competition was to evaluate different network intrusion detection methods in the aspect of distinguishing between normal connections and abnormal ones such as intrusions, attacks or malicious activities. This database contains the records of TCP connections simulated in a military network environment, containing 41 features as shown in Table 2 with the related labeled to specify them as a normal or a specific type of attacks [6,62]. From the 41 features of KDD 99 dataset, there are 32 continues features and 9 nominal variables.

Although KDD99 dataset and DARPA 98 are the most widely used benchmarks in the intrusion detection domain, they suffer from some drawbacks [63]. There are redundant, and duplicate records which cause the classifiers will be biased towards more frequent records. Thus, in order to make the KDD Cup 99 more efficient in evaluating the network intrusion detection methodologies, NSL-KDD data set was proposed by removing all redundant instances and reconstituting the dataset [61]. The 24 kinds of attacks in NSL-KDD training data set fall into four main categories: Denial of Service (DoS), Probe, User to Root (U2R), and Remote to Local (R2L), as described in the following:

- Probe: include attacks which scan the computer networks with the purpose of gathering information and finding vulnerabilities [18].

Table 2
Features of KDD Cup 99 Dataset.

Basic features	1) duration	**Content features**	10) hot	**Time-based traffic features**	23) Count	**Host-based traffic features**	32) dst_host_count
	2) protocol_type		11) num_failed_logins		24) serror_rate		33) dst_host_srv_count
	3) service		12) logged_in		25) rerror_rate		34) dst_host_same_srv_rate
	4) src_bytes		13) num_compromised		26) same_srv_rate		35) dst_host_diff_srv_rate
	5) dst_bytes		14) root_shell		27) diff_srv_rate		36) dst_host_same_src_port_rate
	6) flag		15) su_attempted		28) srv_count		37) dst_host_srv_diff_host_rate
	7) land		16) num_root		29) srv_serror_rate		38) dst_host_serror_rate
	8) wrong_fragment		17) num_file_creations		30) srv_rerror_rate		39) dst_host_srv_serror_rate
	9) urgent		18) num_shells		31) srv_diff_host_rate		40) dst_host_rerror_rate
			19) num_access_files				41) dst_host_srv_rerror_rate
			20). num_outbound_cmds				
			21). is_hot_login				
			22). is_guest_login				

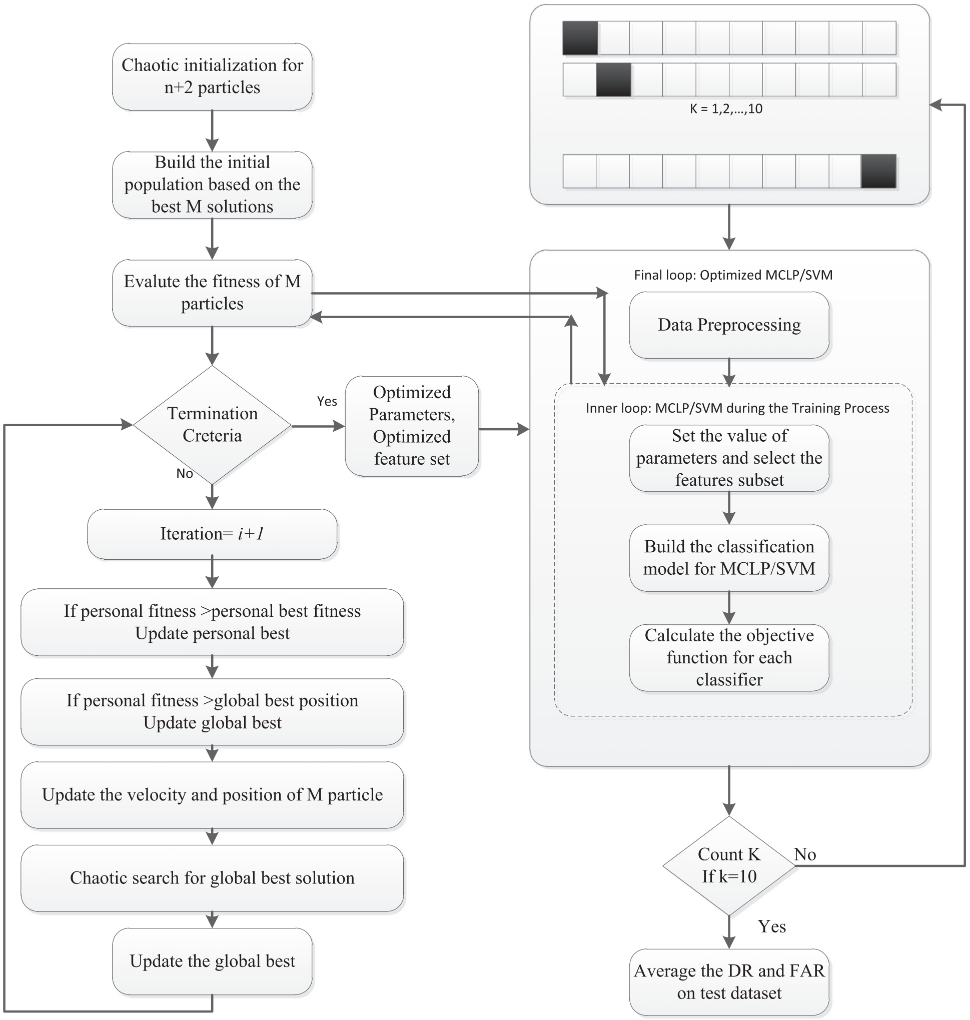

Chaotic initialization for n+2 particles
Build the initial population based on the best M solutions
Evalute the fitness of M particles
Termination Creteria
Yes
No
Optimized Parameters, Optimized feature set
Iteration= i+1
If personal fitness >personal best fitness Update personal best
If personal fitness >global best position Update global best
Update the velocity and position of M particle
Chaotic search for global best solution
Update the global best
K = 1,2,...,10
Final loop: Optimized MCLP/SVM
Data Preprocessing
Inner loop: MCLP/SVM during the Training Process
Set the value of parameters and select the features subset
Build the classification model for MCLP/SVM
Calculate the objective function for each classifier
Count K If k=10
No
Yes
Average the DR and FAR on test dataset

Fig. 4. Framework of proposed TVCPSO–MCLP/SVM.

- Denial of service (DoS): include attacks which try to make a machine or a network unavailable to its intended users by engaging all resources of that machine.
- User to root (U2R): include attacks in which attackers gain root access by access to a normal user at first and then exploit system vulnerabilities to increase its access privilege.
- Remote to user (R2L): include attacks in which attackers try to access to a remote machine by sending packets to that machine over the internet. In this kind of attack, attackers do not have a local account and try to gain it by exploiting from the vulnerabilities [18].

5.2. Experimental setup

The empirical experiment run in an Intel® Core™ i5 CPU @ 1.70 GHz computer with 4.00 GB RAM running Windows 7. The proposed Time-Varying Chaos Particle Swarm Optimization based MCLP/SVM classifier is implemented using MATLAB 2013, and LIBSVM version 3.20 which is developed by Chang and Lin [64] adopted as the SVM package.

In order to, prevent the dominance of feature values in greater numeric ranges to those in smaller numeric ranges, we normalize the data and scale them into the range of $[1,+1]$ before running the algorithms. This normalization also prevents numerical difficulties in the calculation [10]. Furthermore, by using the random sampling and K-fold cross validation proposed by Salzberg [65] we built the training dataset and testing dataset from the original data. We set K as 10 that means the original data randomly sampled into 10 subsets and each subset of the data sharing the same proportion of each class of data. In each run of the model, nine subsets are used to train the model and the reset is used to test the model. By running the model 10 times, each subset of data has an equal chance to be used in testing part, and then the rate of accuracy is computed by taking the average of the accuracy of model on testing subsets. Fig. 4. Shows the architecture of proposed TVCPSO-based parameter determination and feature selection approach for MCLP and SVM.

The details of parameter setting for TVCPSO–MCLP and TVCPSO–SVM are presented in Table 3.

5.3. Experimental results and discussion

To evaluate the performance of both proposed methods in intrusion detection problem, the 10-fold cross-validation has been adopted and the algorithms were carried out ten times and then the obtained results were averaged. According to the confusion matrix presented in Table 1, the following measurements are used to evaluate the performance of these two classifiers:

- Accuracy: the ratio of correct predicted records to the entire records $=\frac{TP+TN}{(TP+TN+FP+FN)}$
- Detection rate$=\frac{TP}{(TP+FN)}$
- False alarm rate$=\frac{FP}{(FP+TN)}$

In order to assess the effectiveness of TVCPO–MCLP and TVCPO–SVM in detecting the intrusions, two different experiments has been done, to not only compare the performance of proposed methods, but also investigate whether feature selection may further improve the results of each one or not? In the first experiment we compare the performance of TVCPSO–MCLP and TVCPSO–SVM, in condition that time varying chaos particle swarm optimization performed to find the optimal values for α^*, β^* in MCLP model and c, γ in SVM without doing feature selection. Table 4-1 and 4-2 presents the results of this experiment for TVCPO–MCLP and TVCPO–SVM, respectively.

Table 3
Parameters for initialization of proposed methods.

Variables	Values for MCLP/SVM
Number of the iterations	200
Number of the particles	8
Ranges for c	$[2^{-6}, 2^{6}]$
Ranges for γ	$[2^{-4}, 2^{4}]$
Ranges for α^*	$[-10^{-5}, -10^{-1}]$
Ranges for β^*	[10, 500]
v_{max} for continues particles $v_{min}=-v_{max}$	60% of the dynamic range of the variable on each dimension
$[v_{min}, v_{max}]$ for discrete particles	[0, 1]
$c_{1i}, c_{1f}, c_{2i}, c_{2f}$	2.5, 0.5,0.5,2.5
w_{min}, w_{max}	0.4, 0.9
w_{DR}, w_{FAR}, w_F	0.70, 0.20, 0.30

In the second experiment, the impact of feature selection, along with setting the parameters of MCLP and SVM simultaneously has been investigated. Table 5-1 and 5-2, represent the results of the second experiment in a confusion matrix.

Skewness of dataset is an influential factor which decreases the detection power of IDSs. To give an example, in KDD cup 99 testing dataset not only do not 11 attack types of “U2R” and “R2L” classes appear in training dataset, but also their distributions are severely unbalanced compare with the other attacks e.g. DoS and Probe. According to the Table 4-1 and 4-2, the results show that both classifiers have a lower detection rate in U2R and R2L class compared to the rest of classes, which indicates the poor performance of proposed methods in detecting the intrusions in highly unbalanced classes. By the way, from the Table 4-1 and 4-2 we can see that TVCPSO–MCLP obtained better detection rate in Probe, U2R and R2L classes compared to the TVCPSO–SVM. By comparing the results presented in Table 5 to the ones in Table 4, it conducted that feature selection has a positive effect on increasing the detection rate of both methods in all the cases.

As mentioned in Section 4.1, the most popular performance metrics to evaluate the effectiveness of an IDS are DR and FAR. An efficient IDS should have high DR and a low FAR. Other commonly used metric is accuracy which means classifying the normal and attack records correctly as normal and attack classes respectively; also it is used to compare the performance of proposed methods. Table 6 summarizes the results of accuracy, detection rate and false alarm rate of each method with and without feature selection. In order to further evaluate the effectiveness of those change applied to the original PSO algorithm, we repeat the experiments by doing parameter setting and feature selection with original PSO and CPSO. Here, for PSO and CPSO the acceleration coefficients c_1 and c_2 were set to 2 and w was set to 0.9, the rest of parameters remained unchanged as mentioned in Table 3.

As shown in Table 6, the value of accuracy, detection rate and false alarm rate of TVCPSO–MCLP without feature selection are 94.69%, 95.19% and 4.81% respectively, while the values of these metrics for TVCPSO–SVM without feature selection are 95.75%, 95.49% and 3.29%. Here, it means that although proposed hybrid TVCPSO–SVM shows better performance in detecting the intrusions and has a lower false alarm rate, our proposed TVCPSO–

Table 4
Result of TVCPSO–MCLP/SVM without feature selection presented in a confusion matrix.

Actual class	1) By TVCPSO–MCLP without feature selection						2) By TVCPSO–SVM without feature selection					
	Predicted class						Predicted class					
	DoS	Normal	Probe	R2L	U2R	DR[a]	DoS	Normal	Probe	R2L	U2R	DR[a]
DoS	44646	983	264	23	11	97.21	45089	748	86	4	0	**98.18**
Normal	1996	64103	985	246	12	95.19	1497	65125	424	285	11	**96.71**
Probe	283	1537	9836	0	0	**84.39**	362	1492	9802	0	0	84.09
R2L	48	249	25	671	2	**67.44**	31	360	22	582	0	58.49
U2R	0	18	0	7	27	**51.92**	1	22	0	10	19	36.54

[a] Detection rate of each class.

Table 5
Result of TVCPSO–MCLP/SVM with feature selection presented in a confusion matrix.

Actual class	1) By TVCPSO–MCLP with feature selection						2) By TVCPSO–SVM with feature selection					
	Predicted class						Predicted class					
	DoS	Normal	Probe	R2L	U2R	DR[a]	DoS	Normal	Probe	R2L	U2R	DR[a]
DoS	45301	604	16	4	0	98.64	45394	469	63	0	1	**98.84**
Normal	1361	65720	175	84	4	97.59	463	66756	70	50	3	**99.13**
Probe	620	778	10246	12	0	87.90	274	974	10408	0	0	**89.29**
R2L	28	205	13	747	2	**75.08**	37	260	19	675	4	67.84
U2R	2	18	1	0	31	**59.62**	3	26	0	2	21	40.38

[a] Detection rate of each class.

Table 6
Performance comparison of proposed methods.

Metrics	PSO-MCLP	CPSO-MCLP	TVCPSO-MCLP	PSO-SVM	CPSO-SVM	TVCPSO–SVM
	Experiment one: Parameter setting without feature selection					
Accuracy	90.72	92.47	**94.69**	93.29	94.11	**95.75**
Detection rate (DR)	91.09	91.26	**95.19**	92.67	92.05	**95.49**
False alarm rate (FAR)	7.78	4.66	**4.81**	4.78	3.34	**3.29**
	Experiment two: Parameter setting with feature selection					
Accuracy	94.50	96.06	**96.88**	96.25	97.21	**97.84**
Detection rate (DR)	94.92	95.42	**97.23**	95.48	95.48	**97.03**
False alarm rate (FAR)	3.90	**2.41**	**2.41**	2.36	**0.72**	0.87

MCLP model also has comparable accuracy with this well-known classifier. Based on the results shown in Table 6 the accuracy of TVCPSO–MCLP and TVCPSO–SVM in situations that parameter setting and feature selection have been applied simultaneously, improved by 1.03% and 1.01 % respectively. Furthermore, Table 6 shows that performance of proposed TVCPSO is consistently better than those of PSO and CPSO in both classifiers. Fig. 5-1 and 5-2 compare the performance of TVCPSO with PSO and CPSO in both classifiers based on the average accuracy with different iterations. These two figures confirm that not only did TVCPSO gain better accuracies in different iterations, but also it is faster in searching the optimum than PSO and CPSO and meeting the stopping criteria.

In order to better interpret the results obtained from our models, a comparison with other existing methods using the KDD cup 99 and NSL-KDD dataset is presented in Table 7. Here, in order to cover more recent researches in our comparison, some

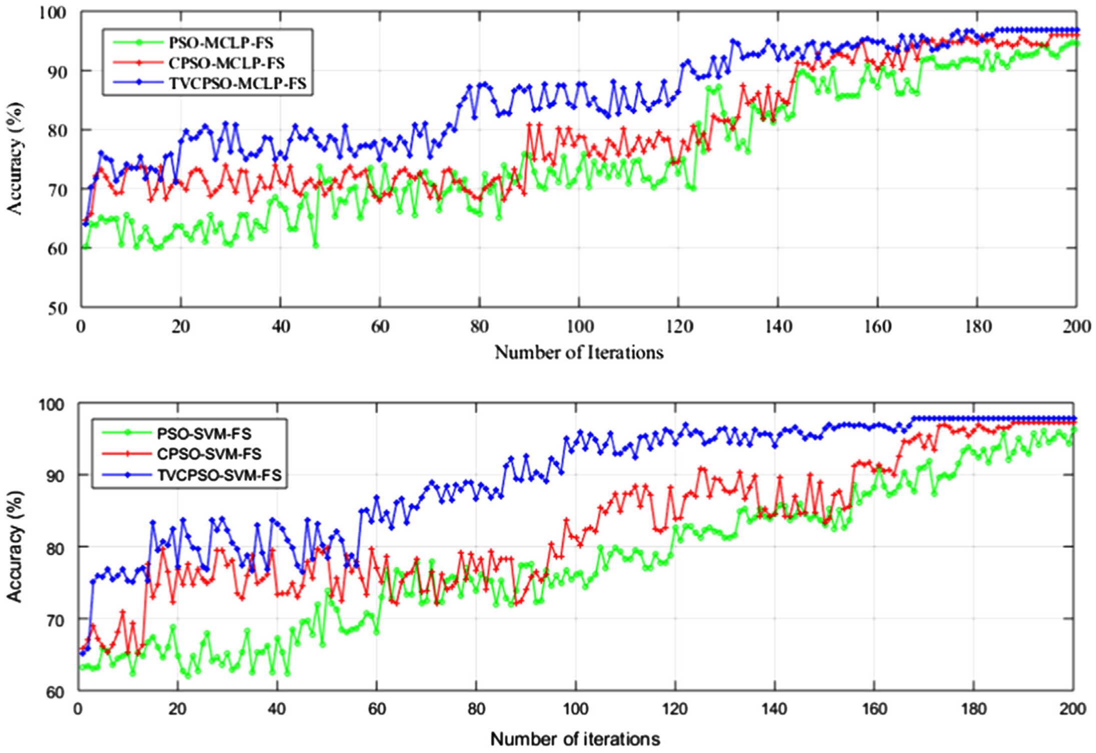


Fig. 5. (1) Compares the average accuracy obtained by MCLP classifier in different iterations on NSL-KDD 99 dataset by using PSO, CPSO and TVCPSO. (2) Compares the average accuracy obtained by the SVM classifier in different iterations on NSL-KDD 99 dataset by using PSO, CPSO and TVCPSO. FS means feature selection.

Table 7
Performance comparison of several IDSs' methods.

Authors	Methods	Data Set	FS	DR	FAR
Kuang et al. [17]	N-KPCA-GA-SVM	KDD	✓	95.26	1.03**
Singh et al. [66]	OS-ELM	NSL-KDD	✓	97.67*	1.74
Yu and J.P. Tsai [67]	GHSOM	KDD		96.02	4.92
Abadeh and Habibi [68]	EFS-ACO	KDD		94.33	N/A
Panda et al. [69]	DM-Naive Bayes	NSL-KDD		96.5	3.0
Kshirsagar and Patil [70]	AdaBoost	NSL-KDD		90.31	3.38
De la Hoz et al. [71]	KernelPCA+SVC	NSL-KDD	✓	93.4	14
Tavallaee et al. [61]	Random Forest	NSL-KDD		80.67	N/A
Ma et al. [72]	BPSO–SVM	KDD		96.77	8.01
Tsang and Kwong [73]	ACC	KDD	✓	N/A	N/A
Kayacik et al. [74]	Hierarchical SOM	KDD		90.6	1.57***
Tsang et al. [75]	MOGFIDS	KDD	✓	92.76	N/A
Proposed Method	TVCPSO–MCLP	NSL-KDD		95.19	4.81
	TVCPSO–MCLP	NSL-KDD	✓	97.23**	2.41
	TVCPSO–SVM	NSL-KDD		95.49	3.29
	TVCPSO–SVM	NSL-KDD	✓	97.03***	0.87*

* Ranked first.
** Ranked second.
*** Ranked third.

methodologies which evaluated based on KDD cup 99 also added to Table 7. According to the detection rate measure, *online sequential extreme learning machine* [66] with a little difference from our model TVCPSO–MCLP obtained the highest rate among the compared methods and our model TVCPSO–MCLP and TVCPSO–SVM along with feature selection stand in the second and third place. It is clear from the results which TVCPSO–SVM model has a distinct advantage over the other methods in terms of false alarm rate. However, it should be noted that Table 7 just provides a snapshot from comparison between our model and some of the existing methods in intrusion detection problem. Hence, it cannot be claimed that our proposed method has a better performance in all the cases and classification problems in context of intrusion detection. It is obvious a comprehensive comparison is a cumbersome task because in most of cases the number of sample size chosen from the original data set are not equal. Moreover, the sampling methods and the objective functions utilized in aforementioned methodologies are not the same. Also in some cases, the researchers did not mention the standard deviation of their results and the computational time of proposed methods. Hence, we believe that the advantages of each method is comparative and different factors should be considered to do an exhaustive comparison.

However, according to the our experiments, it is concluded that feature selection increased the detection rate and decrease the false alarm rate of proposed framework in all cases. In order to further examine which features were more influential on the objective function, the frequency of selected features through one time run of 10-fold cross-validation on the NSL-KDD 99 data set, is shown in Fig. 6.

From the results presented in Fig. 6, we can infer that the most frequent selected features based on the TVCPSO–SVM model are 2, 5, 35, 4, 14, 30, 12, 32, 33, 37, 3, 25, 10, 13, 17, 29, 40 and similarly the most frequent features selected by TVCPSO–MCLP are 4, 35, 12, 2, 31, 33, 5, 6, 14, 23, 29, 36, 10, 15, 22, 25, 30. It means that both classifiers show better results in a case that these features has been used during the classification process. For example, the accuracy of the TVCPSO–SVM increased from 95.75% to 97.84% in a case that feature selection and parameter setting has been applied simultaneously and also the result of this experiment shows the efficiency of selected subset of features on increasing the accuracy of the TVCPSO–MCLP from 94.69% to 96.88%. In order to evaluate

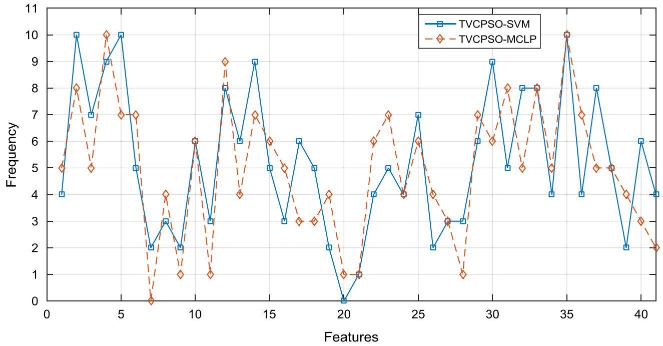


Fig. 6. The frequency of the selected features in one run 10-fold CV process on the NSL-KDD 99 data set.

the appropriateness of selected features by our model, a comparison between different methods presented in Table 8.

Authors in Dehghanimohammadabadi and Keyser [76], proposed a filter method based on Information Gain Ratio (IGR) and the k-means classifier to extract the features with the highest IGR from the dataset and assesses the detection rate of the optimal subset of features. Gradually feature removal method (GFR) [19], by gradually remove the less important features, decides the importance order of 41 features in KDD dataset. Some authors tried to determine the importance of features for each class separately [31,77]. In Guariguata et al. [77], a ranking method is proposed by calculating the weights from the support vector decision function and in Hoz et al. [31] the authors compared three methods which are forward feature selection algorithm (FFSA), modified mutual information-based feature selection algorithm (MMIFS) and linear correlation-based feature selection algorithm (LCFS) for ranking the features in each class of KDDcup99 dataset.

As shown in Table 8, we can conclude that although the order of critical selected features are different in each method, among the most frequent selected features, common features are notable which exist in most circumstances. Concretely, the common features are 2nd, 4th, 5th, 12th, 30th, 31th, 33th, 35th features that stand for protocol_type, src_bytes, dst_bytes, logged_in, srv_rerror_rate, srv_diff_host_rate, dst_host_srv_count, dst_host_diff_srv_rate. Moreover, feature 20th and 21th which refers to num_outbound_cmds and is_hot_login receptively has the lowest frequency in TVCPSO–SVM model which there is already the same results in IGR/K-means method [76]. The lowest frequent features selected by TVCPSO–MCLP model are 7th, 9th, 11th, 20th, 21th, 28th that stand for land, urgent, num_failed_logins, num_outbound_cmds, is_hot_login, srv_count, which in methods like IGR/K-means [76], GFR [19] and SVDF[47] also ranked as the less important features.

6. Conclusions and future work

In this paper, we proposed a time varying chaos particle swarm optimization method to provide a new machine learning intrusion detection methodology based on two conventional classifiers; multiple criteria linear programming (MCLP) and support vector machine (SVM). The proposed method has been applied to setting the parameters of these classifiers as well as providing the most appropriate subset of features, simultaneously. In order to have an efficient IDS with high detection rate and a low false positive rate, a weighted objective function which takes into account these metrics besides feature selection has been proposed. The empirical results show not only is the proposed method more accurate in detecting intrusions, but also it can choose a more discriminative feature subset. In this paper, we evaluate the performance of

Table 8
Selected features by different methods sorted by their importance.

Algorithm	Features
IGR/K-means [76]	35,12,37,11,30,6,14,22,9,3,32,31,5,2,1,17,36,23,16,18,19,10,15,24,33,38,39,25,26,4,29,41,13,28,8,27,34,40,21,7,20
GFR [19]	35,33,2,14,36,10,4,32,40,29,8,38,31,38,34,25,27,15,19,3,23,1,16,13,41,30,7,24,11,21,20,17,9,6,5,39,26,18,12,22,28
SVDF [77][a]	1,5,6,23,24,25,26,32,36,38,39,2,3,4,10,12,29,33,34,7,8,9,11,13,14,15,16,17,18,19,20,21,22,27,28,30,31,35,36,37,40,41
FFSAMMIFSLCFS [31][a]	3,5,38,3
	8,5,23,6,2,24,41,36,3
	36 32,27,23,38,41,24,13,2,40,22,30,25,28,35,26,37,12,36,39,1,10,14,11,17,33,16,19,18,9,5,34,31,6,3,29,3
TVCPSO–MCLP (our model)	35,4,12,2,33,31,5,14,29,6,23,36,30,25,10,15,22,32,37,3,38,1,34,16,13,24,8,19,26,39,17,40,18,27,41,11,8,9,21,20,7
TVCPSO–SVM (our model)	2,5,35,4,14,30,12,32,33,37,3,25,10,13,17,29,40,6,15,18,23,31,38,1,22,24,34,36,41,8,11,16,27,28,7,9,19,26,39,21,20

[a] Features ordered for DOS attack.

multiple criteria linear programming in the intrusion detection problem which has obtained comparable results to the SVM, and also has achieved better detection rates in R2L and U2R attack classes compared to the SVM.

In order to further improve the performance of multiple criteria linear programming for intrusion detection, we plan to apply this model with kernel function. Moreover, the results of SVM classifier were based on the RBF kernel. Hence, the proposed method can be used to optimize the SVM with the other kernels. For the SVM classifier, an objective function can be defined which takes into account the number of support vectors to exploit the maximum generalization capability of SVM as described in Chen et al. [9] and decrease the training time.

Acknowledgments

This work has been partially supported by Grants from National Natural Science Foundation of China (Nos. 61472390, 11271361, 71331005), Major International (Ragional) Joint Research Project (No. 71110107026) and CAS-TWAS President's Fellowship for International PhD Student.

References

[1] C.-F. Tsai, et al., Intrusion detection by machine learning: a review, Expert Syst. Appl. 36 (10) (2009) 11994–12000.
[2] G. Kou, et al., Multiple criteria mathematical programming for multi-class classification and application in network intrusion detection, Inf. Sci. 179 (4) (2009) 371–381.
[3] Cisco Systems, I., Cisco 2015 Annual Security Report. Retrieved from ⟨https://www.cisco.com/web/offer/gist_ty2_asset/Cisco_2015_ASR.pdf⟩, 2015.
[4] Y.Y. Chung, N. Wahid, A hybrid network intrusion detection system using simplified swarm optimization (SSO), Appl. Soft Comput. 12 (9) (2012) 3014–3022.
[5] H.-J. Liao, et al., Intrusion detection system: a comprehensive review, J. Netw. Comput. Appl. 36 (1) (2013) 16–24.
[6] S.X. Wu, W. Banzhaf, The use of computational intelligence in intrusion detection systems: a review, Appl. Soft Comput. 10 (1) (2010) 1–35.
[7] C. Kolias, G. Kambourakis, M. Maragoudakis, Swarm intelligence in intrusion detection: a survey, Comput. Secur. 30 (8) (2011) 625–642.
[8] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey, ACM Comput. Surv. (CSUR) 41 (3) (2009) 15.
[9] Hl Chen, et al., Towards an optimal support vector machine classifier using a parallel particle swarm optimization strategy, Appl. Math. Comput. 239 (0) (2014) 180–197.
[10] S.-W. Lin, et al., Particle swarm optimization for parameter determination and feature selection of support vector machines, Expert Syst. Appl. 35 (4) (2008) 1817–1824.
[11] W.-H. Chen, S.-H. Hsu, H.-P. Shen, Application of SVM and ANN for intrusion detection, Comput. Oper. Res. 32 (10) (2005) 2617–2634.
[12] G. Kim, S. Lee, S. Kim, A novel hybrid intrusion detection method integrating anomaly detection with misuse detection, Expert Syst. Appl. 41 (4) (2014) 1690–1700.
[13] Y. Liao, V.R. Vemuri, Use of K-Nearest Neighbor classifier for intrusion detection1, Comput. Secur. 21 (5) (2002) 439–448.
[14] S. Mukherjee, N. Sharma, Intrusion Detection using Naive Bayes Classifier with Feature Reduction, Procedia Technol. 4 (0) (2012) 119–128.
[15] L. Koc, T.A. Mazzuchi, S. Sarkani, A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier, Expert Syst. Appl. 39 (18) (2012) 13492–13500.
[16] G. Wang, et al., A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering, Expert Syst. Appl. 37 (9) (2010) 6225–6232.
[17] F. Kuang, W. Xu, S. Zhang, A novel hybrid KPCA and SVM with GA model for intrusion detection, Appl. Soft Comput. 18 (2014) 178–184.
[18] S.-W. Lin, et al., An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection, Appl. Soft Comput. 12 (10) (2012) 3285–3290.
[19] Y. Li, et al., An efficient intrusion detection system based on support vector machines and gradually feature removal method, Expert Syst. Appl. 39 (1) (2012) 424–430.
[20] O.L. Mangasarian, D.R. Musicant, Successive overrelaxation for support vector machines, Neural Netw. IEEE Trans. 10 (5) (1999) 1032–1037.
[21] C.-C. Chang, C.-J. Lin, Training v-support vector classifiers: theory and algorithms, Neural Comput. 13 (9) (2001) 2119–2147.
[22] J.A. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Process. Lett. 9 (3) (1999) 293–300.
[23] R. Khemchandani, S. Chandra, Twin support vector machines for pattern classification, Pattern Anal. Mach. Intell. IEEE Trans. 29 (5) (2007) 905–910.
[24] Y. Tian, et al., Nonparallel support vector machines for pattern classification, Cybern. IEEE Trans. 44 (7) (2014) 1067–1079.
[25] X. Chang, et al., Complex event detection using semantic saliency and nearly-isotonic SVM, in: Proceedings of the 32nd International Conference on Machine Learning (ICML-15), 2015.
[26] C.A. Catania, F. Bromberg, C.G. Garino, An autonomous labeling approach to support vector machines algorithms for network traffic anomaly detection, Expert Syst. Appl. 39 (2) (2012) 1822–1829.
[27] X.-s Gan, et al., Anomaly intrusion detection based on PLS feature extraction and core vector machine, Knowl. Based Syst. 40 (2013) 1–6.
[28] Y. Zhang, N. Meratnia, P.J. Havinga, Distributed online outlier detection in wireless sensor networks using ellipsoidal support vector machine, Ad hoc Netw. 11 (3) (2013) 1062–1074.
[29] R. Chitrakar, C. Huang, Selection of Candidate Support Vectors in incremental SVM for network intrusion detection, Comput. Secur. 45 (2014) 231–241.
[30] C.-L. Huang, J.-F. Dun, A distributed PSO–SVM hybrid system with feature selection and parameter optimization, Appl. Soft Comput. 8 (4) (2008) 1381–1391.
[31] E. de la Hoz, et al., Feature selection by multi-objective optimisation: Application to network anomaly detection by hierarchical self-organising maps, Knowl. Based Syst. 71 (2014) 322–338.
[32] M. Yao, et al., A novel classification method based on the ensemble learning and feature selection for aluminophosphate structural prediction, Microporous Mesoporous Mater. 186 (2014) 201–206.
[33] C.O. Sakar, O. Kursun, F. Gurgen, A feature selection method based on kernel canonical correlation analysis and the minimum Redundancy–Maximum Relevance filter method, Expert Syst. Appl. (2012) 39.
[34] S.M.H. Bamakan, et al., A new intrusion detection approach using pso based multiple criteria linear programming, Procedia Comput. Sci. 55 (2015) 231–237.
[35] Y. Shi, et al., Optimization Based Data Mining: Theory and Applications, Springer, Germany, 2011.
[36] R. Sheikhpour, M.A. Sarram, R. Sheikhpour, Particle swarm optimization for bandwidth determination and feature selection of kernel density estimation based classifiers in diagnosis of breast cancer, Appl. Soft Comput. (2015).
[37] A.R. Lima, A.J. Cannon, W.W. Hsieh, Nonlinear regression in environmental sciences by support vector machines combined with evolutionary strategy, Comput. Geosci. 50 (2013) 136–144.
[38] Y. Zhang, et al., Binary PSO with mutation operator for feature selection using decision tree applied to spam detection, Knowl. Based Syst. 64 (2014) 22–31.
[39] Z.-Y. Chen, Z.-P. Fan, M. Sun, A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data, Eur. J. Op. Res. (2012).
[40] H. Zhong, et al., Comparing the learning effectiveness of BP, ELM, I-ELM, and SVM for corporate credit ratings, Neurocomputing 128 (2014) 285–295.
[41] V.N. Vapnik, V. Vapnik, Statistical learning theory, Wiley, New York, 1998.

[42] B.E. Boser, I.M. Guyon, V.N. Vapnik, A training algorithm for optimal margin classifiers. in: Proceedings of the Fifth Annual Workshop on Computational Learning Theory. ACM, 1992.

[43] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.

[44] Q. Wu, R. Law, An intelligent forecasting model based on robust wavelet ν-support vector machine, Expert Syst. Appl. 38 (5) (2011) 4851–4859.

[45] F. Glover, Improved Linear Programming Models for Discriminant Analysis, Decis. Sci. 21 (4) (1990) 771–785.

[46] Y. Shi, et al., Data mining via multiple criteria linear programming: applications in credit card portfolio management, Int. J. Inf. Technol. Decis. Mak. 1 (01) (2002) 131–151.

[47] H. Jiawei, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, San Francisco, CA, itd (2001), p. 5.

[48] J. He, et al., Classifications of credit cardholder behavior by using fuzzy linear programming, Int. J. Inf. Technol. Decis. Mak. 3 (04) (2004) 633–650.

[49] O.A. Arqub, et al., Numerical solutions of fuzzy differential equations using reproducing kernel Hilbert space method, Soft Comput. (2015) 1–20.

[50] O.A. Arqub, Adaptation of reproducing kernel algorithm for solving fuzzy Fredholm–Volterra integrodifferential equations, Neural Comput. Appl. (2015) 1–20.

[51] Y. Shi, Multiple criteria optimization-based data mining methods and applications: a systematic survey, Knowl. Inf. Syst. 24 (3) (2010) 369–391.

[52] C.-W. Hsu, C.-J. Lin, A comparison of methods for multiclass support vector machines, Neural Netw. IEEE Trans. 13 (2) (2002) 415–425.

[53] J. Kennedy, R. Eberhart, Particle swarm optimization. in: Proceedings of the 1995 IEEE International Conference on Neural Networks, 1995. Part 4 (of 6) Perth: pp. 1942–1948.

[54] S. Olariu, A.Y. Zomaya, Handbook of Bioinspired Algorithms and Applications, CRC Press, USA, 2005.

[55] Y. Shi, R. Eberhart, A modified particle swarm optimizer. in Evolutionary Computation, 1998. IEEE World Congress on Computational Intelligence, The 1998 IEEE International Conference on, 1998, IEEE.

[56] A. Ratnaweera, S.K. Halgamuge, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, Evolut. Comput. IEEE Trans. 8 (3) (2004) 240–255.

[57] J. Kennedy, R.C. Eberhart, A discrete binary version of the particle swarm algorithm. in Systems, Man, and Cybernetics, 1997, Computational Cybernetics and Simulation, 1997 IEEE International Conference on, 1997, IEEE.

[58] J. Cai, et al., Chaotic particle swarm optimization for economic dispatch considering the generator constraints, Energy Convers. Manag. 48 (2) (2007) 645–653.

[59] P.J. Angeline, EvolutionaRy Optimization Versus Particle Swarm Optimization: Philosophy and Performance Differences, Evolutionary Programming VII, Springer, Germany, 1998.

[60] B. Liu, et al., Improved particle swarm optimization combined with chaos, Chaos Solitons Fractals 25 (5) (2005) 1261–1271.

[61] M. Tavallaee, et al., A detailed analysis of the *KDD CUP 99* data set. in: Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009, 2009.

[62] The KDD99 Dataset, Reterived April 15, 2015, from 〈http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html〉, 1998.

[63] M.V. Mahoney, P.K. Chan., An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Nnetwork Anomaly Detection, Recent Advances in Intrusion Detection, Springer, Germany, 2003.

[64] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 27:1–27:27.

[65] S.L. Salzberg, On comparing classifiers: pitfalls to avoid and a recommended approach, Data Min. Knowl. Discov. 1 (3) (1997) 317–328.

[66] R. Singh, H. Kumar, R. Singla, An intrusion detection system using network traffic profiling and online sequential extreme learning machine, Expert Syst. Appl. 42 (22) (2015) 8609–8624.

[67] Z. Yu, J.J. Tsai, T. Weigert, An adaptive automatically tuning intrusion detection system, ACM Trans. Auton. Adapt. Syst. (TAAS) 3 (3) (2008) 10.

[68] M.S. Abadeh, J. Habibi, A hybridization of evolutionary fuzzy systems and ant colony optimization for intrusion detection, ISC Int. J. Inf. Secur. 2 (2015) 1.

[69] M. Panda, A. Abraham, M.R. Patra, Discriminative multinomial naive bayes for network intrusion detection, in: Information Assurance and Security (IAS), 2010 Sixth International Conference on, 2010, IEEE.

[70] V. Kshirsagar, D.R. Patil, Application of variant of adaboost based machine learning algorithm in network intrusion detection, Int. J. Comput. Sci. Secur. (IJCSS) 4 (2) (2010) 1–6.

[71] E. de la Hoz, et al., Network Anomaly Classification By Support Vector Classifiers Ensemble and Non-linear Projection Techniques, Hybrid Artificial Intelligent Systems, Springer, Germany (2013), p. 103–111.

[72] J. Ma, X. Liu, S. Liu, A new intrusion detection method based on BPSO-SVM, in: Computational Intelligence and Design, 2008, ISCID'08, International Symposium on, 2008, IEEE.

[73] C.-H. Tsang, S. Kwong, Ant colony clustering and feature extraction for anomaly intrusion detection, Swarm Intelligence in Data Mining, Springer, Germany (2006), p. 101–123.

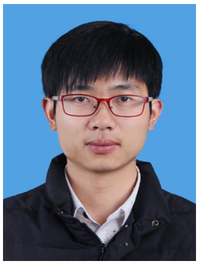
[74] H.G. Kayacik, A.N. Zincir-Heywood, M.I. Heywood, A hierarchical SOM-based intrusion detection system, Eng. Appl. Artif. Intell. 20 (4) (2007) 439–451.

[75] C.-H. Tsang, S. Kwong, H. Wang, Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection, Pattern Recognit. 40 (9) (2007) 2373–2391.

[76] M. Dehghanimohammadabadi, T. Keyser, Tradeoffs between objective measures and execution speed in Iterative Optimization-based Simulation (IOS), Winter Simulation Conference, California, USA, 2015.

[77] L. Guariguata, et al., Global estimates of diabetes prevalence for 2013 and projections for 2035, Diabetes Res. Clin. Pract. 103 (2) (2014) 137–149.

Seyed Mojtaba Hosseini Bamakan is currently a PhD student in School of Management and Economics at University of Chinese Academy of Sciences (UCAS), and researcher in Key Laboratory of Big Data Mining and Knowledge Management, UCAS, China. He reserved his master degree in IT management field from Allameh Tabataba'i University (ATU), Iran in 2009, and BSc in Industrial Management in 2007, IKIU, Iran. His current research interests include Information security, Business intelligence, Data mining and intelligent optimization techniques.



Huadong Wang is currently a PhD student with the College of Mathematica Science, University of Chinese Academy of Sciences, Beijing, China. He is also studying in Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing, China. His current research interests include support vector machines, optimization theory and applications and data mining

Tian Yingjie received the First degree in mathematics in 1994, the masters degree in applied mathematics in 1997, and the Ph.D. degree in management science and engineering. He is currently a Professor with the Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing, China. He has published four books about support vector machines (SVMs), one of which has been cited over 1000 times. His current research interests include SVMs, optimization theory and applications, data mining, intelligent knowledge management, and risk management.

Yong Shi is currently an Executive Deputy Director with the Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing, China. Since 1999, he has been the Charles W. and Margre H. Durham Distinguished Professor of information technology with the College of Information Science and Technology, Peter Kiewit Institute, University of Nebraska, Lincoln, NE, USA. He has published over 17 books, 200 papers in various journals, and numerous conferences/proceedings papers. His current research interests include business intelligence, data mining, and multiple criteria decision making. Dr. Shi was a recipient of many distinguished awards, including the George Cantor Award of the International Society on Multiple Criteria Decision Making (MCDM) in 2009, the Outstanding Young Scientist Award from the National Natural Science Foundation of China in 2001, and the Speaker of Distinguished Visitors Program (DVP) for 1997 to 2000, IEEE Computer Society. He has consulted or worked on business projects for a number of international companies in data mining and knowledge management. He is the Editor-in-Chief of the International Journal of Information Technology and Decision Making (SCI) and Annals of Data Science and a member of editorial boards for a number of academic journals.